# Метод анализа корректирующих способностей кода системы остаточных классов

Н. И. Червяков[1], В. В. Бережной[2], М. А. Дерябин[3], Н. Н. Кучеров[4], А. С. Назаров[5], И. В. Дворянинова[6]
Северо-Кавказский федеральный университет,
Кафедра прикладной математики и математического моделирования
[1]ncherviakov@ncfu.ru, [2]beregnoj@yandex.ru, [3]maderiabin@ncfu.ru, [4]nkucherov@ncfu.ru,
[5]kapitoshking@mail.ru, [6]innadv99@mail.ru

***Аннотация*. Система остаточных классов (СОК) представляет собой один из подходов к построению кодов, исправляющих ошибки. Важными преимуществами СОК являются возможность параллельной обработки частей кода, арифметичность и масштабируемость. Регулируя избыточность кода СОК, можно получать различные свойства корректирующего кода, сочетая эффективность реализации и надежность кодирования. В данной работе предложен метод анализа распределения возможных ошибок по полному диапазону СОК. Данный метод позволяет определить, какими именно корректирующими свойствами будет обладать предложенная СОК, и может быть использован при выборе системы оснований при проектировании конкретных систем хранения и обработки данных.**

***Ключевые слова: избыточная система остаточных классов; коды исправления ошибок; корректирущие способности кодов***

## I. Введение

Многочисленные исследования, проводимые в области обеспечения надежности хранения и обработки информации, убедительно обосновали возможность построения параллельных вычислительных систем, в которых за счет специального кодирования может быть создан иммунитет против самых разнообразных искажений хранимой, обрабатываемой и передаваемой информации. Один из таких способов кодирования представляет арифметика в остаточных классах (модулярная арифметика) [1], которая позволяет использовать параллельную внутреннюю структуру для создания кодов, исправляющих ошибки.

На основе системы остаточных классов (СОК), являющейся ядром модулярной арифметики, стоится множество систем, обеспечивающих надежное хранение, обработку и передачу информации в самых различных областях информационных технологий. Корректирующие свойства СОК используются, например, для обеспечения надежности передачи информации в системах на основе технологии DS-CDMA [2], для обеспечения надежности и безопасности облачных вычислений [3], при проектировании систем надежной гибридной памяти [4]. Являясь с одной стороны полноценной непозиционной системой счисления, что позволяет производить арифметические действия с кодированной информацией, СОК также используется в таких криптографических примитивах, как схемы разделения секрета, которые могут быть использованы для обеспечения безопасности распределенного хранения и обработки данных. Сочетание свойств и особая структура кода делает СОК уникальным и эффективным инструментом для решения многих задач.

Код СОК представляет собой совокупность из $n$ остатков от деления на заранее подобранные модули, являющиеся основаниями, определяющими СОК. Такая структура обеспечивает независимость каждого остатка, что делает СОК хорошо масштабируемой. Для того, чтобы на основе кода СОК построить код исправления ошибок, необходимо к исходному (рабочему) набору оснований добавить дополнительные (контрольные) модули. Корректирующие способности избыточной СОК (ИСОК) хорошо проанализированы в литературе [5–7]. Основой большинства подходов, позволяющих исправлять ошибки по всем модулям СОК, является метод проекций [6]. Сложность локализации многократных ошибок (в нескольких остатках) приводит к появлению множества подходов, направленных на упрощение схемы [5]. При этом, важнейшим вопросом является выбор конкретных модулей СОК, обеспечивающих определенные требуемые корректирующие свойства.

Данная работа посвящена подходу, позволяющему проанализировать структуру и корректирующие свойства ИСОК. В основе предлагаемого метода лежит подробный анализ распределения возможных ошибок по диапазону СОК. Данный подход позволяет подобрать систему оснований СОК для обеспечения требуемых свойств, что позволяет использовать СОК для построения систем с регулируемой избыточностью [8].

## II. Система остаточных классов

Информация, передаваемая по каналам связи или обрабатываемая в вычислительных системах, представляется комбинациями двоичного кода, которые могут быть интерпретированы как числа. Пользуясь этим фактом, для контроля ошибок используется система

Работа выполнена при финансовой поддержке РФФИ, проект № 18-07-00109, и при поддержке Гранта Президента Российской Федерации, проект МК-6294.2018.9

остаточных классов, которая является непозиционной системой счисления. Любое число $A$ в СОК представляется неотрицательными остатками от деления на модули $m_i$, $i = 1,2,\dots,n$: $A = (a_1, a_2, \dots, a_n), a_i = A \bmod m_i = |A|_{m_i}$.

Модулярная арифметика основывается на Китайской теореме об остатках (КТО), утверждающей единственность представления числа в СОК в промежутке $[0, M-1]$, где $M = m_1 \cdot m_2 \cdot \dots \cdot m_n$ – рабочий диапазон представления чисел в СОК, при условии попарной взаимной простоты модулей $m_i$. Величина вычетов $a_i$ по модулю $m_i$ представляет собой целое число в интервале $[0, m_i - 1]$. Одним из важных следствий КТО является ключевой метод перевода чисел из СОК в позиционное представление:

$$A = \left| \sum_{i=1}^{n} a_i B_i \right|_M \quad (1)$$

где $B_i = \frac{M}{m_i} \cdot \left| \left( \frac{M}{m_i} \right)^{-1} \right|_{m_i}$. Данный подход послужил основной множества методов, реализующих как операцию перевода числа из СОК в позиционное представление, так и другие важные операции в СОК.

Являясь системой счисления, СОК позволяет производить с представленными в ней числами ряд операций, многие из которых могут выполняться в параллельной форме независимо по каждому из оснований. При этом малая разрядность остатков позволяет повысить быстродействие обработки данных без существенной избыточности. Равноправность и полная арифметичность остатков СОК дают возможность легко масштабировать код и проводить необходимую реконфигурацию системы в процессе работы.

## III. Виды ошибок и корректирующие способности СОК

Корректирующие свойства системы остаточных классов (СОК) проявляются при введении в нее контрольных (избыточных) модулей [6]. В качестве избыточных модулей берутся числа $m_j$, $j = n+1, n+2, \dots, n+r$, удовлетворяющие условиям КТО, т.е. взаимно-простые между собой и по всей совокупности исходных неизбыточных (рабочих) модулей. Число $R = M \cdot m_{n+1} \cdot \dots \cdot m_{n+r}$ будем называть полным диапазоном СОК.

Методы коррекции ошибок в СОК основаны на базовом принципе [7]: искаженные основания меняют итоговую величину числа так, что оно перестает принадлежать рабочему диапазону $[0, M-1]$. Отслеживая величину числа, представленного в СОК, можно обнаружить наличие ошибки. Определение номеров ошибочных остатков (локализация) и исправление ошибки производится различными методами в зависимости от условий функционирования всей системы [5].

Термин «ошибочный остаток» или «ошибка» в СОК требует дополнительного рассмотрения. Прежде всего следует отметить такие ошибки в двоичных комбинациях по модулям СОК, которые искажают исходное число $a_i$ по основанию $m_i$ так, что результат искажения $a_i^*$ не выходит из диапазона возможных значений остатков $[0, m_i - 1]$. Например, пусть некоторый модуль СОК $m_i$ равен числу 11. Остатками по данному модулю могут быть любые целые числа от 0 до 10. Пусть в ходе обработки информации на некотором этапе остаток по данному модулю был равен $a_i = 6 = 0110_2$. В результате сбоя был искажен младший разряд числа, что привело к следующему значению $a_i^* = 0111_2 = 7$, которое лежит в диапазоне [0,10]. Назовем ошибки такого вида *арифметическими*. Другой вид ошибок выводит остаток из диапазона его представления $[0, m_i - 1]$. Например, остаток $a_i = 6 = 0110_2$ был искажен в старшем разряде $a_i^{**} = 1110 = 14$, что превышает наибольшее значение по модулю $m_i = 11$. Ошибки такого рода назовем *неарифметическими*.

Обнаружение неарифметических ошибок проще, чем арифметических из-за того, что они сразу приводят к появлению запрещенной для заданной системы оснований комбинации. В свою очередь арифметическая ошибка может быть выявлена лишь при переходе к позиционному представлению числа $A$, представленного в СОК [8]. Арифметические ошибки имеют «скрытый» модульный характер в СОК. Их невозможно обнаружить в рамках одного модуля. Рассмотрим далее пример, демонстрирующий корректирующие способности СОК.

*Пример 1.* Пусть задана система оснований $m_1 = 2$, $m_2 = 3$, $m_3 = 5$, $m_4 = 7$, для которой рабочий диапазон определится произведением $M = 210$. Введем одно контрольное основание $m_5 = 11$. Тогда полный диапазон определяется как: $R = m_5 = 2310$. Пусть $A = 114$ есть некоторое число из рабочего диапазона. Представим его остатками по модулям СОК: $A = (0, 0, 4, 2, 4)$. Введем арифметическую ошибку по одному из модулей, например $p_3$. Пусть тогда $A^* = (0, 0, \mathbf{2}, 2, 4)$. Переведем $A^*$ в позиционное представление, пользуясь подходом (1). Величины $B_i$ будут иметь следующие значения: $B_1 = 1155$; $B_2 = 1540$; $B_3 = 1386$; $B_4 = 330$; $B_5 = 210$. Тогда $A^* = 1962$. Сравнивая полученный результат с величиной рабочего диапазона $M = 210$, делаем вывод, что в полученном числе есть ошибка. Условие обнаружения ошибки $A^* > M$ справедливо для всех возможных искажений остатков по одному из оснований только в том случае, если величина контрольного основания $m_j$ превышает величину любого рабочего основания $m_i$ [7].

Для локализации искаженных модулей и исправления ошибок необходимо знать, что происходит с искаженным числом $A^*$ при его преобразовании в позиционную форму, а именно в какую область полного диапазона $[M, R)$ оно попадает, возможно ли при этом определить модуль, по которому произошла ошибка $m_i$ и величину самой ошибки (глубину ошибки) $\Delta a_i$. Для этого предлагается метод исследования распределения ошибок в полном диапазоне СОК, описанный в следующем разделе.

## IV. Метод исследования распределения ошибок в полном диапазоне СОК

При переводе числа $A = (a_1, a_2, \dots, a_n)$ из СОК в позиционное представление согласно (1) необходимо

учесть влияние ошибки в -ом основании $\Delta a_i$ на результат в позиционной системе счисления. Для этого изменим формулу (1) с учетом влияния арифметической ошибки $\Delta a_i$ (глубины ошибки)

$$A^* = |a_1 \cdot B_1 + \cdots + (a_i + \Delta a_i) \cdot B_i + \cdots + a_{n+r} \cdot B_{n+r}|_R$$

откуда

$$A^* = |A + \Delta a_i \cdot B_i|_R \qquad (2)$$

Ключевым свойством ИСОК является то, что обязательно $A^* > M$, то есть полученное ошибочное число превысит рабочий диапазон СОК при условии достаточности количества и размера избыточных модулей. Изменяя значения числа $A$ из рабочего диапазона $[0, M-1]$ и глубины ошибки $\Delta a_i \in [1, m_i - 1]$ можно определить все возможные искаженные числа и то, каким именно образом они распределяются в полном диапазоне СОК. Ошибочные числа $A^*$ при одинаковой глубине ошибки будут расположены последовательно друг за другом с шагом 1 в некотором диапазоне $K_i^s$, где индекс $i$ определяет номер остатка по модулю $m_i$, в котором произошла ошибка, а индекс $s$ равен глубине ошибки $s = \Delta a_i$. Назовём этот диапазон ошибочным. Для определения конкретного ошибочного диапазона достаточно ограничиться вычислением значений ошибки в 0 и в $M-1$, которые и будут определять границы ошибочного интервала $K_i^s$.

*Пример 2.* Пусть задана система рабочих оснований $m_1 = 2$, $m_2 = 3$, $m_3 = 5$, и контрольное основание $m_4 = 7$. Для данной СОК рабочий диапазон $M = 30$, полный диапазон $R = 210$, при этом $B_1 = 105$, $B_2 = 70$, $B_3 = 126$, $B_4 = 120$. Определим ошибочные интервалы $K_i^s$ для всех возможных однократных ошибок, т.е. ошибок по одному основанию, используя выражение (2). Для основания $m_1 = 2$ глубина ошибки принимает только одно значение $\Delta a_1 = 1$. Определим границы формируемого данной ошибкой интервала $K_1^1$: $|0 + 1 \cdot 105|_{150} = 105$, $|29 + 1 \cdot 105|_{150} = 134$. Следовательно, интервал $[105; 134]$ есть рассматриваемый ошибочный интервал $K_1^1$. Аналогично рассчитаем для остальных оснований, результаты сведем в табл. 1.

Наглядное представление диапазонов для данного примера можно видеть на рис. 1 (линия $a$). Данный рисунок точно отображает, что все ошибочные интервалы попадают в диапазон $[M, R]$, что достаточно для установления факта наличия ошибки. Однако, для однозначной локализации ошибки необходимо чтобы искажение любых двух различных чисел из рабочего диапазона не приводило к одинаковому результату, т.е. ошибочные интервалы не должны пересекаться между собой. Для данного набора оснований это условие не выполняется. Например, пусть задано число $A = 19$, которое представлено в данной СОК как $A = (1,1,4,5)$. Ошибка $\Delta a_2 = 1$ искажает представление числа $A$ в СОК следующим образом: $A^* = (1, \mathbf{2}, 4,5)$. Обратный перевод этого числа даст следующий результат: $(1, \mathbf{2}, 4,5) = 89$. Для другого числа $B = 5 = (1,2,0,5)$ ошибка $\Delta a_3 = 4$, что соответствует величине $B^* = (1,2, \mathbf{4}, 5)$, при переводе в позиционное представление приводит к такому же результату $(1, \mathbf{2}, 4,5) = 89$. В данной ситуации невозможно как по позиционному, так и по представлению числа в остатках установить, по какому модулю $m_i$ и с какой величиной $\Delta a_i$ произошла ошибка. Этот факт говорит о неоднозначности локализации ошибки и невозможности её исправления в данном случае.

Рассмотрим другую систему оснований, в которой рабочие основания это $m_1 = 2$, $m_2 = 3$, контрольные – $m_3 = 5$, $m_4 = 7$. Рабочий диапазон $M = 2 \cdot 3 = 6$, полный диапазон $R = 210$. Распределение ошибочных интервалов для такой СОК представлено в табл. 2 и схематически отображено на рис. 1 (линия $b$). Из рисунка видно, что ошибочные интервалы между собой не пересекаются.

ТАБЛИЦА I ГРАНИЦЫ ВСЕХ ВОЗМОЖНЫХ ОШИБОЧНЫХ ИНТЕРВАЛОВ ДЛЯ НАБОРА РАБОЧИХ МОДУЛЕЙ $\{2,3,5\}$ С КОНТРОЛЬНЫМ ОСНОВАНИЕМ 7

| **Модуль с ошибкой** | $\Delta a_i$ | **Границы $K_i^s$** | | $K_i^s$ |
|---|---|---|---|---|
| | | $\mathbf{0}^*$ | $(\boldsymbol{M-1})^*$ | |
| $m_1 = 2$ | 1 | 105 | 134 | $K_1^1$ |
| $m_2 = 3$ | 1 | 70 | 99 | $K_2^1$ |
| | 2 | 140 | 169 | $K_2^2$ |
| $m_3 = 5$ | 1 | 126 | 155 | $K_3^1$ |
| | 2 | 42 | 71 | $K_3^2$ |
| | 3 | 168 | 197 | $K_3^3$ |
| | 4 | 84 | 113 | $K_3^4$ |
| $m_4 = 7$ | 1 | 120 | 149 | $K_4^1$ |
| | 2 | 30 | 59 | $K_4^3$ |
| | 3 | 150 | 179 | $K_4^3$ |
| | 4 | 60 | 89 | $K_4^4$ |
| | 5 | 180 | 209 | $K_4^5$ |
| | 6 | 90 | 119 | $K_4^6$ |

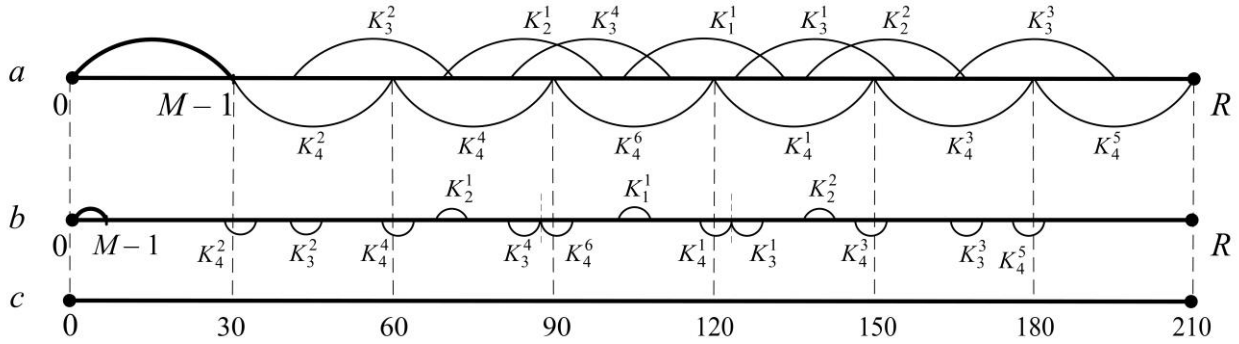


Рис. 1. Распределение ошибочных интервалов по полному диапазону: линия $a$ – для СОК с одним контрольным основанием, линия $b$ – для СОК с двумя контрольными основаниями, линия $c$ – величины чисел

Увеличение количества контрольных оснований и уменьшение рабочего диапазона СОК привело к появлению возможности однозначной локализации ошибки по любому модулю. Очевидно, используя данный подход можно определить наборы оснований, удовлетворяющие требованиям локализации и исправления как однократных, так и многократных ошибок.

ТАБЛИЦА II ГРАНИЦЫ ВСЕХ ВОЗМОЖНЫХ ОШИБОЧНЫХ ИНТЕРВАЛОВ ДЛЯ НАБОРА РАБОЧИХ МОДУЛЕЙ {2,3} С КОНТРОЛЬНЫМИ ОСНОВАНИЯМИ {5,7}

| **Модуль с ошибкой** | $\Delta a_i$ | **Границы $K_i^s$** | | $K_i^s$ |
|---|---|---|---|---|
| | | $\mathbf{0}^*$ | $(\boldsymbol{M-1})^*$ | |
| $m_1 = 2$ | 1 | 105 | 110 | $K_1^1$ |
| $m_2 = 3$ | 1 | 70 | 75 | $K_2^1$ |
| | 2 | 140 | 145 | $K_2^2$ |
| $m_3 = 5$ | 1 | 126 | 131 | $K_3^1$ |
| | 2 | 42 | 47 | $K_3^2$ |
| | 3 | 168 | 173 | $K_3^3$ |
| | 4 | 84 | 89 | $K_3^4$ |
| $m_4 = 7$ | 1 | 120 | 125 | $K_4^1$ |
| | 2 | 30 | 35 | $K_4^3$ |
| | 3 | 150 | 155 | $K_4^3$ |
| | 4 | 60 | 65 | $K_4^4$ |
| | 5 | 180 | 185 | $K_4^5$ |
| | 6 | 90 | 95 | $K_4^6$ |

Алгоритм предлагаемого метода:

1. Задаётся набор рабочих $m_i,\ i = 1,2,\dots,n,$ и контрольных $m_j,\ j = n+1, n+2, \dots, n+r,$ оснований, а так же количество возможных ошибок $t \le r$.

2. Вычисляются все константные параметры, необходимые для расчетов: рабочий диапазон $M$, полный диапазон $R$, константы КТО $B_1, B_2, \dots, B_{n+r}$.

3. На основе выражения (2) определяется левая граница каждого из ошибочных интервалов $K_i^S$, где под $S$ понимается комбинация ошибок $(\Delta a_1, \dots, \Delta a_{n+r})$, в которой ненулевых значение $\Delta a_i$ может быть не больше $t$, соответствующая числу 0: $0^* = \left|\sum_{i=1}^{n+r} \Delta a_i B_i\right|_R$. Величина $(M-1)^*$, соответствующая крайнему значению диапазона, на которое действует та же ошибка, определяется из выражения $(M-1)^* = 0^* + M - 1$.

4. Полученные значения определяют границы ошибочных интервалов $K_i^S = [0^*, (M-1)^*]$.

Количество ошибочных интервалов, формируемое кратными ошибками, определяется выражением

$$N_t = \sum_{i_1=1}^{n+r} \sum_{i_2=i_1+1}^{n+r} \cdots \sum_{i_t=i_{t-1}+1}^{n+r} \left(m_{i_1} - 1\right)\left(m_{i_2} - 1\right) \cdots \left(m_{i_t} - 1\right).$$

Если учитывать ошибки всех кратностей вплоть до $t$, то общее число ошибок необходимо найти сумму $N = N_1 + N_2 + \cdots + N_t$.

Основываясь на методе оценки корректирующих свойств кода СОК, можно подбирать параметры кодирования, оценивая количество неисправляемых ошибок. Предложенный подход позволяет подобрать максимально эффективную с точки зрения избыточности и корректирующих способностей систему оснований, устанавливая при этом в случае необходимости точные позиции и величины неисправляемых ошибок. В большинстве работ, посвященных корректирующим свойствам СОК, утверждается [7], что для исправления $t$ ошибок необходимо добавить $r = 2t$ избыточных оснований, при этом требуя чтобы $m_j > m_i$, где $m_j$ – избыточные основания, $m_i$ – рабочие. Предложенный подход позволяет более гибко подойти к вопросу подбора оснований за счет возможности варьирования полученной системы по избыточности и корректирующим способностям.

## V. ЗАКЛЮЧЕНИЕ

Предложенный метод оценки корректирующих способностей кодов СОК, позволяет наглядно представить распределение арифметических ошибок по остаткам представления числа в полном диапазоне $R$. Получаемое при проведении исследований распределение ошибок позволяет оценить корректирующие способности выбранной системы оснований. При этом, можно видеть каким образом изменение числа и величин избыточных оснований влияет на распределение ошибок и, соответственно на корректирующие способности СОК. Разработанный подход позволяет точно оценить количество неисправляемых ошибок в коде СОК в случае недостаточной избыточности. Используя данный метод относительно несложно подобрать такие системы оснований, которые будут обеспечивать требуемую надежность хранения и обработки данных в реальных системах, использующих СОК.

## СПИСОК ЛИТЕРАТУРЫ


[1] Szabo N.S., Tanaka R.I. Residue arithmetic and its applications to computer technology. McGraw-Hill, 1967.

[2] Hanzo L., Yang L.L., Kuan E.L., Yen K. Single-and multi-carrier DS-CDMA: multi-user detection, space-time spreading, synchronisation, standards and networking. John Wiley & Sons, 2003.

[3] Tchernykh A., Schwiegelsohn U., Talbi E.G., Babenko M.. Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. Journal of Computational Science. 2016.

[4] Haron N. Z., Hamdioui S. Redundant residue number system code for fault-tolerant hybrid memories. ACM Journal on Emerging Technologies in Computing Systems, 2011, Vol. 7., №. 1, pp. 4.

[5] Tay T.F., Chang C.H. Fault-Tolerant Computing in Redundant Residue Number System. Embedded Systems Design with Special Arithmetic and Number Systems. Springer, Cham, 2017, pp. 65-88.

[6] Goh V. T., Siddiqi M. U. Multiple error detection and correction based on redundant residue number systems. IEEE Transactions on Communications, 2008. Vol. 56., No 3, pp. 325-330.

[7] Ding C., Pei D., Saiomaa A. Chinese remainder theorem: applications in computing, coding, cryptography. World Scientific, Singapore, 1996.

[8] Pleshchinskii N., Tormasov A., Tumakov D. Analysis of the Fault Tolerance of the Distributed Data Storage with Controlled Redundancy. Applied Mathematical Sciences, 2015. Vol. 9. No. 141, pp. 7011-7025.