DOI: 10.15514/ISPRAS-2023-35(5)-11

Оптимизация алгоритма деления чисел в системе остаточных классов на основе функции ядра Акушского

[1] *В.В. Луценко,* ORCID: 0000-0003-4648-8286 <*vvlutcenko@ncfu.ru*>
[2,3] *М.Г. Бабенко,* ORCID: 0000-0001-7066-0061 <*mgbabenko@ncfu.ru*>
[3,4] *А.Н. Черных,* ORCID: 0000-0001-5029-5212 <*chernykh@cicese.mx*>
[2] *М.А. Лапина,* ORCID: 0000-0001-8117-9142 <*mlapina@ncfu.ru*>

[1] *Северо-Кавказский центр математических исследований, Северо-Кавказский федеральный университет,*
355017, Россия, г. Ставрополь, ул. Пушкина, д. 1.
[2] *Северо-Кавказский федеральный университет,*
355017, Россия, г. Ставрополь, ул. Пушкина, д. 1.
[3] *Институт системного программирования РАН им. В.П. Иванникова,*
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.
[4] *Центр научных исследований и высшего образования Энсенада,*
В.С. 22860, Мексика.

Аннотация. Система остаточных классов широко применяются в криптографии, цифровой обработке сигналов, системах обработки изображений и других областях, где требуется выполнение операций деления. Однако, операция деления является наиболее сложной с точки зрения вычислений в системе остаточных классов. В статье представлен оптимизированный алгоритм деления, основанный на функции ядра Акушского. Показано, что предложенный алгоритм по скорости вычислений эффективней, чем классическое итерационное деление.

Ключевые слова: система остаточных классов; функция ядра Акушского; модулярная арифметика; немодульные операции; итерационное деление.

Для цитирования: Луценко В.В., Бабенко М.Г., Черных А.Н., Лапина М.А. Оптимизация алгоритма деления чисел в системе остаточных классов на основе функции ядра Акушского. Труды ИСП РАН, том 35, вып. 5, 2023 г., стр. 157–168. DOI: 10.15514/ISPRAS–2023–35(5)–11.

Благодарности: Исследование выполнено за счет гранта Российского научного фонда № 19-71-10033, https://rscf.ru/project/19-71-10033/.

Optimization of a number division algorithm in the residue number system based on the Akushsky core function

[1] *V.V. Lutsenko* ORCID: 0000-0003-4648-8286 <*vvlutcenko@ncfu.ru*>
[2,3] *M.G. Babenko* ORCID: 0000-0001-7066-0061 <*mgbabenko@ncfu.ru*>
[3,4] *A.N. Tchernykh,* ORCID: 0000-0001-5029-5212 <*chernykh@cicese.mx*>
[2] *M.A. Lapina,* ORCID: 0000-0001-8117-9142 <*mlapina@ncfu.ru*>

[1] *North-Caucasus Center for Mathematical Research, North-Caucasus Federal University,*
1, Pushkin st., Stavropol, 355017, Russia.
[2] *North-Caucasus Federal University, Stavropol,*
1, Pushkin st., Stavropol, 355017, Russia.
[3] *Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[4] *CICESE Research Center,*
Ensenada, Baja California, 22860, Mexico.

Abstract. Residue number systems find wide application in cryptography, digital and image signal processing, and other domains necessitating division operations. Nevertheless, division is the most computationally intensive activity in residue number systems. An optimized division algorithm based on the Akushsky core function is presented in this paper. The suggested method exhibits superior computational efficiency when compared to the conventional iterative division.

Keywords: residue number system; Akushsky core function; modular arithmetic; non-modular operation; iterative division.

For citation: Lutsenko V.V., Babenko M.G., Tchernykh A.N., Lapina M.A. Optimization of a number division algorithm in the residue number system based on the Akushsky core function. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 5, 2023. pp. 157-168 (in Russian). DOI: 10.15514/ISPRAS-2023-35(5)-11.

Acknowledgements. The research was supported by the Russian Science Foundation Grant No. 19-71-10033, https://rscf.ru/en/project/19-71-10033/.

1. Введение

В последнее десятилетие система остаточных классов (СОК) стала объектом повышенного внимания в сфере вычислительной техники. Это нетрадиционное представление чисел, основанное на арифметике остаточных классов, стало фундаментом для многочисленных исследований и практических применений в различных областях, включая цифровую обработку сигналов, системы обработки изображений, помехоустойчивое кодирование, криптографию, квантовые автоматы, нейрокомпьютеры, системы с массовым параллелизмом операций, облачные вычисления и многие другие [1–7]. Система остаточных классов используется в архитектуре математических сопроцессоров криптосистем RSA [14].

Одной из наиболее сложных операций в СОК является деление чисел. Сложность этой операции в СОК привела к необходимости разработки оптимизированных алгоритмов, способных обеспечивать быстрые результаты при выполнении деления, особенно в случаях, где требуются большие объемы вычислений.

Существующие алгоритмы деления в СОК обычно можно разделить на две категории: алгоритмы, использующие умножение [8], и алгоритмы, использующие вычитание [9]. Большинство алгоритмов, основанных на умножении, требуют предварительного вычисления обратной величины делителя, что влечет за собой дополнительные вычислительные затраты. В то время как алгоритмы, основанные на вычитании, могут обойтись без обратной величины делителя, они, в свою очередь, могут использовать другие немодульные операции, что также снижает эффективность.

Цель данной статьи – исследовать и усовершенствовать алгоритм деления чисел в системе остаточных классов с использованием функции ядра Акушского.

Статья имеет следующую структуру. В разделе 2 рассматривается система остаточных классов. В разделе 3 представлены основы функции ядра Акушского. Затем, в разделе 4 исследуется метод итерационного деления на основе функции ядра. В разделе 5 представлен способ оптимизации итерационного деления. Наконец, в разделе 6 проведен анализ эффективности предложенного метода. В заключении суммируются полученные результаты.

2. Система остаточных классов

Представление числа в СОК основано на китайской теореме об остатке (КТО). Пусть $p_1, p_2, \dots, p_n$ – взаимно простые модули, и $P = p_1 \cdot p_2 \cdot \dots \cdot p_n$ – их произведение, называемое динамическим диапазоном. Для каждого числа X, существует набор остатков $x_1, x_2, \dots, x_n$, где $0 \le x_i < p_i$, эти остатки образуют представление числа в СОК [10]. Математически это можно выразить следующим образом:

$$x_i \equiv X(mod\ p_i). \tag{1}$$

Таким образом, число X записывается в СОК в следующей форме:

$$X = (x_1, x_2, \dots, x_n), \tag{2}$$

Вычеты (1) x_i можно вычислить по формуле:

$$x_i = X - \left\lfloor \frac{X}{p_i} \right\rfloor \cdot p_i, \tag{3}$$

Для выполнения операций над числами в СОК, таких как сложение и умножение, операции выполняются независимо над остатками по каждому модулю. Вычисления в СОК выполняются по формуле:

$$X * Y = (x_1 * y_1, x_2 * y_2, \dots, x_n * y_n). \tag{4}$$

где $*$ обозначает арифметические операции: сложение $(+)$, вычитание $(-)$ или умножение $(\cdot)$. Каждый модуль СОК взаимно прост с другими модулями, т.е. выполняется условия: $\text{НОД}(p_i, p_j) = 1$, для всех $i \ne j$.

3. Функция ядра Акушского

Поиск позиционных характеристик, которые позволили бы уменьшить вычислительную сложность алгоритма сравнения чисел в СОК привели исследователей Акушского И.Я., Бурцева В.М. и Пака И.Т. к построению новой конструкции, названной функцией ядра Акушского [11]. Задается функция ядра Акушского следующей формулой:

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

где целые числа w_i – постоянные определяемые выбором точки интерполяции. Константы w_i задают вес каждого из частных $\left\lfloor \frac{X}{p_i} \right\rfloor$ в формуле (5) тем самым задавая функцию ядра и придавая ей различные свойства.

Веса w_i, участвующие в формуле (5), предоставляют нам значительную гибкость и могут быть подобраны с учетом конкретной цели. Эти коэффициенты w_i играют ключевую роль в определении уникальных функций ядра и могут быть настроены в соответствии с требованиями задачи, над которой мы работаем. Алгоритм определения оптимальных весов для функции Акушского представлен в статье [12].

Основное свойство функции ядра заключается в том, что ее максимальный диапазон изменяется и может быть значительно меньше числа P в зависимости от выбора весов. Например, мы можем использовать произвольное значение $C(P)$ в качестве $C(P)$, при

условии, что оно обладает необходимыми свойствами для решения нашей конкретной задачи. Это значение называется диапазоном функции ядра и определяется выражением.

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

где $P_i = \frac{P}{p_i}$.

Учитывая, что $P_i \equiv 0(mod\ p_i)$ для любого $i \ne j$, константы w_i этой функции могут быть определены из соотношения

$$w_i \equiv C(P) \cdot P_i^{-1}(mod\ p_i). \tag{7}$$

При этом необходимо учитывать, что (7) задает некий класс вычетов для каждого i, и числа w_i могут оказаться отрицательными или положительными в каждом конкретном случае.

Значение функции ядра $C(X)$, заданной весами $w_1, w_2, \dots, w_n$, при условии $0 \le C(X) \le C(P), X \in [0, P)$, можно вычислить с использованием формулы

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

где $c_i = C(B_i)$ и $B_i = P_i \cdot |P_i^{-1}|_{p_i}$, $|P_i^{-1}|_{p_i}$ это мультипликативная инверсия P_i по модулю p_i.

4. Итерационное деление

Рассмотрим применение алгоритма деления с использованием ядерной позиционной характеристики.

Алгоритм деления базируется на особой простоте выполнения элементарных операций деления на 2 и на основание СОК, а также на определения факта переполнения при сложении двух чисел [13].

Рассмотрим процесс итерационного деления делимого $A = (\alpha_1, \alpha_2, \dots, \alpha_n)$ с ядром C_A на делитель $B = (\beta_1, \beta_2, \dots, \beta_n)$ с ядром C_B поэтапно.

Первый этап – если $\beta_1 = 0$, то делим B на p_1 в противном случае делим B на 2, получаем B_1. Делим B_1 на p_1, если $\beta_1' = 0$, и на 2, в противном случае получаем B_2 и т.д. до $B_k = 1$.

Параллельно с этим делим A на p_1, если $\beta_1 = 0$, и на 2, в противном случае получаем A_1. Делим A_1 на p_1, если $\beta_1' = 0$, и на 2, в противном случае получаем A_2 и т.д. до A_k.

Второй этап – вычисляется первая невязка:

$$A - BA_k = Q^{(1)}. \tag{9}$$

Третий этап в соответствии с первым этапом деления делителя B вычисляется так:

$$Q_1, Q_2, \dots, Q_k^{(1)}. \tag{10}$$

При этом если имеется возможность запомнить соответствующие делители $d_1, d_2, \dots, d_k$ делителя B, который является как бы ведущим в рассматриваем процессе деления, то не надо повторять деление B, и содержание этапа сводится лишь к делению $Q^{(1)}$.

Четвертый этап – вычисляется вторая невязка:

$$Q_1 - Q_k^{(1)} \cdot B = Q^{(2)}. \tag{11}$$

Эти этапы повторяются до получения нулевого промежуточного частного $Q_i^{(l+1)} = 0$ $(i = 1,2, \dots, k)$. Тогда искомое частное:

$$Z = \frac{A}{B} = A_k + Q_k^{(1)} + Q_k^{(2)} + \dots + Q_k^{(l)}, \tag{12}$$

где

$$Q_k^{(l+1)} = 0. \tag{13}$$

Таким образом, алгоритм итерационного деления чисел A и B будет иметь вид:

Алгоритм. Алгоритм итерационного деления чисел A и B.

Input: $A \xrightarrow{RNS} (\alpha_1, \alpha_2, \dots, \alpha_n)$,

$B \xrightarrow{RNS} (\beta_1, \beta_2, \dots, \beta_n)$,

$p_1, p_2, \dots, p_n, w_1, w_2, \dots, w_n$,

$P = p_1 \cdot p_2 \cdot \dots \cdot p_n, P_i = \frac{P}{p_i}, C_P = \sum_{i=1}^{n} w_i \cdot P_i$ для всех $i = \overline{1,n}$,

$c_i = C(B_i), B_i = P_i \cdot |P_i^{-1}|_{p_i}$ для всех $i = \overline{1,n}$,

Output: $Z = A/B$.

1. **Function** basic_division(X,k)
1.1. Initialize X_k, C_{X_k} to empty list;
1.2. **If** $x_1 = 0$ **then**
1.2.1. Append the $(x_1, x_2, \dots, x_n)/p_1$ to the list X_k;
1.2.2. Append the $C_{X_1} = C(X_1)$ to the list C_{X_k};
1.3. **Else**
1.3.1. Append the $(x_1, x_2, \dots, x_n)/2$ to the list X_k;
1.3.2. Append the $C_{X_1} = C(X_1)$ to the list C_{X_k};
1.4. **For** $i := 2$ **to** k **do:**
1.4.1. **If** $x_1' = 0$ **then**
1.4.1.1. Append the X_i/p_1 to the list X_k;
1.4.1.2. Append the $C_{X_i} = C(X_i)$ to the list C_{X_k};
1.5.2. **Else**
1.5.2.1. Append the $X_i/2$ to the list X_k;
1.5.2.2. Append the $C_{X_i} = C(X_i)$ to the list C_{X_k};
1.5. **return** X_k, C_{X_k}
2. Initialize A_k, C_{A_k}, B_k, C_{B_k} to empty list;
3. **If** $\beta_1 = 0$ **then**
3.1. Append the $(\beta_1, \beta_2, \dots, \beta_n)/p_1$ to the list B_k;
3.2. Append the $C_{B_1} = C(B_1)$ to the list C_{B_k};
3.3. Append the $(\alpha_1, \alpha_2, \dots, \alpha_n)/p_1$ to the list A_k;
3.4. Append the $C_{A_1} = C(A_1)$ to the list C_{A_k};
4. **Else**
4.1. Append the $(\beta_1, \beta_2, \dots, \beta_n)/2$ to the list B_k;
4.2. Append the $C_{B_1} = C(B_1)$ to the list C_{B_k};
4.3. Append the $(\alpha_1, \alpha_2, \dots, \alpha_n)/2$ to the list A_k;
4.4. Append the $C_{A_1} = C(A_1)$ to the list C_{A_k};
5. **while** $B_k \neq 1$ **do:**
5.1. **If** $\beta_1' = 0$ **then**
5.1.1. Append the B_i/p_1 to the list B_k;
5.1.2. Append the $C_{B_i} = C(B_i)$ to the list C_{B_k};
5.1.3. Append the A_i/p_1 to the list A_k;
5.1.4. Append the $C_i = C(A_i)$ to the list C_{A_k};
5.2. **Else**
5.2.1. Append the $B_i/2$ to the list B_k;
5.2.2. Append the $C_{B_i} = C(B_i)$ to the list C_{B_k};
5.2.3. Append the $A_i/2$ to the list A_k;
5.2.4. Append the $C_i = C(A_i)$ to the list C_{A_k};
5. Initialize $Q^{(k)}, Q_k^{(l)}$ to empty list;
6. Append the $Q^{(1)} = A - BA_k$ to the list $Q^{(k)}$;
7. $Q_k^{(1)}, C_{Q_k^{(1)}}$ = basic_division$\left(Q^{(1)}, k\right)$

8. Append the $Q_k^{(1)}$ to the list $Q_k^{(l)}$;
9. Append the $Q^{(2)} = Q_1 - Q_k^{(1)} \cdot B$ to the list $Q^{(k)}$;
10. **while** $Q_i^{(I)} \neq 0$ **do:**
10.1. $Q_k^{(i)}, C_{Q_k^{(i)}}$ = basic_division$\left(Q^{(i)}, k\right)$
10.2. Append the $Q_k^{(i)}$ to the list $Q_k^{(l)}$;
10.3. Append the $Q^{(i)} = Q_i - Q_k^{(i)} \cdot B$ to the list $Q^{(k)}$;
11. $Z = A_k + \sum_{i=1}^{l} Q_k^{(i)}$;
End.

Пример 1. Рассмотрим СОК с основаниями $p_1 = 7, p_2 = 9, p_3 = 11$, с системой весов $w_1 = -1, w_2 = -1, w_3 = 3$, ядром диапазона $C_P = 13$.

Пусть дано делимое $A = (5,7,10)$ с ядром $C_A = 7$ и делитель $B = (3,6,10)$ с ядром $C_B = 0$.

Первый этап:

$$B_1 = (3,6,10)/2 \approx (2,5,9)/2 = (1,7,10)\ c\ C_{B_1} = -1;$$
$$B_2 = (1,7,10)/2 \approx (0,6,9)/2 = (0,3,10)\ c\ C_{B_2} = -2;$$
$$B_3 = (0,6,9)/p_1 = (0,6,9)/7 = (3,3,3)\ c\ C_{B_3} = 0;$$
$$B_4 = (3,3,3)/2 \approx (2,2,2)/2 = (1,1,1)\ c\ C_{B_4} = 0.$$

Параллельно выполняем операции над делимым A:

$$A_1 = (5,7,10)/2 \approx (4,6,2)/2 = (2,3,10)\ c\ C_{A_1} = 2;$$
$$A_2 = (2,3,10)/2 \approx (1,2,9)/2 = (4,1,10)\ c\ C_{A_2} = 0;$$
$$A_3 = (4,1,10)/7 = (0,6,6)/7 = (1,6,4)\ c\ C_{A_3} = 0;$$
$$A_4 = (1,6,4)/2 \approx (0,5,3)/2 = (0,7,7)\ c\ C_{A_4} = -1.$$

Первое промежуточное частное $A_4 = (0,7,7)$с $C_{A_4} = -1$.

Второй этап – вычисляется первый невязка.

Для этого найдем $A_4 \times B = (0,7,7) \cdot (3,6,10) = (0,6,4)$ с истинным ядром $C_{A_4B} = 11$.

Вычисляем функции четности A_4, B и $A_4 \cdot B$. Получаем $\varphi(A_4) = 1, \varphi(B) = 1, \varphi(A_4 \cdot B) = 1$.

Так как $\varphi(A_4) \cdot \varphi(B) = \varphi(A_4 B)$, то, следовательно, $A_4 B < P$. Найдем невязку $Q^{(1)} = A - A_4 B = (5,7,10) - (0,6,4) = (5,1,6)$ с расчетным ядром $C_{Q^{(1)}} = 9$.

Так как $\bar{C}_{Q^{(1)}} \neq C_{Q^{(1)}}$, то истинная невязка равна $Q_n^{(1)} = P - Q^{(1)} = (2,8,5)$ с ядром $C_{Q_n^{(1)}} = 3$ и знаком минус.

Третий этап – повторение первого этапа для $Q_n^{(1)}$:

$$Q_1^{(1)} = (2,8,5)/2 = (1,4,8)\ c\ C_{Q_1^{(1)}} = 0;$$
$$Q_2^{(1)} = (1,4,8)/2 \approx (0,3,7)/2 = (0,6,9)\ c\ C_{Q_2^{(1)}} = -1;$$
$$Q_3^{(1)} = (0,6,9)/7 = (6,6,6)\ c\ C_{Q_3^{(1)}} = 0;$$
$$Q_4^{(1)} = (6,6,6)/2 = (3,3,3)\ c\ C_{Q_4^{(1)}} = 0.$$

Второе промежуточное частное $Q_4^{(1)} = (3,3,3)$ с $C_{Q_4^{(1)}} = 0$.

Четвертый этап – вычисляется вторая невязка.

Следуя второму этапу, получим $Q_4^{(1)} \cdot B = (3,3,3) \times (3,6,10) = (2,0,8)$ с истинным ядром $C_{Q_4^{(1)} B} = 3$. Здесь $\varphi\left(Q_4^{(1)}\right) = 1, \varphi\left(Q_4^{(1)} \cdot B\right) = 1$ и $\varphi\left(Q_4^{(1)}\right) \cdot \varphi(B) = \varphi\left(Q_4^{(1)} \cdot B\right)$, т. е. $Q_4^{(1)} \cdot B < P$.

Найдем невязку с учетом знака минус у $Q_n^{(1)}Q^{(2)} = Q_n^{(1)} \cdot B - Q_4^{(1)} = (2,0,8) - (2,8,5) = (0,1,3)$ с расчетным ядром $\bar{C}_{Q^{(2)}} = 1$ и истинным ядром $C_{Q^{(2)}} = 1$.

Так как $\bar{C}_{Q^{(2)}} = C_{Q^{(2)}}$, то получена истинная невязка.

$$Q_1^{(2)} = (0,1,3)/2 \approx (6,0,2)/2 = (3,0,1) \text{ с } C_{Q_1^{(2)}} = 1;$$

$$Q_2^{(2)} = (3,0,1)/2 \approx (2,8,0)/2 = (1,4,0) \text{ с } C_{Q_2^{(2)}} = 1;$$

$$Q_3^{(2)} = (0,6,9)/7 \approx (0,3,10)/7 = (3,3,3) \text{ с } C_{Q_3^{(2)}} = 0;$$

$$Q_4^{(2)} = (6,6,6)/2 \approx (2,2,2)/2 = (1,1,1) \text{ с } C_{Q_4^{(2)}} = 0.$$

Третье промежуточное частное $Q_4 = (1,1,1)$ с $C_{Q_4^{(2)}} = 0$.

На этом деление можно закончить, так как Q_4 будет обязательно равно нулю.

Составим частное:

$\frac{A}{B} = \frac{(5,7,10)}{(3,6,10)} = (0,7,7) - (3,3,3) + (1,1,1) = (5,5,5)$ с расчетным ядром $C_{A/B} = -1 - 0 + 0 - (-1) = 0$.

Действительно,

$$\frac{A}{B} = \frac{(5,7,10)}{(3,6,10)} = \frac{439}{87} = 5\frac{4}{84}.$$

В следующем разделе рассмотрим возможность оптимизации расчета функции ядра.

5. Оптимизация итерационного деления

Как видно из предыдущего раздела, функция ядра играет важную роль при выполнении итерационного деления, возникает необходимость определения знака числа, а также уточнения значения ядра в случае возникновения так называемых критических ядер [11]. Для оптимизации итерационного деления докажем следующую теорему.

Теорема 1. $C\left(\frac{X}{2}\right) = \frac{C(X) - \sum^n{}_{i=1} \; w_i}{\frac{|x_i|_2=1}{2}}$.

Доказательство.

Так как значение функции ядра Акушского вычисляется по формуле (6), тогда:

$$C\left(\frac{X}{2}\right) = \sum_{i=1}^{n} w_i \cdot \left\lfloor \frac{X}{2p_i} \right\rfloor.$$

Рассмотрим функцию ядра в следующем виде:

$$C(X) = \sum_{i=1}^{n} w_i \cdot \left\lfloor 2 \cdot \frac{X}{2p_i} \right\rfloor = \sum_{i=1}^{n} w_i \cdot \left\lfloor 2 \cdot \left(\left\lfloor \frac{X}{2p_i} \right\rfloor + \left\{ \frac{X}{2p_i} \right\} \right) \right\rfloor$$

$$= \sum_{i=1}^{n} w_i \cdot \left\lfloor 2 \left\lfloor \frac{X}{2p_i} \right\rfloor + 2\left\{ \frac{X}{2p_i} \right\} \right\rfloor = 2\sum_{i=1}^{n} w_i \cdot \left\lfloor \frac{X}{2p_i} \right\rfloor + \sum_{i=1}^{n} w_i \cdot \left\lfloor 2\left\{ \frac{X}{2p_i} \right\} \right\rfloor$$

$$= 2 \cdot C\left(\frac{X}{2}\right) + \sum_{i=1}^{n} w_i \cdot \left\lfloor 2\frac{|X|_{2p_i}}{2p_i} \right\rfloor = 2 \cdot C\left(\frac{X}{2}\right) + \sum_{i=1}^{n} w_i \cdot \left\lfloor \frac{|X|_{2p_i}}{p_i} \right\rfloor,$$

так как $|X|_2 = 0$ и $|X|_{p_i} = x_i$, тогда:

$$C(X) = 2 \cdot C\left(\frac{X}{2}\right) + \sum_{i=1}^{n} w_i \cdot \left\lfloor \frac{|x_i|_{2p_i} + x_i}{p_i} \right\rfloor = 2 \cdot C\left(\frac{X}{2}\right) + \sum_{\substack{i=1 \\ |x_i|_2=1}}^{n} w_i.$$

Отсюда следует, что:

$$C\left(\frac{X}{2}\right) = \frac{C(X) - \sum^n{}_{i=1} \; w_i}{\frac{|x_i|_2=1}{2}}.$$

Теорема доказана.

Предложенный подход позволяет увеличить скорость вычисления функции ядра Акушского, а значит увеличить эффективность выполнения итерационного деления в СОК.

6. Оценка эффективности

Для подтверждения свойств предложенной оптимизации метода деления мы реализуем его на языке Python и сравним производительность с классическим итерационным делением. Эксперименты проводились в операционной системе Windows 10 Home Edition на базе компьютера с процессором Intel Core i7-7700HQ 2,80 ГГц, оперативной памятью DDR4 8 ГБ 1196 МГц и твердотельным накопителем SSD 512 ГБ.

Эксперимент заключается в следующем, исследование проводится в два этапа:

Этап А – исследование производительности 4 наборов модулей, размерностью от 8 до 32 бит;

Этап Б – исследование производительности 8 комплектов, от 3 до 10 модулей, размерность каждого модуля 8 бит.

При проведении двухэтапного моделирования были получены временные характеристики каждого метода. В каждом из результатов было получено среднее значение после 100 итераций. Результаты отражены в табл. 1 и 2. Время указано в секундах.

Табл. 1. Результат этапа А (Бит - размер одного набора в битах)
Table 1. Result of stage A (Bit is the size of one set in bits)

Bit	Итерационное деление	Модифицированное итерационное деление
8	0.0000543	0.0000521
16	0.0000628	0.0000598
24	0.0000646	0.0000629
32	0.0000918	0.0000877

Из таблицы видно, что предложенный нами метод, использующий для расчета функции ядра форму, предложенную в разделе 5, в среднем на 4% быстрее. Уменьшение вычислительных затрат приводит к экономии энергии и снижению энергопотребления, что делает его перспективным для вычислительных систем. Кроме того, эффективность алгоритма может привести к снижению требований к аппаратному обеспечению, что еще больше повышает его пригодность для использования в средах с ограниченными ресурсами.

7. Заключение

В данной статье мы исследовали оптимизацию алгоритма деления чисел в системе остаточных классов с использованием функции ядра Акушского. Был разработан более эффективный и быстрый алгоритм расчета функции ядра Акушского, для использования в итерационном делении.

Мы провели эксперименты и показали, что наш метод улучшает производительность итерационного деления. В будущих исследованиях планируется реализовать аппаратную

реализацию предлагаемого алгоритма, что позволит дать более точную оценку времени, потреблении энергии и площади.

Алгоритмы деления в системе остаточных классов, в том числе предложенная нами модификация итерационного деления имеют ограничение, связанное с тем, что результатом деления может быть только целое число. Однако, нами ведутся поиски метода, позволяющего разрешить данную проблему.

Результаты нашей статьи могут быть использованы в приложениях где используется криптосистема RSA, в системах обработки изображений и других областях, где необходимы высокопроизводительные вычисления.

Табл. 2. Результат этапа Б (p[n] - длина набора модулей, где n - количество модулей в наборе)
Table 2. Result of stage B (p[n] is the length of the set of modules, where n is the number of modules in the set)

p[n]	Итерационное деление	Модифицированное итерационное деление
3	0.0000728	0.0000701
4	0.0001046	0.0001022
5	0.0001251	0.0001045
6	0.0001452	0.0001413
7	0.0001586	0.0001459
8	0.0001663	0.0001621
9	0.0002076	0.0001920

Список литературы / References

[1]. Mohan P. V. A., Mohan P. V. A. Residue Number Systems. Cham, Switzerland: Birkhäuser, 2016. – pp. 16-24.

[2]. Chervyakov N., Babenko M., Tchernykh A., Kucherov N., Miranda-López V., Cortés-Mendoza J. M. AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security. Future Generation Computer Systems, vol. 92, 2019, pp. 1080-1092.

[3]. Kasianchuk, M. M., Yakymenko, I. Z., Nykolaychuk, Y. M. (2021). Symmetric cryptoalgorithms in the residue number system. Cybernetics and Systems Analysis, 57(2), 2021, pp. 329-336.

[4]. Tchernykh, A., Schwiegelsohn, U., Talbi, E. G., Babenko, M. (2019). Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. Journal of Computational Science, vol. 36, 2019, P. 100581.

[5]. Червяков, Н. И., Коляда, А. А., Ляхов, П. А., Бабенко, М. Г., Лавриненко, И. Н., Лавриненко, А. В. Модулярная арифметика и ее приложения в инфокоммуникационных технологиях. М.: ФИЗМАТЛИТ, 2017, 440 с. / Chervyakov, N. I., Kolyada, A. A., Lyakhov, P. A., Babenko, M. G., Lavrinenko, I. N., Lavrinenko, A. V. Modular arithmetic and its applications in info-communication technologies. Moscow: Fizmatlit, 2017, 440 p. (in Russian).

[6]. Molahosseini A. S., Sorouri S., Zarandi A. A. E. Research challenges in next-generation residue number system architectures. Computer Science & Education (ICCSE), 7th International Conference, 2012, pp. 1658–1661.

[7]. Червяков Н.И., Ляхов П.А., Оразаев А.Р. Компьютерная оптика, том. 42, вып. 4, 2018 г., стр. 667-678: DOI: 10.18287/2412-6179-2018-42-4-667-678./ Chervyakov N.I., Lyakhov P.A., Orazaev A.R. Computer Optics, 2018, vol. 42, issue 4, pp. 667-678 (in Russian). DOI: 10.18287/2412-6179-2018-42-4-667-678.

[8]. Червяков Н.И., Лавриненко И. Н. Модулярные методы и алгоритмы деления на основе спуска Ферма и итераций Ньютона. Инфокоммуникационные технологии, том. 7, вып. 4, 2009 г., стр. 9–12. / Chervyakov N.I., Lavrinenko I.N. Modular methods and algorithms of division based on Fermat's descent and Newton's iterations. Info-communication Technologies, 2009, vol. 7, issue 4, pp. 9-12 (in Russian).

[9]. Червяков Н.И. Методы, алгоритмы и техническая реализация основных проблемных операций, выполняемых в системе остаточных классов. Инфокоммуникационные технологии, том. 9, вып. 4, 2011 г., стр. 4–12. / Cherviakov N.I. Methods, algorithms and technical implementation of the main problem operations performed in the residual class system. Info-communication Technologies, 2011, vol. 9, issue 4, pp. 4-12 (in Russian).

[10]. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. М., Советское радио, 1968, 440 с. / Akushsky I. Ya., Yuditsky D. I. Computer arithmetic in residual classes. Moscow, Soviet Radio, 1968, 440 p. (in Russian).

[11]. Акушский И. Я., Бурцев В. М., Пак И. Т. О новой позиционной характеристике непозиционного кода и ее приложении. Теория кодирования и оптимизация сложных систем. Алма-Ата, Наука, КазССР, 1977, стр. 8–16. / Akushsky, I.Y., Burtsev, V.M., Pak, I.T. (1977) About the New Positional Characteristic of the Non-Positional Code and Its Application. In Theory of Coding and Optimization of Complex Systems, Alma-Ata, Nauka, KazSSR, 1977, pp. 8–16 (in Russian).

[12]. Shiriaev, E., Kucherov, N., Babenko, M., Lutsenko, V., Al-Galda, S. Algorithm for Determining the Optimal Weights for the Akushsky Core Function with an Approximate Rank. Applied Sciences, 13(18), 2023, 10495. https://doi.org/10.3390/app131810495.

[13]. Акушский И. Я., Бурцев В. М., Пак И. Т. Алгоритмы деления с использованием ядерной характеристики. Теория кодирования и оптимизация сложных систем. Алма-Ата, Наука, КазССР, 1977, стр. 26–33. / Akushsky, I.Y., Burtsev, V.M., Pak, I.T. (1977) Division Algorithms Using Core Characteristics. In Theory of Coding and Optimization of Complex Systems, Alma-Ata, Nauka, KazSSR, 1977, pp. 26–33 (in Russian).

[14]. Diffie W., Hellman M. E. New Directions in Cryptography. IEEE Transactions on Information Theory, 22(18), pp. 644–654. doi:10.1109/TIT.1976.1055638

Информация об авторах / Information about authors

Владислав Вячеславович ЛУЦЕНКО – аспирант, кафедры вычислительной математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВПО «Северо-Кавказский федеральный университет». Сфера научных интересов: высокопроизводительные вычисления, система остаточных классов, умный город, нейронные сети, интернет вещей.

Vladislav Vyacheslavovich LUTSENKO – postgraduate student, Department of Computational Mathematics and Cybernetics, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov, North Caucasus Federal University. Research interests: high-performance computing, residue number system, smart city, neural networks, Internet of Things.

Михаил Григорьевич БАБЕНКО – доктор физико-математических наук, заведующий кафедры вычислительной математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова ФГАОУ ВПО «Северо-Кавказский федеральный университет». Сфера научных интересов: облачные вычисления, высокопроизводительные вычисления, система остаточных классов, нейронные сети, криптография.

Mikhail Grigoryevich BABENKO – Dr. Sci (Phys.-Math.), 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. His research interests include cloud computing, high-performance computing, residue number systems, neural networks, cryptography.

Андрей Николаевич ЧЕРНЫХ получил степень кандидата наук в Институте точной механики и вычислительной техники РАН. В настоящее время он является профессором Центра научных исследований и высшего образования в Энсенаде, Нижняя Калифорния, Мексика. В научном плане его интересуют многоцелевая оптимизация распределения ресурсов в облачной среде, проблемы безопасности, планирования, эвристики и метаэвристики, энергосберегающие алгоритмы, интернет вещей.

Andrei TCHERNYKH received his Cand. Sci. (Phys.-Math.) degree at the Institute of Precise Mechanics and Computer Engineering of the Russian Academy of Sciences. Now he is holding a full professor position in computer science at CICESE Research Center, Ensenada, Baja California, Mexico. He is interesting in grid and cloud research addressing multi-objective resource optimization, both, theoretical and experimental, security, uncertainty, scheduling, heuristics and meta-heuristics, adaptive resource allocation, energy-aware algorithms and Internet of Things.

Мария Анатольевна ЛАПИНА – кандидат физико-математических наук, доцент, доцент кафедры информационной безопасности автоматизированных систем ФГАОУ ВО «Северо-Кавказский федеральный университет». Сфера научных интересов: цифровые технологии, управление информационной безопасностью, процессный подход, образовательный процесс, криптография.

Maria Anatolievna LAPINA – Cand. Sci. (Phys.-Math.), Associate Professor, Associate Professor of the Department of Information Security of Automated Systems of the North Caucasus Federal University. Research interests: digital technologies, information security management, process approach, educational process, cryptography.