S©MR

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ
Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru
ISSN 1813-3304

Vol. 20, No. 2, pp. 144–145 (2023)
https://doi.org/10.33048/semi.2023.20.???

УДК 004.272
MSC 11A07

HIGH SPEED ALGORITHM FOR NUMBER SIGN DETECTION IN RNS BASED ON AKUSHSKY CORE FUNCTION FOR A SPECIAL SET OF MODULI $\{2^{2n}-1, 2^{n+a}, 2^{2n}+1\}$

V.V. LUTSENKO, **M.V. ZGONNIKOV**, AND **M.G. BABENKO**

Communicated by P.P. Petrov

Abstract: The Residue Number System is widely used in various fields, such as cryptography, digital signal processing and image processing, where it is necessary to perform the operation of number sign determination. However, number sign detection is one of the most challenging computational problems in the context of the Residue Number System. This paper presents an algorithm for number sign detection in the Residue Number System based on the use of the minimal Akushsky kernel function for a special set of moduli $\{2^{2n}-1, 2^{n+a}, 2^{2n}+1\}$. The results obtained show that the proposed algorithm outperforms other number sign detection methods by an average of 88.04 %.

Keywords: residue number system, Akushsky core function, sign detection, high-speed algorithm, approximate method.

Lutsenko, V.V., Zgonnikov M.V., Babenko M.G., High speed algorithm for number sign detection in RNS based on Akushsky core function for a special set of moduli $\{2^{2n}-1, 2^{n+a}, 2^{2n}+1\}$.

The research was supported by the Russian Science Foundation Grant No. 22-71-10046, https://rscf.ru/en/project/22-71-10046/.

Received December, 11, 2023, Published December, 31, 2023.

1 Introduction

In the realm of numerical systems, a plethora of options exist, each with their own advantages and drawbacks. Among the array of options under consideration, the Residue Number System (RNS) has garnered significant attention from scholars, particularly within the realm of computer systems research. One of the most striking features of the RNS is its exceptional properties, particularly its effectiveness in multiplication, subtraction, and addition operations [14]. RNS stands out for its ability to independently execute crucial mathematical operations on each residue digit, removing the necessity for laborious carry propagation. This distinctive feature not only enhances computation speed, but it also presents energy efficiency benefits, making RNS a compelling choice for specific applications [16]. However, it is important to acknowledge that RNS has its limitations. Although RNS has impressive performance in some domains, it has limitations in others. The detection of signs, division and magnitude comparison are all actions that pose challenges within the RNS framework and require careful consideration. These limitations are particularly important in crucial tasks that depend on RNS, such as arithmetic calculations and computational geometry [1].

In investigating the identification of the sign in the RNS, various strategies have been explored [2]. A common method for detecting sign in the RNS involves the conversion of the system into a Weighted Number System (WNS) [17]. This is followed by comparison with a threshold value set at half of the RNS range. Although this algorithm has been widely used, there are ongoing efforts to improve its efficiency, resulting in advancements in the field.

Previous studies, such as [3, 4], have introduced algorithms for determining the sign of a number using a specific set of moduli $\{2^n - 1, 2^n, 2^n + 1\}$. The conversion from RNS to the Mixed Radix System (MRS) follows the Mixed Radix Conversion (MRC) algorithm, which is detailed in [5]. Additionally, the paper by [6] presents a novel technique for sign determination that utilizes the Akushsky core function and the number's rank. Furthermore, a novel technique for comparing values within the RNS structure sans the need for converting into the WNS is suggested. This method is grounded on the diagonal function's competency to determine the polarity of a value straightforwardly within RNS, as explained in [7].

In this paper, we investigate sign detection algorithms in RNS. Our study presents a new high-speed algorithm using the Akushsky core function. This algorithm is specifically designed to work with $\{2^{2n} - 1, 2^{n+a}, 2^{2n} + 1\}$ moduli. In the following sections, we will delve into the details of this method and explain how it works.

The article will be structured as follows: Section 2 provides an introduction to the fundamental terminology of the Residue Number System and outlines its general properties and methods. Section 3 delves into the Akushsky core function, elucidating its properties and methods. This section also explores the concept of positional characteristics of numbers and offers insights into

various techniques for sign detection. Section 4 introduces our proposed algorithm and establishes its theoretical foundations. Section 5 presents a detailed account of an experimental study, including the obtained results and comprehensive discussions of the findings. Section 6 encapsulates the article with conclusions drawn from the study and an outline of potential directions for future research.

2 The Residue Number System

The RNS is defined by a collection of positive coprime integers, denoted as p_i, which serve as the foundational elements of the RNS [13]. The system's dynamic range, represented as P, is determined by the multiplication of these individual moduli p_i where.

$$P = \prod_{i=1}^{n} p_i. \tag{1}$$

Let denote any integer. The Chinese Remainder Theorem (CRT) [12] stipulates, that any integer $X \in [0, P)$ has a representation $(x_1, x_2, ..., x_n)$ in RNS $\{p_1, p_2, ..., p_n\}$, where residues $x_i = |X|_{p_i}$, also called residue digits, are defined as

$$x_i \equiv X \pmod{p_i}. \tag{2}$$

To conduct operations on numbers in RNS, mathematical operations are executed independently on the remainders obtained from each moduli. For example, calculations in RNS are performed according to equation:

$$X * Y = (x_1 * y_1, x_2 * y_2, ..., x_n * y_n). \tag{3}$$

for $X, Y \in [0, P)$ and $* \in \{+, -, \times\}$. Each RNS moduli is pairwise coprime with other moduli, i.e., condition is met: $gcd(p_i, p_j) \neq 1, \forall i \neq j$.

Operations on each i-th residue operate independently, enabling concurrent processing. In addition, the self-contained nature of these residues, based on the RNS system, provides the opportunity to study self-correction properties. Whenever necessary, each residue can be inspected for correctness and, if required, reintroduced or extracted from the system, without violating the condition that $X < P$, as the system transitions to the new basis.

Non-modular RNS operations refer to mathematical operations such as: division, determining the number's sign, expanding the base, and comparison.

The sign in the residual class system is most often introduced by splitting the range into two parts, then given the dynamic range of P in the RNS we can represent the numbers

$$-\frac{P-1}{2} \leq X \leq \frac{P-1}{2}, if\ P\ is\ odd,$$

$$-\frac{P}{2} \leq X \leq \frac{P}{2} - 1, if\ P\ is\ even.$$

Then X is positive if $0 \leq X \leq \frac{P}{2} - 1$ if P is even, $0 \leq X \leq \frac{P-1}{2}$ if X is odd. X is negative if $\frac{P}{2} \leq X < P$ if P is even, $\frac{P+1}{2} \leq X < P$ if X is odd.

The conventional method for determining the sign of a number in RNS requires transforming the number to WNS and examining its positional representation against the criterion that if the outcome is $X > \frac{P}{2}$, then P is subtracted to indicate a negative number. CRT improves the conversion of a number from RNS to WNS by adhering to the following rules. When using the comparison system (2) to represent a number within the RNS framework, it provides the solution for X. To ascertain a specific number, the system must be effectively resolved. The CRT plays a crucial role by leveraging a theorem related to the inverses of moduli, aiding in the resolution of this system. Consequently, the desired number can be efficiently retrieved by using the following equation.

$$X = \left| \sum_{i=1}^{n} P_i \cdot x_i \cdot \left| P_i^{-1} \right|_{p_i} \right|_P = \sum_{i=1}^{n} P_i \cdot x_i \cdot \left| P_i^{-1} \right|_{p_i} - r_X P, \tag{4}$$

where P is the dynamic range, $P_i = \frac{P}{p_i}$, $\left| P_i^{-1} \right|_{p_i}$ is the multiplicative inversion of P_i modulo p_i, which can be achieved through various methods, such as utilizing the Euclidean algorithm [8]. r_X is the rank of the number indicates how frequently the range value must be subtracted from the resulting number to realign it within the specified range.

Some other techniques, such as the MRS method [15], use different RNS properties to determine the sign of a number. However, the majority of these approaches are problematic due to their computational complexity ranging from $O(n)$ to $O(n^2)$. Furthermore, these methods convert the number from RNS to WNS, which reduces the speed and effectiveness of data processing in RNS-based algorithms. The core function approach can avoid any conversions by utilizing the positional characteristic of a number. Nevertheless, this approach has some drawbacks due to its execution complexity.

3 Methods for Determining the Sign of a Number Based on the Core Function

3.1. Core Function Method. The previous approach poses challenges due to its intricate execution. This article will examine a superior method proposed by Akushsky in his publication [9], let's examine it.

Led by the objective of reducing computational complexity in determining the sign of a number in RNS through identifying positional characteristics, I. Y. Akushsky, V. M. Burtsev, and I. T. Pak [10] developed a new function known as the Akushsky core function. The function defined by the following equation.

$$C(X) = \sum_{i=1}^{n} w_i \left\lfloor \frac{X}{p_i} \right\rfloor. \tag{5}$$

Where integer values of w_i represent constants determined by the choice of the interpolation point, they define each specific core function and may vary depending on the particular problem at hand. Moreover, the weights

of w_i can be arbitrary to a certain extent. An algorithm for determining the optimal weights for the Akushsky function is presented in [11]. The core function range value is calculated as:

$$C(P) = C_P = \sum_{i=1}^{n} w_i P_i. \tag{6}$$

Furthermore, for the basis $B_i = P_i \cdot \left|P_i^{-1}\right|_{p_i}$, the core is also defined:

$$C(B_i) = B_i \cdot \frac{C(P)}{P} - \frac{w_i}{p_i}. \tag{7}$$

Have a set of moduli and weights as well as the core functions of the bases $C(B_i)$, we can obtain a function of the number X itself:

$$C(X) = \left| \sum_{i=1}^{n} x_i \cdot C(B_i) \right|_{C_P}. \tag{8}$$

The positional characteristic of a number can be determined using its core function without the need to translate the number in WNS. Thus, $C(X)$ suffices in determining the number's sign based on the definition of its core function. Based on CRT, if $C(X) < C\left(\frac{P}{2}\right)$, then the number is positive, and if $C(X) \geq C\left(\frac{P}{2}\right)$, then the number is negative.

Example 1. We are given a system of bases $p_1 = 15, p_2 = 8, p_3 = 17$ and weights $w_1 = 0, w_2 = 0, w_3 = 1$.The dynamic range $P = 15 \cdot 8 \cdot 17 = 2040$ and $\frac{P}{2} = \frac{2040}{2} = 1020$.

Firstly, find the values of P_i and B_i:

$$P_1 = \frac{P}{p_1} = 136, P_2 = \frac{P}{p_2} = 255, P_3 = \frac{P}{p_3} = 120.$$

$$B_1 = P_1 \cdot \left|P_1^{-1}\right|_{p_1} = 136, B_2 = P_2 \cdot \left|P_2^{-1}\right|_{p_2} = 1785, B_3 = P_3 \cdot \left|P_3^{-1}\right|_{p_3} = 120.$$

Secondly, calculate the values of $C(P)$ and $C\left(\frac{P}{2}\right)$:

$$C(P) = \left\lfloor \frac{2040}{17} \right\rfloor = 120,$$

$$C\left(\frac{P}{2}\right) = \left\lfloor \frac{1020}{17} \right\rfloor = 60.$$

Then, let us find $C(B_i)$ by using equation (7):

$$C(B_1) = 136 \cdot \frac{120}{2040} = 8, C(B_2) = 1785 \cdot \frac{120}{2040} = 105,$$

$$C(B_3) = 120 \cdot \frac{120}{2040} - \frac{1}{17} = 7.$$

Having these values, you can calculate the sign of the number $X = (2, 7, 13)$, according to the equation (8):

$$C(X) = |2 \cdot 8 + 7 \cdot 105 + 13 \cdot 7|_{120} = |842|_{120} = 2.$$

Thus $2 < 60$, so $X = (2, 7, 13)$ is a positive number which is true because $X = 47 > 0$.

3.2. Core Function Method Based on the Rank of a Number. While the examined method is more efficient than the CRT method, it should be noted that the division operation modulo $C(P)$ can become increasingly complex and may negatively impact system performance, particularly when dealing with larger moduli. Therefore, it is necessary to investigate alternative approaches for quickly determining the sign of a number using Akushsky's central function.

As previously explained, the utilization of non-modular operations within the RNS framework presents multiple challenges. To address these challenges, a proposed solution involves integrating positional characteristics, which not only indicate a number's primary function but also its ordinal value.

Nonetheless, defining and comparing these positional features may pose significant difficulties.

$$x_1B_1 + x_2B_2 + ... + x_nB_n = N \pmod{P} = N + r_XP.$$

Here, the rank denotes the coefficient of P, rather than a modular quantity. The relationship between the rank of the number and the core is elaborated in depth in [10]. Thus, equation (7) can be presented as follows:

$$X = \left(\sum_{i=1}^{n} x_i \cdot B_i\right) - r_XP, \tag{9}$$

where r_X—rank of number X. Rank calculation equation is as follows:

$$r_X = \left\lfloor \frac{\sum_{i=1}^{n} x_i \cdot B_i}{P} \right\rfloor. \tag{10}$$

Given equation (4) and the fact that p_i is a divisor of $C(B_i)$, and often $P_n = C(P)$, we obtain

$$C(X) = \left(\sum_{i=1}^{n} x_i \cdot C(B_i)\right) - r_XC(P). \tag{11}$$

As a result, the function is applied to the rank of the number. Additionally, the resulting expression becomes more efficient than the one obtained in Section 3.1 due to the elimination of the complex modulo division. However, this result was also found in [10], and its effectiveness was demonstrated in the hardware implementation of the RNS system based on programmable logic integrated circuits.

Example 2. We are given a system of bases $p_1 = 15, p_2 = 8, p_3 = 17$ and weights $w_1 = 0, w_2 = 0, w_3 = 1$. The dynamic range $P = 2040$ and $\frac{P}{2} = 1020$.

From the previous example we have the values of P_i, B_i, $C(P)$, $C\left(\frac{P}{2}\right)$ and $C(B_i)$:

$$P_1 = 136, P_2 = 255, P_3 = 120, B_1 = 136, B_2 = 1785, B_3 = 120.$$

$$C(P) = 120, C\left(\frac{P}{2}\right) = 60, C(B_1) = 8, C(B_2)\,105, C(B_3) = 7.$$

For this part we have to find r_X, using equation (10):

$$r_X = \left\lfloor \frac{2 \cdot 136 + 7 \cdot 1785 + 13 \cdot 120}{2040} \right\rfloor = 7.$$

Having these values, you can calculate the sign of the number $X = (2, 7, 13)$, according to the equation (11):

$$C(X) = (2 \cdot 8 + 7 \cdot 105 + 13 \cdot 7) - 7 \cdot 120 = 2.$$

Thus $2 < 8$, so $X = (2, 7, 13)$ is a positive number.

3.3. The Core Function Method Based on the Rank of Number by Chervyakov. There are numerous methods to calculate the rank of the RNS number. Let us discuss them.

In [12], Chervyakov proposed another equation for calculating the rank of a number. Equation (10) includes changes based on the properties of the RNS basis, which eliminate unnecessary operations and derive a rank from the expression.

$$r_X = \left\lfloor \sum_{i=1}^{n} x_i \cdot \frac{\left|P_i^{-1}\right|_{p_i}}{p_i} \right\rfloor. \tag{12}$$

Example 3. We are provided with a set of bases $p_1 = 15, p_2 = 8, p_3 = 17$, along with corresponding weights $w_1 = 0, w_2 = 0, w_3 = 1$. The dynamic range $P = 15 \cdot 8 \cdot 17 = 2040$ and $\frac{P}{2} = \frac{2040}{2} = 1020$.

The values of P_i, B_i, $C(P)$, $C\left(\frac{P}{2}\right)$ and $C(B_i)$ remain the sames:

$$P_1 = 136, P_2 = 255, P_3 = 120, B_1 = 136, B_2 = 1785, B_3 = 120.$$

$$C(P) = 120, C\left(\frac{P}{2}\right) = 60, C(B_1) = 8, C(B_2)\,105, C(B_3) = 7.$$

Let us calculate the value of r_X, using the equation (12):

$$r_X = \left\lfloor 2 \cdot \frac{1}{15} + 7 \cdot \frac{7}{8} + 13 \cdot \frac{1}{17} \right\rfloor = 7.$$

With these values, we can calculate the sign of the number $X = (2, 7, 13)$, according to the equation (11):

$$C(X) = (2 \cdot 8 + 7 \cdot 105 + 13 \cdot 7) - 7 \cdot 120 = 2.$$

Thus $2 < 8$, so $X = (2, 7, 13)$ is a positive number.

Although this equation increases productivity by several percent, it may be more efficient to use approximate methods to obtain a numerical rank.

3.4. The Core Function Method Based on the Approximate Rank of a Number. A function for determining the sign of a number based on a core function with approximate rank is proposed in [6]. The approximate rank calculation improves the efficiency by the use of less strict iterative equations, but the calculation accuracy is sacrificed. Since RNS is an integer numbering scheme, it is possible to maintain accuracy within the boundaries required to maintain correct integer values. However, as RNS is an integer number system, it is possible to maintain accuracy within the necessary limits for preserving the correct integer value.

The coefficient: $N = \log_2 (P \cdot ((\sum_{i=1}^{n} p_i) - n))$ is utilized to derive an estimated rank by connecting the RNS number system and the binary number system through 2^N. With the help of the approximation factor, a correlation can then be established between the number's rank and 2^N.

$$k_i = \left\lfloor \frac{\left|P_i^{-1}\right|_{p_i} 2^N}{p_i} \right\rfloor. \tag{13}$$

By substituting equation (13) into (12), we can obtain a modified equation that finds a number's rank. The accuracy and synergy of Chervyakov's approximate method and Akushsky's method for calculating the core function have decreased. This makes the execution of this equation faster. The rank of a number's approximate value can be found by this equation.

$$r_X = \left\lfloor \sum_{i=1}^{n} \frac{k_i x_i}{2^N} \right\rfloor. \tag{14}$$

the values of 2^N and the value of the approximate coefficients $x_1, x_2, ..., x_n$ can be found at the stage of precomputation, which reduces ther complexity in the execution of the algorithm and make the function faster, compered to other metthods using core function due to an slight decrease in memory consumption.

Example 4. We use all the same moduli $p_1 = 15, p_2 = 8, p_3 = 17$ and weights $w_1 = 0, w_2 = 0, w_3 = 1$. The dynamic range $P = 2040$ and $\frac{P}{2} = 1020$.

The values of P_i, B_i, $C(P)$, $C\left(\frac{P}{2}\right)$ and $C(B_i)$ remain the sames:

$$P_1 = 136, P_2 = 255, P_3 = 120, B_1 = 136, B_2 = 1785, B_3 = 120.$$

$$C(P) = 120, C\left(\frac{P}{2}\right) = 60, C(B_1) = 8, C(B_2)\, 105, C(B_3) = 7.$$

Additionally , we have to find the values of N and k_i:

$$N = \log_2 (2040 \cdot (15 + 8 + 17) - 3) = \log_2 75480,$$

$$k_1 = \left\lfloor \frac{1 \cdot 75480}{15} \right\rfloor = 5032, k_2 = \left\lfloor \frac{7 \cdot 75480}{8} \right\rfloor = 66045,$$

$$k_3 = \left\lfloor \frac{1 \cdot 75480}{17} \right\rfloor = 4440.$$

Let us calculate the value of r_X, using the equation (14):

$$r_X = \left\lfloor \frac{5032 \cdot 2}{75480} + \frac{66045 \cdot 7}{75480} + \frac{4440 \cdot 13}{75480} \right\rfloor = 7.$$

With these values, we can calculate the sign of the number $X = (2, 7, 13)$, according to the equation (11):

$$C(X) = (2 \cdot 8 + 7 \cdot 105 + 13 \cdot 7) - 7 \cdot 120 = 2.$$

Thus $2 < 8$, so $X = (2, 7, 13)$ is a positive number.

4 Minimal Function of the Akushsky Core

Consider the RNS moduli: $\left\{2^{2n} - 1, 2^{n+a}, 2^{2n} + 1\right\}$, where $0 \le a < n$. We define the Akushsky core function for determining the sign of a number as follows:

$$S(X) = w_1 \left\lfloor \frac{X}{2^{2n} - 1} \right\rfloor + w_2 \left\lfloor \frac{X}{2^{n+a}} \right\rfloor + w_3 \left\lfloor \frac{X}{2^{2n} + 1} \right\rfloor. \tag{15}$$

and $S_P = S(P) = w_1 P_1 + w_2 P_2 + w_3 P_3$, where $P = \left(2^{2n} - 1\right) 2^{n+a} \left(2^{2n} + 1\right)$ and $P_1 = 2^{n+a} \left(2^{2n} + 1\right), P_2 = \left(2^{2n} - 1\right) \left(2^{2n} + 1\right), P_3 = (2^{2n} - 1) 2^{n+a}$ and $w_1, w_2, w_3 \in \mathbb{Z}$.

We study what restrictions are imposed on the coefficients w_1, w_2 and w_3 of the core function $S(X)$ so that the function satisfies the condition $\forall X \in [0, P) : 0 \le S(X) \le S_P$. For this purpose, we prove Lemmas 4.1, 4.2, and 4.3.

Lemma 1. *If* $\forall X \in [0, P) : 0 \le S(X) \le S_P$, *then* $\begin{cases} 0 \le w_1 + w_2 \le S_P, \\ 0 \le w_1 + w_2 + w_3 \le S_P. \end{cases}$

Proof. We calculate $S\left(2^{2n} - 1\right), S(P - 1)$ we get:

$$\begin{aligned} S\left(2^{2n} - 1\right) &= w_1 \left\lfloor \frac{2^{2n} - 1}{2^{2n} - 1} \right\rfloor + w_2 \left\lfloor \frac{2^{2n} - 1}{2^{n+a}} \right\rfloor + w_3 \left\lfloor \frac{2^{2n} - 1}{2^{2n} + 1} \right\rfloor \\ &= w_1 + w_2 (2^{n-a} - 1)\ S(P - 1) \\ &= w_1 (P_1 - 1) + w_2 (P_2 - 1) + w_3 (P_3 - 1) \\ &= w_1 P_1 - w_1 + w_2 P_2 - w_2 + w_3 P_3 - w_3 \\ &= w_1 P_1 + w_2 P_2 + w_3 P_3 - w_1 - w_2 - w_3 \\ &= S_P - w_1 - w_2 - w_3. \end{aligned}$$

Since $0 \le S\left(2^{2n} - 1\right) \le S_P$ and $S(P - 1) \le S_P$ hence $0 \le w_1 + w_2 \le S_P$ and $0 \le S_P - w_1 - w_2 - w_3 \le S_P$, so $0 \le w_1 + w_2 \le S_P$ and $0 \le w_1 + w_2 + w_3 \le S_P$. □

Lemma 2. *If* $\forall X \in [0, P) : 0 \le S(X) \le S_P$, *then the following statements are true:*

If $a = 0$ *then* $0 \le w_1 + w_2 \le S_P$ *and* $0 \le w_1 + w_2 + w_3 \le S_P$.

If $a \ge 1$ *then* $0 \le w_1 + w_2 + w_3 \le S_P$ *and* $0 \le w_2 \le S_P$.

Proof. If $a = 0$, then $2^{n+a} = 2^n$. Calculate (2^n) and $S\left(2^{2n}+1\right)$, we get:

$$S\left(2^n\right) = w_1\left\lfloor\frac{2^n}{2^{2n}-1}\right\rfloor + w_2\left\lfloor\frac{2^n}{2^n}\right\rfloor + w_3\left\lfloor\frac{2^n}{2^{2n}+1}\right\rfloor = w_1 + w_2.$$

$$S\left(2^{2n}+1\right) = w_1\left\lfloor\frac{2^{2n}+1}{2^{2n}-1}\right\rfloor + w_2\left\lfloor\frac{2^{2n}+1}{2^n}\right\rfloor + w_3\left\lfloor\frac{2^{2n}+1}{2^{2n}+1}\right\rfloor$$
$$= w_1 + 2^n w_2 + w_3$$

Given that $a \geq 1$ we calculate $S\left(2^{2n}+1\right)$ and $S\left(2^{n+a}\right)$, we get:

$$S\left(2^{2n}+1\right) = w_1\left\lfloor\frac{2^{2n}+1}{2^{2n}-1}\right\rfloor + w_2\left\lfloor\frac{2^{2n}+1}{2^{n+a}}\right\rfloor + w_3\left\lfloor\frac{2^{2n}+1}{2^{2n}+1}\right\rfloor$$
$$= w_1 + 2^{n-a}w_2 + w_3.$$

$$S\left(2^{n+a}\right) = w_1\left\lfloor\frac{2^{n+a}}{2^{2n}-1}\right\rfloor + w_2\left\lfloor\frac{2^{n+a}}{2^{n+a}}\right\rfloor + w_3\left\lfloor\frac{2^{n+a}}{2^{2n}+1}\right\rfloor = w_2.$$

Since $\forall X \in [0,P) : 0 \leq S(X) \leq S_P$ therefore $0 \leq w_1 + w_2 \leq S_P, 0 \leq w_1 + w_2 + w_3 \leq S_P$ if $a = 0$ and $0 \leq w_1 + w_2 + w_3 \leq S_P, 0 \leq w_2 \leq S_P$ if $a \geq 1$. □

Lemma 3. *If* $\forall X \in [0,P) : 0 \leq S(X) \leq S_P$ *and* $S_P = 2^b$ *then* $w_1 \geq 1$.

Proof. Since $S_P = 2^b$, then

$$S_P = w_1P_1 + w_2P_2 + w_3P_3 = 2^b.$$

Assume that $w_3 = 0$, then $\left(2^{2n}-1\right)|2^b$, which is incorrect for all $n \geq 2$. Hence, the assumption is incorrect and $w_3 \neq 0$. It follows from Lemma 4.1 that $0 \leq w_3 \leq S_P$ and $w_1 \neq 0$, so $w_3 \geq 1$. □

Theorem 1. *If* $\forall X \in [0,P) : 0 \leq S(X) \leq S_P < P$ *and* $S_P = 2^b$ *then* $b \geq n+1+a$ *and* $w_1 = 2^{b-n-a-1}, w_2 = 0$ *and* $w_3 = -2^{b-n-a-1}$.

Proof. It follows from the condition of the theorem that the equality is satisfied:

$$w_1\left\lfloor\frac{P}{2^{2n}-1}\right\rfloor + w_2\left\lfloor\frac{P}{2^{n+a}}\right\rfloor + w_3\left\lfloor\frac{P}{2^{2n}+1}\right\rfloor = 2^b,$$

Hence, $w_1P_1 + w_2P_2 + w_3P_3 = 2^b$.

Case 1. If $a = 0$ then S_P can be represented in the following form:

$$S_P = P_3(w_1 + w_2 + w_3) + (P_2 - P_3)(w_1 + w_2) + (P_1 - P_2)(w_1)$$
$$= (2^{2n}-1)2^n(w_1 + w_2 + w_3) + \left(2^{2n}-1\right)(w_1 + w_2) + (2^n+1)w_1$$

It follows from Lemmas 4.1, 4.2 and 4.3 that: $w_1 + w_2 + w_3 \geq 0, w_2 \geq 0$ and $w_1 \geq 1$, so: $S_P \geq 2^n + 1$ and $b \geq n$. Note that

$$\begin{cases} w_1 \equiv 2^{b-n-1} \pmod{2^{2n}-1}, \\ w_2 \equiv 0 \pmod{2^n}, \\ w_3 \equiv -2^{b-n-1} \pmod{2^{2n}+1}. \end{cases}$$

Note that when $b = n+1$ and $w_1 = 1, w_2 = 0$ and $w_3 = -1$, the conditions of Lemmas 4.1, 4.2 and 4.3 are satisfied.

$$S_P = \left\lfloor \frac{\left(2^{2n}-1\right)2^n\left(2^{2n}+1\right)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{\left(2^{2n}-1\right)2^n\left(2^{2n}+1\right)}{2^{2n}+1} \right\rfloor$$

$$= 2^n\left(2^{2n}+1\right) - \left(2^{2n}-1\right)2^n = 2^{n+1}$$

Therefore, $b = n+1$.

Case 2. If $0 \le a < n$ then S_P can be represented in the following form:

$$S_P = P_2\left(w_1+w_2+w_3\right) + \left(P_3-P_2\right)\left(w_1+w_3\right) + \left(P_1-P_3\right)\left(w_1\right)$$

$$= \left(2^{2n}-1\right)\left(2^{2n}+1\right)\left(w_1+w_2+w_3\right)$$

$$+ \left(2^{2n}-1\right)\left(2^{n+a}-2^{2n}-1\right)\left(w_1+w_3\right) + 2^{n+a+1}w_1$$

It follows from Lemmas 4.1, 4.2 and 4.3 that: $w_1+w_2+w_3 \ge 0$, $w_1+w_2 \ge 0$ and $w_1 \ge 1$, hence:

$$S_P \ge 2^{n+a+1} > 2^{n+a},$$

Hence, $b > n+a$.

Note that

$$\begin{cases} w_1 \equiv 2^{b-n-a-1} \pmod{2^{2n}-1}, \\ w_2 \equiv 0 \pmod{2^{n+a}}, \\ w_3 \equiv -2^{b-n-a-1} \pmod{2^{2n}+1}. \end{cases}$$

Note also that when $b = n+a+1, w_1 = 1, w_2 = 0$ and $w_3 = -1$, the conditions of Lemmas 4.1, 4.2 and 4.3 are satisfied.

$$S_P = \left\lfloor \frac{\left(2^{2n}-1\right)2^{n+a}\left(2^{2n}+1\right)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{\left(2^{2n}-1\right)2^{n+a}\left(2^{2n}+1\right)}{2^{2n}+1} \right\rfloor$$

$$= 2^{n+a}\left(2^{2n}+1\right) - \left(2^{2n}-1\right)2^{n+a} = 2^{n+a+1}$$

Hence, $b = n+a+1$. □

Let us consider a minimal core function satisfying Theorem 1 and show that it can be used to efficiently implement the number sign function in RNS.

The function of number sign detection will take the form:

$$S(X) = \left\lfloor \frac{X}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor$$

$$P = \left(2^{2n}-1\right)2^{n+a}\left(2^{2n}+1\right)$$

$$S(P) = 2^{n+a+1}$$

$$\frac{P}{2} = \left(2^{2n}-1\right)2^{n+a-1}\left(2^{2n}+1\right)$$

Lemma 4. $\forall X \in [0, P) : S(X) \ge 0$.

Proof. Since $\forall n : 2^{2n} - 1 < 2^{2n} + 1$, then

$$\frac{X}{2^{2n}-1} > \frac{X}{2^{2n}+1},$$

hence

$$\left\lfloor \frac{X}{2^{2n}-1} \right\rfloor \geq \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor,$$

which means $\forall X \in [0, P) : S(X) \geq 0$. □

Lemma 5. $\forall X \in [0, P) : S(X) \leq 2^{n+a+1}$.

Proof. Consider the function $G(P - X) = 2^{n+a+1} - S(X)$, we have:

$$G(P-X) = \left\lceil \frac{P-X}{2^{2n}-1} \right\rceil - \left\lceil \frac{P-X}{2^{2n}+1} \right\rceil,$$

Suppose $P - X = Y$, then $Y \in (0, P]$ and the function $G(Y)$ takes the form:

$$G(Y) = \left\lceil \frac{Y}{2^{2n}-1} \right\rceil - \left\lceil \frac{Y}{2^{2n}+1} \right\rceil,$$

Since $\forall n : 2^{2n} - 1 < 2^{2n} + 1$, then

$$\frac{Y}{2^{2n}-1} > \frac{Y}{2^{2n}+1},$$

consequently

$$\left\lceil \frac{Y}{2^{2n}-1} \right\rceil \geq \left\lceil \frac{Y}{2^{2n}+1} \right\rceil,$$

then $\forall Y \in (0, P] : G(Y) \geq 0$.

□

Theorem 2. $\forall X \in [0, P)$ *the following conditions are satisfied:*

1. *If* $S(X) > 2^{n+a}$, *then* $X > \left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right)$.
2. *If* $S(X) < 2^{n+a}$, *then* $X < \left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right)$.

Proof. Case 1. If $X \geq \frac{P}{2} + 2^{2n} - 1$ then $S(X) \geq 2^{n+a}$). Consider numbers of the form $Y = \left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) + 2^{2n} - 1 + X$, where $0 \leq X < \frac{P}{2} - 2^{2n} + 1$ we get:

$$\begin{aligned} S(Y) &= \left\lfloor \frac{\left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) + 2^{2n} - 1 + X}{2^{2n}-1} \right\rfloor \\ &\quad - \left\lfloor \frac{\left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) + 2^{2n} - 1 + X}{2^{2n}+1} \right\rfloor \\ &= 2^{n+a} + 1 + \left\lfloor \frac{X}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{X + 2^{2n} - 1}{2^{2n}+1} \right\rfloor \\ &= 2^{n+a} + 1 + \left\lfloor \frac{X}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor + \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor - \left\lfloor \frac{X + 2^{2n} - 1}{2^{2n}+1} \right\rfloor \end{aligned}$$

$$= 2^{n+a} + 1 + S(X) + \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor - \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor.$$

If $\left\lfloor \frac{X}{2^{2n}+1} \right\rfloor = \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor$, then $S(Y) = 2^{n+a} + 1 + S(X)$. It follows from Lemma 3.4 that $S(X) \geq 0$, hence: $S(Y) > 2^{n+a}$.

If $\left\lfloor \frac{X}{2^{2n}+1} \right\rfloor = \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor - 1$, then $S(Y) = 2^{n+a} + S(X), \quad S(X) \geq 0$, hence: $S(Y) \geq 2^{n+a}$.

Case 2. If $X \leq \frac{P}{2} - 2^{2n} + 1$ then $S(X) \leq 2^{n+a})$. Consider numbers of the form $G = \left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) - 2^{2n} + 1 - X$, where $0 \leq X \leq \frac{P}{2} - 2^{2n} + 1$ we get:

$$S(G) = \left\lfloor \frac{\left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) - 2^{2n} + 1 - X}{2^{2n}-1} \right\rfloor$$

$$- \left\lfloor \frac{\left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) - 2^{2n} + 1 - X}{2^{2n}+1} \right\rfloor 2^{n+a} - 1$$

$$- \left\lfloor \frac{X}{2^{2n}-1} \right\rfloor + \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor$$

$$= 2^{n+a} - 1 - \left\lfloor \frac{X}{2^{2n}-1} \right\rfloor + \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor - \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor + \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor$$

$$= 2^{n+a} - 1 - S(X) - \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor + \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor.$$

If $\left\lfloor \frac{X}{2^{2n}+1} \right\rfloor = \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor$, then $S(G) = 2^{n+a} - 1$, It follows from Lemma 4.4 that $S(X) \leq 0$, hence: $S(G) < 2^{n+a}$.

If $\left\lfloor \frac{X}{2^{2n}+1} \right\rfloor = \left\lfloor \frac{X+2^{2n}-1}{2^{2n}+1} \right\rfloor - 1$, then $S(G) = 2^{n+a}$.

Case 3. If $\frac{P}{2} - 2^{2n} + 1 < X < \frac{P}{2} + 2^{2n} - 1$ then $S(X) = 2^{n+a})$. Consider numbers of the form $U = \left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) + X$, we get:

$$S(U) = \left\lfloor \frac{\left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) + X}{2^{2n}-1} \right\rfloor$$

$$- \left\lfloor \frac{\left(2^{2n}-1\right) 2^{n+a-1} \left(2^{2n}+1\right) + X}{2^{2n}+1} \right\rfloor = 2^{n+a} + \left\lfloor \frac{X}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{X}{2^{2n}+1} \right\rfloor.$$

If $0 \leq X < 2^{2n} - 1$, then $\left\lfloor \frac{X}{2^{2n}-1} \right\rfloor = 0$ and $\left\lfloor \frac{X}{2^{2n}+1} \right\rfloor = 0$, hence $S(U) = 2^{n+a}$.

If $-2^{2n} + 1 < X < 0$, then $\left\lfloor \frac{X}{2^{2n}-1} \right\rfloor = -1$ and $\left\lfloor \frac{X}{2^{2n}+1} \right\rfloor = -1$, hence $S(U) = 2^{n+a}$. □

Theorem 3. *To calculate the core function, we must demonstrate the subsequent statement: If* $0 \leq S(X) < S_P$, *then* $S(X) = \left|\sum_{i=1}^{n} k_i x_i\right|_{S_P}$, *else* $S(X) = S_P$.

The function $S(X)$ can be calculated using the following equation:

$$S(X) = \left| \sum_{i=1}^{n} k_i x_i \right|_{S_P}. \tag{16}$$

where $k_i = S\left(\left|P_i^{-1}\right|_{p_i} P_i\right), P_i = \frac{P}{p_i}, S_P = S(P)$ and $P = \prod_{i=1}^{n} p_i$.

Let us calculate the constants

$$S_P = S(P) = \left(2^{2n}+1\right)2^{n+a} - \left(2^{2n}-1\right)2^{n+a}$$

$$= \left(\left(2^{2n}+1\right) - \left(2^{2n}-1\right)\right)2^{n+a} = (2^{2n}+1-2^{2n}+1)2^{n+a} = 2^{n+a+1}$$

$$P_1 = 2^{n+a}(2^{2n}+1)$$

$$P_2 = (2^{2n}-1)(2^{2n}+1)$$

$$P_3 = (2^{2n}-1)\ 2^{n+a}$$

$$\left|P_1^{-1}\right|_{p_1} = \left|\frac{1}{2^{n+a}(2^{2n}+1)}\right|_{2^{2n}-1} = \left|\frac{1}{2^{a+1}}\right|_{2^{2n}-1} = \left|2^{n+a+1}\right|_{2^{2n}-1}$$

$$= \begin{cases} 2^{n-1}\ if\ a=0, \\ 2^{n-a-1}\ if\ 1 \le a < n. \end{cases}$$

$$\left|P_2^{-1}\right|_{p_2} = \left|\frac{1}{(2^{2n}-1)(2^{2n}+1)}\right|_{2^{n+a}} = \left|-\frac{1}{1}\right|_{2^{n+a}} = \left|2^{n+a}-1\right|_{2^{n+a}}$$

$$= \begin{cases} 2^{n}-1\ if\ a=0, \\ 2^{n+a}-1\ if\ 1 \le a < n. \end{cases}$$

$$\left|P_3^{-1}\right|_{p_3} = \left|\frac{1}{(2^{2n}-1)\,2^{n+a}}\right|_{2^{2n}+1} = \left|2^{n+1}+2^{n}+1\right|_{2^{2n}+1}$$

$$= \begin{cases} 2^{n-1}\ if\ a=0, \\ 2^{n-a-1}\ if\ 1 \le a < n. \end{cases}$$

If $a = 0$, then $\left|P_1^{-1}\right|_{p_1} = 2^{n-1}, \left|P_2^{-1}\right|_{p_2} = 2^{n}-1, \left|P_3^{-1}\right|_{p_3} = 2^{n-1}$, consequently,

$$k_1 = S\left(\left|P_1^{-1}\right|_{p_1} P_1\right)$$

$$= \left\lfloor \frac{2^{n-1}2^{n}(2^{2n}+1)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{2^{n-1}2^{n}\left(2^{2n}+1\right)}{2^{2n}+1} \right\rfloor = 2^{2n-1}+1-2^{n-1}2^{n} = 1,$$

$$k_2 = S\left(\left|P_2^{-1}\right|_{p_2} P_2\right)$$

$$= \left\lfloor \frac{(2^{n}-1)\left(2^{2n}-1\right)\left(2^{2n}+1\right)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{(2^{n}-1)\left(2^{2n}-1\right)\left(2^{2n}+1\right)}{2^{2n}+1} \right\rfloor$$

$$= (2^{n}-1)\left(2^{2n}+1\right) - (2^{n}-1)\left(2^{2n}-1\right) = 2\left(2^{n}-1\right) = 2^{n+1}-2$$

$$k_3 = S\left(\left|P_3^{-1}\right|_{p_3} P_3\right) = \left\lfloor \frac{2^{n-1}2^{n}(2^{2n}-1)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{2^{n-1}2^{n}(2^{2n}-1)}{2^{2n}+1} \right\rfloor$$

$$= 2^{n-1}2^n - \left(2^{2n-1} - 1\right) = 2^{2n-1} - 2^{2n-1} + 1 = 1.$$

If $a \geq 1$, then $\left|P_1^{-1}\right|_{p_1} = 2^{n-a-1}, \left|P_2^{-1}\right|_{p_2} = 2^{n+a} - 1, \left|P_3^{-1}\right|_{p_3} = 2^{n-a-1}$, consequently,

$$k_1 = S\left(\left|P_1^{-1}\right|_{p_1} P_1\right) = \left\lfloor \frac{2^{n-a-1}2^{n+a}(2^{2n}+1)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{2^{n-a-1}2^{n+a}\left(2^{2n}+1\right)}{2^{2n}+1} \right\rfloor$$

$$= 2^{2n-1} + 1 - 2^{n-1}2^n = 1$$

$$k_2 = S\left(\left|P_2^{-1}\right|_{p_2} P_2\right),$$

$$= \left\lfloor \frac{\left(2^{n+a}-1\right)\left(2^{2n}-1\right)\left(2^{2n}+1\right)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{\left(2^{n+a}-1\right)\left(2^{2n}-1\right)\left(2^{2n}+1\right)}{2^{2n}+1} \right\rfloor$$

$$= \left(2^{n+a}-1\right)\left(2^{2n}+1\right) - \left(2^{n+a}-1\right)\left(2^{2n}-1\right) = 2\left(2^{n+a}-1\right) = 2^{n+a+1} - 2$$

$$k_3 = S\left(\left|P_3^{-1}\right|_{p_3} P_3\right) = \left\lfloor \frac{2^{n-a-1}2^{n+a}(2^{2n}-1)}{2^{2n}-1} \right\rfloor - \left\lfloor \frac{2^{n-a-1}2^{n+a}(2^{2n}-1)}{2^{2n}+1} \right\rfloor$$

$$= 2^{n-a-1}2^{n+a} - \left(2^{2n-1} - 1\right) = 2^{2n-1} - 2^{2n-1} + 1 = 1$$

Then a full-fledged sign detection function, based on the minimum core function for the set of moduli $\left\{2^{2n}-1, 2^{n+a}, 2^{2n}+1\right\}$, takes the following form:

$$Sign\left(X\right) = \begin{cases} 1 & if\ S\left(X\right) = \left|\sum_{i=1}^{n} k_i x_i\right|_{S_P} < S\left(\frac{P}{2}\right) \\ 0 & if\ S\left(X\right) = \left|\sum_{i=1}^{n} k_i x_i\right|_{S_P} \geq S\left(\frac{P}{2}\right) \end{cases} \tag{17}$$

Example 5. For $n = 2$ and $a = 1$ bases $p_1 = 15, p_2 = 8, p_3 = 17$ the volume of the dynamic range $P = 2040$ and $\frac{P}{2} = 1020$. The values of k_i are: $k_1 = 1, k_2 = 2^4 - 2 = 14, k_3 = 1$.

Let's find the values of $S\left(P\right)$ and $S\left(\frac{P}{2}\right)$:

$$S\left(P\right) = \left\lfloor \frac{2040}{15} \right\rfloor - \left\lfloor \frac{2040}{17} \right\rfloor = 16,$$

$$S\left(\frac{P}{2}\right) = \left\lfloor \frac{1020}{15} \right\rfloor - \left\lfloor \frac{1020}{17} \right\rfloor = 8.$$

Having these values, you can calculate the sign of the number $X = (2, 7, 13)$, according to the equation (16):

$$S\left(X\right) = \left|1 \cdot 2 + 14 \cdot 7 + 1 \cdot 13\right|_{16} = \left|113\right|_{16} = 1.$$

Thus $1 < 8$, so $X = (2, 7, 13)$ is a positive number.

In the next section, a performance study of the considered methods will be conducted in order to confirm the relevance of the obtained result in the study.

5 Performance Evaluation

The algorithm proposed in Section 4 presents an advantage over other methods discussed in Section 3.

To support this statement, we performed a comparison of all algorithm performances in the Python language. The experiments were conducted on a computer operating the Windows 10 Pro Edition system, with a zen 5 3500X 3.6 GHz processor, 16 GB DDR4 3200 MHz RAM, and a 256 Go NVMe SSD.

The times taken to execute each sign detection algorithm were measured in nanoseconds. The RNS sets that cover 8, 16, 24, and 32 bits, as specified in Table 5.1, were chosen for modeling purposes. The outcomes of the modeling procedure are condensed in Table 5.2.

Таблица 1. Sets of moduli for modelling

Algorithm	Dynamic range size, bits			
	8	16	24	32
Core Function	{3, 5, 7, 11}	{13, 17, 19, 23}	{59, 61, 67, 71}	{251, 257, 263, 269}
Rank Core	{3, 5, 7, 11}	{13, 17, 19, 29}	{59, 61, 67, 71}	{251, 257, 263, 271}
Rank Chervyakov Core	{3, 5, 7, 11}	{13, 17, 19, 23}	{53, 61, 67, 73}	{241, 257, 263, 277}
Approximate Rank Core	{3, 5, 7, 11}	{13, 17, 19, 29}	{59, 61, 67, 71}	{251, 257, 263, 269}
Minimal Core	{15, 4, 17}	{63, 16, 65}	{255, 128, 257}	{4095, 256, 4097}

Таблица 2. The result of the simulation, ns

Algorithm	Dynamic range size, bits			
	8	16	24	32
Core Function	35600	35900	36100	36800
Rank Core	35100	36300	37100	38100
Rank Chervyakov Core	47900	53200	53600	57100
Approximate Rank Core	34700	35400	36300	38700
Minimal Core	3700	3800	4000	4200

The table shows that our proposed algorithm is 88.04 % faster on average. The reduced computational burden leads to energy conservation and lower power consumption, making it a desirable choice for energy-efficient computing platforms and low-power embedded systems. Additionally, the algorithm's efficiency could potentially reduce hardware requirements, thus improving its suitability for resource-limited environments.

6 Conclusion

We have developed and implemented a high-speed algorithm to detect number signs in the RNS. This method utilizes the Akushsky core function

for a special set of moduli $\{2^{2n}-1, 2^{n+a}, 2^{2n}+1\}$. We conducted experiments and showed that our method improves the performance of number sign detection operation by 88.04 % on average. These results may be useful in tasks related to cryptography, number theory and other areas where RNS is applied.

Although this article focuses on the theoretical and algorithmic aspects, future research should investigate practical applications of the method and its integration into real-world systems. Furthermore, exploring the optimisation potential for a wider range of RNS sets remains a valuable area of research.

References

[1] E. Shiryaev, E. Golimblevskaia, M. Babenko, A. Tchernykh, B. Pulido-Gaytan, *Improvement of the Approximate Method for the Comparison Operation in the RNS*, 2020 International Conference Engineering and Telecommunication (En&T), (2020), 1–6. IEEE.

[2] K. Isupov, *High-Performance Computation in Residue Number System Using Floating-Point Arithmetic*, Computation, **9**(2), (2021), 9.

[3] L. Deng, W. Liu, D. Li, B. O. Mohammed, *A new sign detection design for the residue number system based on quantum-dot cellular automata*, Photonic Network Communications, **42**(1), (2021), 70–80.

[4] L. Sousa, P. Martins, *Sign Detection and Number Comparison on RNS 3-Moduli Sets* $\{2^n-1, 2^{n+x}, 2^n+1\}$, Circuits, Systems, and Signal Processing, **36**(3), (2017), 1224–1246.

[5] M. Akkal, P. Siy, *Optimum RNS sign detection algorithm using MRC-II with special moduli set*, Journal of Systems Architecture, **54**(10), (2008), 911–918.

[6] E. Shiriaev, N. Kucherov, M. Babenko, A. Nazarov, *Fast Operation of Determining the Sign of a Number in RNS Using the Akushsky Core Function*, Computation, **11**(7), (2023), 124.

[7] M. Babenko, M. Deryabin, S. J. Piestrak, P. Patronik, N. Chervyakov, A. Tchernykh, A. Avetisyan, *RNS number comparator based on a modified diagonal function*, Electronics, **9**(11), (2020), 1784.

[8] W. S. Brown, J. F. Traub, *On Euclid's Algorithm and the Theory of Subresultants*, Journal of the ACM, **18**(4), (1971), 505–514.

[9] I.Y. Akushsky, V.M. Burtsev, I.T. Pak, *About the New Positional Characteristic of the Non-Positional Code and Its Application*, In Theory of Coding and Optimization of Complex Systems, **1** (1977), 8–16.

[10] I.Y. Akushsky, V.M. Burtsev, I.T. Pak, *Calculation of the Positional Characteristic (Core) of the Non-Positional Code*, In Theory of Coding and Optimization of Complex Systems, (1977), 17–25.

[11] E. Shiriaev, N. Kucherov, M. Babenko, V. Lutsenko, S. Al-Galda, *Algorithm for Determining the Optimal Weights for the Akushsky Core Function with an Approximate Rank*, Applied Sciences, **13**(18), (2023), 10495.

[12] N. I. Chervyakov, A. S. Molahosseini, P. A. Lyakhov, M. G. Babenko, M. A. Deryabin, *Residue-to-binary conversion for general moduli sets based on approximate Chinese remainder theorem*, International journal of computer mathematics, **94**(9), (2017), 1833–1849.

[13] H. L. Garner, *The residue number system*, western joint computer conference, (1959), 146–153.

[14] P. Albicocco, G. C. Cardarilli, A. Nannarelli, M. Re, *Twenty years of research on RNS for DSP: Lessons learned and future perspectives*, (2014), 436–439. IEEE
[15] N. M. Yassine, *Matrix mixed-radix conversion for RNS arithmetic architectures*, Proceedings of the 34th Midwest Symposium on Circuits and Systems, (1992), 273–278.
[16] C. H. Vun, A. B. Premkumar, W. Zhang, *A new RNS based DA approach for inner product computation*, IEEE Transactions on Circuits and Systems I: Regular Papers, **60**(8), (2013), 2139–2152.
[17] A. Nazarov, M. Babenko, E. Golimblevskaia, *Hardware implementation of the reverse conversion RNS-WNS on FPGA*, 2020 International Conference Engineering and Telecommunication (En&T), (2020), 1–5.

Vladislav Vyacheslavovich Lutsenko
North-Caucasus Federal University,
Pushkin st., 1,
355017, Stavropol, Russia
Email address: vvlutcenko@ncfu.ru

Mikhail Vasilievich Zgonnikov
Lycee du Dauphine,
Bd Remy Roure, 38,
26100, Romans-sur-Isere, France
Email address: zgonnikovmichka@gmail.com

Mikhail Grigoryevich Babenko
North-Caucasus Federal University,
Pushkin st., 1,
355017, Stavropol, Russia
Email address: mgbabenko@ncfu.ru