

УДК 004.27

DOI 10.17513/snt.40438

МЕТОД БЫСТРОГО ОПРЕДЕЛЕНИЯ ЗНАКА ЧИСЛА В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ

Ляхов П.А.

ФГАОУ ВО «Северо-Кавказский федеральный университет», Ставрополь,

e-mail: ljahov@mail.ru

Задача на определение знака числа является одной из проблемных в системе остаточных классов. Эта операция играет фундаментальную роль, поскольку лежит в основе других сложных для реализации операций, таких как сравнение чисел и деление. Традиционные алгоритмы основываются на вычислениях с определенными, специально подобранными наборами модулей, что делает такие алгоритмы эффективными только для ограниченного круга задач. Цель данной работы состоит в разработке нового алгоритма определения знака числа в системе остаточных классов. Новизна представленного алгоритма заключается в использовании дробных значений в модифицированной версии Китайской теоремы об остатках, что в свою очередь обеспечивает универсальность алгоритма и его применимость в системах остаточных классов любого типа. Разработанный подход позволяет эффективно определять знак числа в системе остаточных классов, не накладывая ограничений на выбор конкретного набора модулей. Программное моделирование показало увеличение быстродействия работы разработанного алгоритма в сравнении с известным методом на основе быстрого преобразования в обобщенной позиционной системе счисления. Полученные результаты могут эффективно использоваться в различных системах цифровой обработки сигналов и задачах машинного обучения.

Ключевые слова: система остаточных классов, немодульные операции, определение знака числа, набор модулей, дробные величины, динамический диапазон

METHOD OF RAPID DETERMINATION OF THE SIGN OF A NUMBER IN THE RESIDUE NUMBER SYSTEM

Lyakhov P.A.

North Caucasian Federal University, Stavropol, e-mail: ljahov@mail.ru

The problem of determining the sign of a number is one of the problematic ones in the system of residual classes. This operation plays a fundamental role, since it underlies other operations that are difficult to implement, such as number comparison and division. Traditional algorithms are based on calculations with certain, specially selected sets of modules, which makes such algorithms effective only for a limited range of problems. The purpose of this work is to develop a new algorithm for determining the sign of a number in the system of residual classes. The novelty of the presented algorithm lies in the use of fractional values in a modified version of the Chinese remainder theorem, which in turn ensures the universality of the algorithm and its applicability to systems of residual classes of any type. The developed approach allows one to effectively determine the sign of a number in the system of residual classes without imposing restrictions on the choice of a specific set of modules. Software modeling showed an increase in the performance of the developed algorithm in comparison with the known method based on fast transformation in the generalized positional number system. The obtained results can be effectively used in various digital signal processing systems and machine learning problems.

Keywords: residue number system, non-modular operations, determining the sign of a number, set of modules, fractional values, dynamic range

Введение

Система остаточных классов (СОК) является альтернативой двоичной системе счисления и представляет собой перспективный способ ускорения арифметических вычислений. Благодаря возможности параллельной обработки чисел с небольшой разрядностью СОК существенно уменьшает время выполнения операций сложения, вычитания и умножения [1]. Эти преимущества делают СОК особенно привлекательной для задач, в которых преобладают модульные арифметические вычисления [2]. Широкое применение данный подход находит в цифровой обработке сигналов [3; 4]. Так, на основе вычислений в СОК решается ряд задач по цифровой фильтрации видеосигна-

лов [5] и интеллектуальному анализу визуальных данных [6; 7], разработке отказоустойчивых модульных нейрокомпьютеров [8]. СОК применяют в том числе для различных задач в области криптографии [9; 10]. Облачные технологии [11] и беспроводные сенсорные сети [12] – это области, которые требуют наличия высокопроизводительных и надежных решений на основе СОК. В настоящее время активно разрабатываются подобные решения для спутниковой связи [13; 14], вычислений на основе ДНК [15] и во многих других направлениях.

Тем не менее СОК имеет и определенные недостатки. Ключевая трудность заключается в реализации немодульных операций, таких как определение знака

числа [16], сравнение чисел [17], обнаружение переполнения, деление, извлечение корня и некоторые другие. Поскольку эти операции не относятся к модульным, поиск эффективных методов их выполнения остаётся актуальной задачей. В последние годы было опубликовано множество работ, направленных на совершенствование алгоритмов выполнения таких операций в СОК для конкретных наборов модулей, например вида $2^n \pm 1$, $2n \pm 1$ и т. д. [18; 19]. Современное развитие теории СОК отличается множеством узкоспециализированных исследований, сосредоточенных на отдельных наборах модулей, которые зачастую не пересекаются между собой. Это приводит к тому, что модульный набор, эффективный для одной задачи, может оказаться совершенно неприменимым для другой, что подробно представлено в работе авторов Karlun D. и др. [20]. Такое положение дел значительно усложняет практическое использование СОК из-за разнообразия и несогласованности существующих решений, а также отсутствия универсальных и эффективных подходов. Исследователями Younes и Steffan была предпринята попытка обобщить существующие подходы к выполнению операций сложения и умножения в различных реализациях СОК [21].

Одной из основных проблемных операций в СОК является определение знака числа. Эта операция играет фундаментальную роль, поскольку служит основой других сложных в СОК операций, таких как сравнение чисел и деление. В настоящее время разработаны эффективные алгоритмы определения знака, однако они применимы лишь к СОК с определёнными, специально подобранными наборами модулей. Авторами Xu, Bian, Yao предложен алгоритм определения знака для СОК вида $\{2^{n+1}-1, 2^n-1, 2^n\}$, однако скорость проведения операции представляется недостаточной [22]. Tomczak разработал новый алгоритм определения знака в СОК вида $\{2^n-1, 2^n, 2^{n+1}\}$, требующий больших аппаратных затрат [23]. Mohan и Phalguna также оценивают данный набор модулей для двух методов получения чисел со смешанным основанием счисления, что доказывает необходимость высоких аппаратных затрат [24]. Setiaarif и Siy предложили алгоритм определения знака, требующий построения модулей СОК по заданным условиям [25]. Kumar и Chang разработали новый подход к решению задачи сравнения чисел в СОК для специализированного набора из пяти оснований $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1, 2^{n-1}-1\}$, что позволяет повысить энергоэффективность, однако скорость вычислений недо-

статочна [26]. Особенность подхода заключается в том, что знаки операндов при сравнении, а также их разность определяются после масштабирования на коэффициент $(2^{2n}-1)(2^{n-1}-1)$, что позволяет уменьшить размер сумматоров по модулю с $5n$ бит до n и $n+1$ бит. Однако все известные работы по данной проблеме имеют общий недостаток, заключающийся в ограниченности применения на практике. Для устранения этой проблемы необходимо разработать новые методы и алгоритмы определения знака числа в СОК, пригодные для произвольных конфигураций модулей, что обосновывает выбор цели исследования.

Целью исследования является разработка нового алгоритма определения знака числа в СОК на основе Китайской теоремы об остатках с вычислениями над дробными величинами.

Материалы и методы исследования

В данном исследовании применен новый подход к определению знака числа в СОК на основе преобразования метода вычислений по Китайской теореме об остатках над кольцом целых чисел к вычислениям над дробными числами. Такой подход привел к увеличению разрядности обрабатываемых чисел, однако позволил избежать вычислительно сложной операции деления чисел. С использованием классических математических методов была доказана теорема, подтверждающая корректность выполнения операции определения знака числа в СОК на основе предложенного алгоритма. Для оценки эффективности алгоритма была проведена его программная реализация на языке Python с моделированием в различных СОК, имеющих динамические диапазоны от 256 до 2048 бит включительно. Кроме того, было выполнено сравнение предложенной разработки с известным алгоритмом на основе обобщенной позиционной системы счисления.

Результаты исследования и их обсуждение

СОК задаётся множеством положительных взаимно простых между собой чисел $\{m_1, m_2, \dots, m_n\}$, называемых модулями. Произведение этих модулей $M = m_1 m_2 \dots m_n$ определяет диапазон СОК. В рамках такой системы любое целое число X , которое находится в интервале $0 \leq X < M$, можно однозначно представить в виде набора остатков $(x_1, x_2, \dots, x_n)$, где x_i вычислен как остаток от деления X на соответствующий модуль m_i .

$$x_i = X \bmod m_i = |X|_{m_i}, \quad 0 \leq x_i < m_i. \quad (1)$$

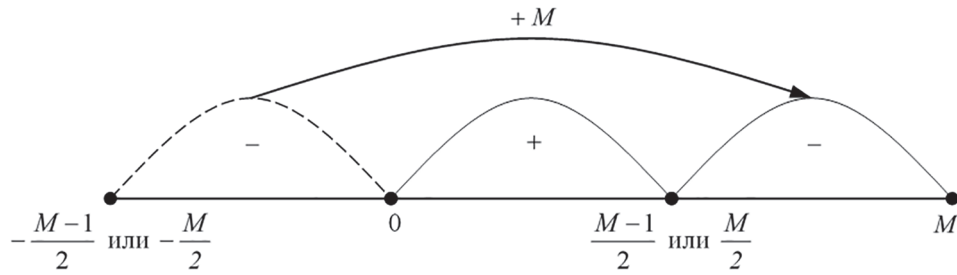


Рис. 1. Визуализация расположения положительных и отрицательных чисел в СОК

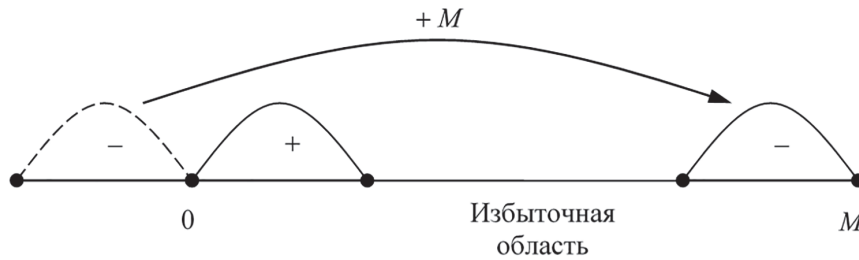


Рис. 2. Визуализация расположения положительных и отрицательных чисел в избыточной СОК

Положительные и отрицательные числа в СОК представляются за счет условного деления диапазона M на две примерно равные части, при этом одна часть соответствует положительным числам, другая – отрицательным. Знак числа X определяется тем, в какую из данных частей оно попадает:

$$\begin{aligned} -\frac{M-1}{2} \leq X \leq \frac{M-1}{2}, & \text{ если } M \text{ нечетное,} \\ -\frac{M}{2} \leq X \leq \frac{M}{2}, & \text{ если } M \text{ четное.} \end{aligned} \quad (2)$$

Это условие определяет однозначность представления числа X в СОК. На рисунке 1 представлено распределение положительных и отрицательных чисел в системе остаточных классов на числовой оси. Отрицательные значения, входящие в соответствующие диапазоны (2), смещены вправо на M единиц, что обеспечивается циклической природой представления чисел в СОК на основе алгебры колец классов вычетов.

Арифметические операции сложения, вычитания и умножения в СОК реализуются по следующим правилам:

$$X \pm Y = \left(|x_1 \pm y_1|_{m_1}, |x_2 \pm y_2|_{m_2}, \dots, |x_n \pm y_n|_{m_n} \right) \quad (3)$$

$$X \cdot Y = \left(|x_1 \cdot y_1|_{m_1}, |x_2 \cdot y_2|_{m_2}, \dots, |x_n \cdot y_n|_{m_n} \right) \quad (4)$$

В формулах (3) и (4) проявляется главное достоинство СОК – указанные операции выполняются параллельно по каждому из модулей.

При разработке отказоустойчивых систем применяется избыточная система остаточных классов (RRNS). В такой системе модули делятся на информационные и дополнительные (избыточные). Благодаря включению избыточных модулей полный диапазон системы становится больше,

чем диапазон её информационной части. Разряды СОК по избыточным модулям на практике используются для обнаружения и исправления ошибок вычислительной системы. На рисунке 2 продемонстрировано расположение положительных и отрицательных чисел в избыточной СОК.

Целью данной статьи является разработка эффективных алгоритмов для определения знака числа, заданного в виде набора остатков $(x_1, x_2, \dots, x_n)$. В дальнейшем будет

представлен новый метод, позволяющий проверять, принадлежит ли число интервалу положительных или отрицательных значений в СОК, как это показано на рисунках 1 и 2. Математической основой вычисления позиционной характеристики числа в СОК является его преобразование в позиционную форму представления. Преобразование осуществляется на базе Китайской теоремы об остатках (КТО, CRT), что обеспечивает однозначное восстановление значения по его представлению в вычетах. Пусть в СОК с модулями $m_1, m_2, \dots, m_n$ задано число X , удовлетворяющее неравенству $0 \leq X < M = m_1 m_2 \dots m_n$ и представленное в виде набора остатков $X = (x_1, x_2, \dots, x_n)$. Восстановление исходного числа X по этим остаткам выполняется согласно выражению (6):

$$X = \left| \sum_{i=1}^n x_i \left| M_i^{-1} \right|_{m_i} M_i \right|_M, \quad (5)$$

где $M_i = M / m_i$, а $\left| M_i^{-1} \right|_{m_i}$ это мультипликативная обратная величина для числа M_i^{-1} по модулю m_i .

Разделив обе части равенства (5) на M , получим:

$$\frac{X}{M} = \left| \sum_{i=1}^n x_i \frac{\left| M_i^{-1} \right|_{m_i} M_i}{M} \right|_1, \quad (6)$$

в котором $| \cdot |_1$ означает дробную часть числа.

Положив $k_i = \frac{\left| M_i^{-1} \right|_{m_i} M_i}{M}$, тогда преобразуем выражение (6) в компактную форму:

$$\frac{X}{M} = \left| \sum_{i=1}^n x_i k_i \right|_1. \quad (7)$$

Все значения $\frac{X}{M}, k_1, k_2, \dots, k_n$ в формуле (7), которые представляют собой бесконечные периодические дроби, лежат в интервале $[0, 1)$. Коэффициенты k_i зависят только от выбранных модулей СОК и, следовательно, являются её априорными характеристиками.

Сопоставление формул (5) и (7) позволяет выделить ряд отличий, имеющих практическое значение. Формула (5) требует выполнения операции нахождения остатка по большому модулю M , что на практике связано с высокой вычислительной сложностью. В формуле (7) указанная операция заменяется определением дробной части

числа, что значительно упрощает реализацию – фактически сводится к удалению целой части, то есть переноса в разряд единиц. Однако это преимущество сопровождается существенным недостатком: вычисления по формуле (7) требуют работы с бесконечными дробями. Тогда необходимо ответить на вопрос: с какой точностью необходимо производить вычисления по формуле (7), чтобы обеспечить корректное восстановление позиционного значения числа X на практике? Чтобы ответить на поставленный вопрос, введем методику определения знака числа в СОК, которая будет основана на применении Китайской теоремы об остатках и операциях с дробными значениями.

Обозначим через $F\left(\frac{X}{M}\right)$ бесконечную периодическую двоичную дробь, соответствующую отношению $\frac{X}{M}$. При усечении

этой дроби, начиная с $(N+1)$ -го бита после запятой, получится конечная двоичная дробь,

которую обозначим как $\left[F\left(\frac{X}{M}\right) \right]_{2^{-N}}$.

Например, если $F\left(\frac{X}{M}\right) = 0,011101110\dots$,

то $\left[F\left(\frac{X}{M}\right) \right]_{2^{-6}} = 0,011101$.

Для введенных величин выполняется неравенство

$$\left[F\left(\frac{X}{M}\right) \right]_{2^{-N}} < F\left(\frac{X}{M}\right) < \left[F\left(\frac{X}{M}\right) \right]_{2^{-N}} + 2^{-N},$$

описывающее положение на числовой оси точного значения $F\left(\frac{X}{M}\right)$ относительно

округленной величины $\left[F\left(\frac{X}{M}\right) \right]_{2^{-N}}$.

В работе [27] была продемонстрирована высокая эффективность применения вычислений над дробными величинами для реализации проблемной операции деления в СОК. Далее покажем, как аналогичный подход может быть использован для реализации вычислительно сложной операции определения знака числа в СОК.

Пусть в вычислениях по формуле (7) используются округленные значения вида $\left[F(k_i) \right]_{2^{-N}}$. Следующая теорема предоставляет теоретическое обоснование корректности применения таких приближенных

величин для точного восстановления чисел при обратном преобразовании из системы с дополнительным кодом (СОК) в позиционную систему счисления

Теорема 1. Минимальное значение N , обеспечивающее корректное преобразование числа X из СОК в позиционную систему счисления, согласно формуле

$$\frac{X}{M} = \left| \sum_{i=1}^n x_i [F(k_i)]_{2^{-N}} \right|, \quad (8)$$

будет точным, равно

$$N = \log_2(M\mu), \quad (9)$$

где $\mu = -n + \sum_{i=1}^n m_i$.

Доказательство. Очевидно, что

$$[F(k_i)]_{2^{-N}} < F(k_i) < [F(k_i)]_{2^{-N}} + 2^{-N},$$

для всех $i = 1, 2, \dots, n$. Просуммировав последнее неравенство по всем модулям СОК, получим:

$$\sum_{i=1}^n [F(k_i)]_{2^{-N}} x_i \leq \sum_{i=1}^n F(k_i) x_i \leq \sum_{i=1}^n [F(k_i)]_{2^{-N}} x_i + 2^{-N} \sum_{i=1}^n x_i. \quad (10)$$

Величина $\left| \sum_{i=1}^n k_i x_i \right|$ отражает точное положение числа X на числовой оси. Для обеспечения точного определения значения X необходимо выбрать параметр N таким образом, чтобы в произвольный интервал $\left[\sum_{i=1}^n [F(k_i)]_{2^{-N}} x_i, \sum_{i=1}^n [F(k_i)]_{2^{-N}} x_i + 2^{-N} \sum_{i=1}^n x_i \right]$ попадало не более одного допустимого значения из диапазона системы СОК. Это требование эквивалентно условию $2^{-N} \sum_{i=1}^n x_i < \frac{1}{M}$.

Введем обозначение $\mu = -n + \sum_{i=1}^n m_i$. Наименьшее значение N , при котором возможно точное преобразование числа X из СОК в позиционную систему счисления с использованием формулы (8), определяется выражением $N = \log_2(M\mu)$, что и требовалось доказать.

Применение константы N , вычисленной по формуле (9), в формуле (8) для обратного преобразования числа X по вычислительной сложности приблизительно соответствует выполнению преобразования по формуле (5).

Точное местоположение X на числовой оси не является обязательным условием при определении его знака в СОК. Для корректного определения знака достаточно лишь узнать, в какой из интервалов, изображенных на рисунках 1, 2, попадает исследуемое число. Если использовать в формуле (8) вместо N уменьшенную величину $\tilde{N}$, $\tilde{N} < N$, то интервал (10) увеличится. При этом в интервал (10) могут попадать несколько дробей, соответствующих различным числам в СОК, поэтому использование $\tilde{N}$ для обратного преобразования не даст корректного результата. Однако при решении задачи об определении знака числа в СОК этот увеличенный интервал все еще может быть использован.

В произвольный промежуток $\left[\sum_{i=1}^n [F(k_i)]_{2^{-\tilde{N}}} x_i, \sum_{i=1}^n [F(k_i)]_{2^{-\tilde{N}}} x_i + 2^{-\tilde{N}} \sum_{i=1}^n x_i \right]$ могут попасть не более $2^{-\tilde{N}} M \sum_{i=1}^n x_i$ дробей $F\left(\frac{X}{M}\right)$, соответствующих различным числам в СОК.

Если учесть, что $0 \leq x_i < m_i$ (формула (1)), то $\max \left\{ 2^{-\tilde{N}} M \sum_{i=1}^n x_i \right\} = 2^{-\tilde{N}} M \left(-n + \sum_{i=1}^n m_i \right) = 2^{-\tilde{N}} M \mu$.

Для определения знака числа в СОК по его дробной части $F\left(\frac{X}{M}\right)$ необходимо учитывать следующие условия:

- если значение $F\left(\frac{X}{M}\right)$ удовлетворяет $0 < F\left(\frac{X}{M}\right) < \frac{1}{2}$, то число X является положительным;

- если $F\left(\frac{X}{M}\right)$ удовлетворяет $\frac{1}{2} < F\left(\frac{X}{M}\right) < 1$, то число X отрицательное.

Ранее доказанная теорема 1 может служить теоретической основой для разработки нового алгоритма определения знака числа в СОК. Ниже представлен разработанный алгоритм определения знака числа в СОК, основанный на использовании величины $\left[F\left(\frac{X}{M}\right) \right]_{2^{-N}}$, где N определяется по формуле (9).

Алгоритм

1. В СОК с модулями $\{m_1, m_2, \dots, m_n\}$ предварительно вычисляются константы

$$M = m_1 m_2 \dots m_n, \quad \mu = -n + \sum_{i=1}^n m_i, \quad N = \log_2(M\mu) \quad \text{и} \quad k_i = \frac{|M_i^{-1}|_{m_i} M_i}{M}.$$

2. Для констант k_i , определенных в пункте 1, определяются дробные величины $[F(k_i)]_{2^{-N}}$.

3. Вычисляется $\left[F\left(\frac{X}{M}\right) \right]_{2^{-N}} = \left| \sum_{i=1}^n x_i [F(k_i)]_{2^{-N}} \right|_1.$

4. При условии $0 < \left[F\left(\frac{X}{M}\right) \right]_{2^{-N}} < \frac{1}{2}$, число X положительное.

Если $\frac{1}{2} < \left[F\left(\frac{X}{M}\right) \right]_{2^{-N}} < 1$, то X отрицательное.

Заданной точности N , из пунктов 5-6 приведенного алгоритма, достаточно для корректного определения знака числа в СОК. Это обеспечивается теоремой 1. Следующий пример демонстрирует работоспособность метода и его вычислительную сложность, сравнимую с применением классической КТО.

Пример 1. Пусть СОК задана основаниями $m_1 = 2, m_2 = 3, m_3 = 5$ и $m_4 = 7$.

Тогда $M = 2 \cdot 3 \cdot 5 \cdot 7 = 210$, $\mu = -4 + 2 + 3 + 5 + 7 = 13$, $N = \log_2(210 \cdot 13) = 12$,

$$k_1 = \frac{1}{2}, \quad k_2 = \frac{1}{3}, \quad k_3 = \frac{3}{5}, \quad k_4 = \frac{4}{7}.$$

Дроби, соответствующие константам $k_i, i = 1, \dots, 4$, будут иметь вид:

$$[F(k_1)]_{2^{-12}} = 0,100000000000, \quad [F(k_2)]_{2^{-12}} = 0,010101010101,$$

$$[F(k_3)]_{2^{-12}} = 0,100110011001, \quad [F(k_4)]_{2^{-12}} = 0,100100100100.$$

Определим знак числа $X = 200 = (0, 2, 0, 4)$

$$\begin{aligned} \left[F\left(\frac{X}{M}\right) \right]_{2^{-12}} &= |0 \cdot 0,100000000000 + 10 \cdot 0,010101010101 + \\ &+ 0 \cdot 0,100110011001 + 100 \cdot 0,100100100100|_1 = 0,111100111010. \end{aligned}$$

Так как $0,1 < 0,111100111010 < 1$, то число $X = 200 = (0, 2, 0, 4)$ отрицательное.

Далее будет показано, как применение разработанного алгоритма позволяет ускорить определение знака числа в СОК при программной реализации вычислительной системы.

С целью сравнения работы алгоритмов определения знака числа на основе метода быстрого преобразования в обобщенной позиционной системе счисления (ОПСС), описанного в [28], и на основе КТО с дробными величинами было проведено программное моделирование. Мо-

делирование проводилось на языке Python 3.11. Для вычислений использовалась ЭВМ с процессором Intel Core i5-12450H, с 16 ГБ оперативной памяти и 64-битной операционной системой Windows 10. Для моделирования предложенного алгоритма были выбраны СОК, представленные в таблице 1. Данный выбор объясняется тем, что представленные наборы модулей СОК₁ – СОК₆ позволяют представить востребованные на практике числовые диапазоны от 256 до 2048 бит.

Таблица 1

Наборы модулей СОК, перекрывающие различные динамические диапазоны

Динамический диапазон	n	Набор модулей
СОК_1 , $M > 2^{256}$	9	365284271, 365284273, 365284277, 365284279, 365284289, 365284291, 365284295, 365284296, 365284297
СОК_2 , $M > 2^{384}$	13	779763569, 779763570, 779763571, 779763577, 779763581, 779763583, 779763587, 779763589, 779763593, 779763599, 779763601, 779763613, 779763617
СОК_3 , $M > 2^{512}$	17	1164971089, 1164971099, 1164971107, 1164971113, 1164971117, 1164971119, 1164971123, 1164971131, 1164971141, 1164971147, 1164971149, 1164971153, 1164971155, 1164971157, 1164971159, 1164971161, 1164971162
СОК_4 , $M > 2^{768}$	25	1768648181, 1768648183, 1768648187, 1768648199, 1768648207, 1768648213, 1768648223, 1768648229, 1768648237, 1768648241, 1768648243, 1768648247, 1768648249, 1768648253, 1768648261, 1768648267, 1768648271, 1768648277, 1768648279, 1768648283, 1768648285, 1768648287, 1768648289, 1768648291, 1768648292
СОК_5 , $M > 2^{1024}$	17	1357157737484444893, 1357157737484444897, 1357157737484444899, 1357157737484444911, 1357157737484444921, 1357157737484444923, 1357157737484444927, 1357157737484444933, 1357157737484444935, 1357157737484444939, 1357157737484444941, 1357157737484444947, 1357157737484444951, 1357157737484444953, 1357157737484444957, 1357157737484444958, 1357157737484444959
СОК_6 , $M > 2^{2048}$	33	4809544787461394623, 4809544787461394627, 4809544787461394633, 4809544787461394639, 4809544787461394641, 4809544787461394647, 4809544787461394653, 4809544787461394657, 4809544787461394663, 4809544787461394671, 4809544787461394677, 4809544787461394681, 4809544787461394683, 4809544787461394689, 4809544787461394693, 4809544787461394699, 4809544787461394707, 4809544787461394713, 4809544787461394719, 4809544787461394723, 4809544787461394731, 4809544787461394737, 4809544787461394741, 4809544787461394749, 4809544787461394753, 4809544787461394759, 4809544787461394761, 4809544787461394765, 4809544787461394767, 4809544787461394769, 4809544787461394771, 4809544787461394773, 4809544787461394774

Примечание: составлено автором на основе проведенных вычислений.

Таблица 2

Время работы алгоритмов определения знака числа в СОК

СОК	Предложенный алгоритм	Метод ОПСС [17]
СОК_1	12.1219	18.0698
СОК_2	18.4128	39.2565
СОК_3	30.8549	43.8699
СОК_4	20.1168	71.3287
СОК_5	18.6590	41.3983
СОК_6	52.5191	154.1096

Примечание: составлено автором с использованием разработанного алгоритма и метода из источника [28].

Критерием для сравнения алгоритмов выбрано время выполнения, выраженное в секундах. Было проведено 1 000 000 итераций для каждого из алгоритмов во всех мо-

делируемых случаях. В таблице 2 приведены результаты моделирования алгоритмов с указанием среднего времени выполнения.

Предложенный алгоритм выигрывает в скорости во всех рассмотренных случаях. Для СОК_1 время работы сокращается в ≈ 1.5 раза по сравнению с алгоритмами на основе МРС. В случае СОК_2 время работы сокращается в ≈ 1.8 раза. Для СОК_3 сокращение временных затрат составляет ≈ 1.4 раза. Для СОК_4 время сокращается в ≈ 3.5 раза. Для СОК_5 время сокращается в ≈ 2.2 раза. Наконец, для СОК_6 время работы сокращается в ≈ 2.9 раза. Другими словами, время работы предложенного алгоритма составляет 34–67% от времени работы алгоритма из [28]. Указанные цифры зависят от выбранной СОК и ее характеристик: модулей, их количества и т.д.

Данные из таблицы 2 позволяют сделать вывод о том, что предложенный алгоритм быстрее известного аналогичного

алгоритма определения знака числа в СОК с модулями произвольного вида. Продемонстрированный результат может быть использован для разработки новых способов реализации проблемных операций СОК – сравнения чисел и деления. Кроме того, более быстрое определение знака числа в СОК на основе предложенного метода позволит значительно расширить область практического применения СОК, особенно в области реализации систем машинного обучения и для цифровой обработки сигналов и изображений.

Заключение

Предложенный в статье алгоритм определения знака числа в СОК основан на модифицированной версии Китайской теоремы об остатках с использованием вычислений над дробными величинами. Результаты моделирования продемонстрировали, что данный алгоритм превосходит по скорости алгоритм на основе ОПСС. Предложенный в данной работе алгоритм является универсальным и не привязан к какому-либо набору модулей СОК. Данное обстоятельство может быть использовано в приложениях, использующих динамическое изменение параметров вычислительной системы.

Предложенный алгоритм определения знака позволяет ускорить данную проблемную операцию в СОК. Стоит отметить, что определение знака является базовой операцией для других проблемных задач, к которым относятся: сравнение чисел в СОК, деление в СОК, обнаружение переполнения диапазона, локализация ошибки и другие. Применение предложенного алгоритма позволит улучшить время выполнения перечисленных операций. Любое улучшение проблемных операций в СОК может расширить границы ее применения и найти новые приложения на практике, в которых СОК может быть использована с максимальной эффективностью, например в цифровой обработке сигналов, изображений и видео, а также в машинном обучении.

Список литературы

1. Куприянов М.С., Холод И.И., Прокопчина С.В. Мягкие вычисления и измерения. М.: Издательский дом «Научная библиотека», 2019. 562 с. ISBN: 978-5-6042213-9-6.
2. Ляхов П.А. Разработка методов и алгоритмов вейвлет-анализа для цифровой обработки сигналов: дис. ... канд. физ.-мат. наук. Ставрополь, 2012. 209 с. URL: <https://www.dissercat.com/content/razrabotka-metodov-i-algoritmov-veivlet-analiza-dlya-tsifrovoy-obrabotki-signalov> (дата обращения: 26.06.2025).
3. Cardarilli G., Nunzio L.D., Fazzolari R., Nannarelli A., Petricca M., Re M. Design Space Exploration Based Methodology for Residue Number System Digital Filters Implementation // IEEE Transactions on Emerging Topics in Computing. 2022. Vol. 10. № 1. P. 186–198. DOI: 10.1109/tetc.2020.2997067.
4. Givaki K., Khonsari A., Gholamrezaei M., Gorgin S., Najafi M.H. A Generalized Residue Number System Design Approach for Ultralow-Power Arithmetic Circuits Based on Deterministic Bit-Streams // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2023. Vol. 42. № 11. P. 3787–3800. DOI: 10.1109/TCAD.2023.3250603.
5. Калита Д.И. Разработка фильтров высокой эффективности для объектов цифровых систем видеонаблюдения на основе системы остаточных классов: дис. ... канд. тех. наук. Ставрополь, 2017. 226 с. URL: <https://www.dissercat.com/content/razrabotka-filtrov-vysokoi-effektivnosti-dlya-obektov-tsifrovyykh-sistem-videonablyudeniya> (дата обращения: 26.06.2025).
6. Валуева М.В. Разработка методов и алгоритмов построения цифровых устройств интеллектуального анализа визуальных данных: 2.3.5 «Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей»: диссертация на соискание ученой степени кандидата технических наук. Ставрополь, 2023. 154 с.
7. Валуева М.В. Разработка методов и алгоритмов построения цифровых устройств интеллектуального анализа визуальных данных: дис. ... канд. тех. наук. Ставрополь, 2023. 154 с. URL: <https://www.dissercat.com/content/razrabotka-metodov-i-algoritmov-postroeniya-tsifrovyykh-ustroystv-intellektualnogo-analiza> (дата обращения: 26.06.2025).
8. Лавриненко А.В. Разработка методов моделирования вычислительных структур отказоустойчивых модулярных нейрокомпьютеров для обработки данных большой размерности: дис. ... канд. тех. наук. Ставрополь, 2016. 179 с. URL: <https://www.dissercat.com/content/razrabotka-metodov-modelirovaniya-vychislitelnykh-struktur-otkazoustoichivyykh-modulyarnyykh> (дата обращения: 26.06.2025).
9. Shen S., Yang H., Liu Y., Liu Z., Zhao Y. CARM: CUDA-Accelerated RNS Multiplication in Word-Wise Homomorphic Encryption Schemes for Internet of Things // IEEE Transactions on Computers. 2022. P. 1–12. DOI: 10.1109/TC.2022.3227874.
10. Бабенко М.Г. Математические модели, методы и алгоритмы обработки зашифрованных данных в распределенных средах: дис. ... докт. физ.-мат. наук. Москва, 2022. 415 с. URL: <https://www.dissercat.com/content/matematicheskie-modeli-metody-i-algoritmy-obrabotki-zashifrovannykh-dannykh-v-raspredelemnnykh> (дата обращения: 26.06.2025).
11. Al Badawi A., Polyakov Y., Aung K.M., Veeravalli B., Rohloff K. Implementation and Performance Evaluation of RNS Variants of the BFV Homomorphic Encryption Scheme // IEEE Trans Emerg Top Comput. 2021. Vol. 9. № 2. P. 941–956. DOI: 10.1109/TETC.2019.2902799.
12. Adday G.H., Subramaniam S.K., Zukarnain Z.A., Samian N. Fault Tolerance Structures in Wireless Sensor Networks (WSNs): Survey, Classification, and Future Directions // Sensors. 2022. Vol. 22. № 16. P. 6041. URL: www.mdpi.com/1424-8220/22/16/6041 (дата обращения: 15.04.2025).
13. Chervyakov N.I., Lyakhov P.A., Kalita D.I., Shulzhenko K.S. Effect of RNS moduli set selection on digital filter performance for satellite communications // 2015 International Siberian Conference on Control and Communications (SIBCON). IEEE. 2015. P. 1–7. DOI: 10.1109/SIBCON.2015.7147268.
14. Chistousov N.K., Kalmykov I.A., Dukhovnyy D.V. Mathematical and structural models of the ofdm transmission system built on the basis of orthogonal Dobeche transformations in resolution class codes // Современные наукоемкие технологии (Modern High Technologies). 2023. № 8. P. 84–90. URL: top-technologies.ru/en/article/view?id=39735 (дата обращения: 15.04.2025).
15. Li Y., Xiao L. An Improved Molecular Computing Model of Modular-Multiplication over Finite Field GF(2n) // 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT). IEEE. 2016. P. 262–267. URL: ieeexplore.ieee.org/document/7943368 (дата обращения: 15.04.2025).
16. Shiriaev E., Kucherov N., Babenko M., Nazarov A. Fast Operation of Determining the Sign of a Number in RNS

Using the Akushsky Core Function // *Computation*. 2023. Vol. 11. P. 124. DOI: 10.3390/computation11070124.

17. Isupov K. Using Floating-Point Intervals for Non-Modular Computations in Residue Number System // in *IEEE Access*. 2020. Vol. 8. P. 58603-58619. DOI: 10.1109/ACCESS.2020.2982365.

18. Бергерман М.В. Использование системы остаточных классов с модулями вида $\{2n-1, 2n, 2n+1\}$ для снижения аппаратных затрат цифрового фильтра // *Известия высших учебных заведений. Поволжский регион. Технические науки*. 2023. № 1. С. 32–43. DOI: 10.21685/2072-3059-2023-1-3.

19. Yamani S.V., Raghu Vamsi Kudulla H.K., Bhavani D.D. Area Efficient Sparse-4 Diminished-1 Modulo $2n+1$ Adder // *IEEE Wireless Antenna and Microwave Symposium (WAMS)*, Visakhapatnam, India, 2024. P. 1-7. DOI: 10.1109/WAMS59642.2024.10527894.

20. Kaplun D., Voznesensky A., Veligosha A.V., Kalmykov I.A. Technique to Adjust Adaptive Digital Filter Coefficients in Residue Number System Based Filters // *IEEE Access*. 2021. Vol. 9. P. 82402–82416. DOI: 10.1109/ACCESS.2021.3085704.

21. Younes D., Steffan P. A comparative study on different moduli sets in residue number system // 2012 International Conference on Computer Systems and Industrial Informatics. IEEE. 2012. P. 1–6. DOI: 10.1109/ICCSII.2012.6454344.

22. Minghe Xu, Zhenpeng Bian, Ruohu Yao. Fast Sign Detection Algorithm for the RNS Moduli Set $\{2^{n+1}-1, 2^n-1, 2^n\}$ // *IEEE Trans Very Large Scale Integr VLSI Syst*. 2015. Vol. 23. № 2. P. 379–383. DOI: 10.1109/TVLSI.2014.2308014.

23. Tomczak T. Fast Sign Detection for RNS $\{2^n-1, 2^n, 2^{n+1}\}$ // *IEEE Transactions on Circuits and Systems I: Regular Papers*. 2008. Vol. 55. № 6. P. 1502–1511. DOI: 10.1109/TCSI.2008.917994.

24. Mohan P.V.A., Phalguna P.S. Evaluation of Mixed-Radix Digit Computation Techniques for the Three Moduli RNS $\{2^n-1, 2^n, 2^{n+1}-1\}$ // *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2021. Vol. 68. № 4. P. 1418–1422. DOI: 10.1109/TCSII.2020.3035350.

25. Setiaarif E., Siy P. A new moduli set selection technique to improve sign detection and number comparison in Residue Number System (RNS) // *NAFIPS 2005 – 2005 Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE. P. 766–768. DOI: 10.1109/NAFIPS.2005.1548635.

26. Kumar S., Chang C.-H. A Scaling-Assisted Signed Integer Comparator for the Balanced Five-Moduli Set RNS $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1, 2^{n-1}-1\}$ // *IEEE Trans Very Large Scale Integr VLSI Syst*. 2017. Vol. 25. № 12. P. 3521–3533. DOI: 10.1109/TVLSI.2017.2748984.

27. Hung C.Y., Parhami B. An approximate sign detection method for residue numbers and its application to RNS division // *Computers & Mathematics with Applications*. 1994. Vol. 27. № 4. P. 23–35. DOI: 10.1016/0898-1221(94)90052-3.

28. Gbolagade K.A., Cotofana S.D. An $O(n)$ Residue Number System to Mixed Radix Conversion technique // 2009 IEEE International Symposium on Circuits and Systems. IEEE. 2009. P. 521–524. URL: ieeexplore.ieee.org/document/5117800 (дата обращения: 15.04.2025).