УДК 621.391

Н.И. Червяков, М.Г. Бабенко, П.А. Ляхов, И.Н. Лавриненко

ПРИБЛИЖЕННЫЙ МЕТОД ОПРЕДЕЛЕНИЯ ЗНАКА ЧИСЛА В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ И ЕГО ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

N.I. Chervyakov, M.G. Babenko, P.A. Lyakhov, I.N. Lavrinenko

APPROXIMATE METHOD FOR DETERMINING THE NUMBER SIGN IN RESIDUE NUMBER SYSTEM AND IT'S TECHNICAL SALES

Предложен приближенный метод определения знака числа в системе остаточных классов, основанный на быстром приближенном вычислении относительной величины числа к диапазону системы. Описана техническая реализация метода, отличающаяся простотой реализации. Показано, что применение данного метода позволяет существенно сократить время выполнения основных проблемных операций в системе остаточных классов.

СИСТЕМА ОСТАТОЧНЫХ КЛАССОВ. МОДУЛЯРНАЯ АРИФМЕТИКА. ПОЗИЦИОННАЯ ХАРАКТЕРИСТИКА. ПРИБЛИЖЕННЫЙ МЕТОД. ЗНАК ЧИСЛА.

This paper proposes an approximate method for determining the sign of the number in the residue number systems. This method is based on fast approximate calculation of the relative magnitude of the number to the range of the system. Technical implementation of the proposed method, which is easy to implement. It is shown that the application of the proposed method significantly reduces the time to perform basic operations in the problem of residue number systems.

RESIDUE NUMBER SYSTEM. MODULAR ARITHMETIC. POSITIONAL CHARACTERISICS. APPROXIMATE METHOD. NUMBER SIGN.

Современное состояние развития инфокоммуникационных технологий в области обработки и передачи данных характеризуется интенсивным внедрением новых принципов и подходов к обработке информации. Результаты теоретических и практических разработок отечественных и зарубежных специалистов со всей определенностью указывают на то, что одним из перспективных многообещающих путей решения задач сокращения времени обработки данных и повышения надежности вычислительных средств является применение различных форм параллельной обработки данных, в т. ч. и на основе числовых систем с параллельной структурой. Одно из магистральных направлений среди современных подходов к созданию отказоустойчивых высокопроизводительных средств обработки данных – использование системы остаточных классов (СОК) [1].

Если фиксированный ряд положительных чисел $p_1, p_2, ..., p_n$ назвать основаниями (модулями) СОК, то под системой остаточных классов понимается такая непозиционная система счисления, в которой любое целое положительное число A представляется в виде набора остатков (вычетов) от деления представляемого числа на выбранные основания системы $A = (\alpha_1, \alpha_2, ..., \alpha_n)$, где α_i – наименьшие неотрицательные вычеты (остатки) числа по модулям $p_1, p_2, ..., p_n$. Цифры α_i данного представления по выбранным модулям образуются следующим образом

$$a_i = res + A(\mathrm{mod}\ p_i) = \\ = A - \left[\frac{A}{p_i}\right] p_i, (\forall i \in [1, n]), \quad (1)$$

где $\left[\frac{A}{p_i}\right]$ – целочисленное частное, p_i –

основания (модули) – взаимно-простые числа. В теории чисел доказано, что если $\forall\ i \neq j\ (p_i, p_j) = 1$, то представление (1) является единственным, при условии $0 \le A \le P$, где $P = p_1 p_2 ... p_n = \prod_{i=1}^{n} p_i$ – диапазон представления чисел, т. е. существует число A, для которого:

$$A \equiv \alpha_1 (\mathrm{mod}\ p_1);$$
$$A \equiv \alpha_2 (\mathrm{mod}\ p_2);$$
$$...$$
$$A \equiv \alpha_n (\mathrm{mod}\ p_n).$$

Основным преимуществом такого представления является тот факт, что выполнение операций сложения, вычитания и умножения реализуется очень просто по формуле

$$A * B = (\alpha_1, \alpha_2, ..., \alpha_n) * (\beta_1, \beta_2, ..., \beta_n) =$$
$$= ((\alpha_1 * \beta_1) \mathrm{mod}\ p_1, (\alpha_2 * \beta_2) \mathrm{mod}\ p_2,$$
$$..., (\alpha_n * \beta_n) \mathrm{mod}\ p_n),$$

где $*$ обозначает одну из операций: сложение, вычитание или умножение. Эти операции носят название *модульных*, т. к. для их выполнения в СОК достаточно одного такта обработки численных значений, причем эта обработка происходит параллельно и величина числа в каждом разряде не зависит от других разрядов.

К достоинствам такого представления и обработки чисел относится также малоразрядность остатков, что позволяет эффективно применять табличные методы обработки. Вычислительные системы, построенные на основе СОК, обладают высокой производительностью и надежностью [2]. Однако возникают серьезные трудности при реализации непозиционных процедур, к которым относятся: нахождение вычета (остатка) числа; определение знака числа (в СОК знак числа представлен в неявном виде); сравнение модулярных чисел; определение переполнения; операции деления, масштабирования, расширения, исправления ошибок и др. [3]. Выполнение этих операций является довольно проблематичным. Большинство приложений СОК не требуют использования этих операций. Фундаментальной операцией здесь является определение знака модулярного числа, которое может использоваться при обнаружении переполнения динамического диапазона, сравнении величин чисел, исправлении ошибок и других операций, время выполнения которых может быть уменьшено до времени выполнения модульного деления вместе со сложением, вычитанием и умножением, а также масштабированием вместе с расширением [4].

Необходимо отметить, что даже в тех случаях, когда СОК ограничена приложениями, в которых преобладающими операциями являются сложение и вычитание, нет возможности полностью исключить проблематичные операции. Так, в вычислениях особенно важно масштабирование, т. к. во многих приложениях, для которых СОК особенно хорош, могут встретиться операции, приводящие к росту чисел, которые, в свою очередь, могут привести к переполнению [5]. Поэтому чтобы гарантировать, что все результаты лежат в пределах допустимого диапазона, необходимо проводить контроль в процессе производимых вычислений. В связи с этим возникает необходимость быстрого выполнения указанных выше операций.

Приближенный метод выполнения основных проблемных операций в системе остаточных классов

В настоящее время известны следующие методы определения позиционных характеристик (ПХ) модулярного представления чисел [2, 6]:

ортогональных базисов;

интервальных оценок;

с использованием коэффициентов обобщенной позиционной системы счисления (ОПСС) и др.

Суть метода ортогональных базисов состоит в переходе к позиционному представлению через модуль всей системы $P = p_1 p_2 ... p_n$, что разрушает идею модулярной арифметики. Недостаток метода ортогональных базисов заключается в том, что приходится иметь дело с большими числами ортогональных базисов и, кроме того, действия сложения и умножения надо выполнять в позиционной системе счисления, а полученный результат необходимо вводить в

диапазон вычитанием величины, кратной P, которая определяется рангом числа. Нахождение ранга числа связано с вычислительной сложностью. Метод интервальных оценок сокращает модуль до величины $\frac{P}{p_i}$, и только метод ОПСС позволяет выполнять преобразования по модулю p_i. Метод ОПСС универсальный. К недостаткам этого метода можно отнести его сложность и избыточность ПХ при вычислении некоторых немодульных процедур.

Метод ортогональных базисов и метод ОПСС – точные методы определения ПХ, а метод интервальных оценок – приближенный метод, к характеристике которого относится ПХ номера интервала числа, суть которого состоит в том, что числовой диапазон P разбивается на p_i интервалов

$$\left[j\frac{P}{p_i}, (j+1)\frac{P}{p_i} \right], \text{ при } j = 1, 2, \ldots, p_i - 1.$$

Определение номера интервала, в котором расположено число, позволяет получить оценку немодульного числа по его величине с точностью до величины интервала, что ограничивает область его применения.

Процесс определения знака числа сводится к операции выявления принадлежности интервала, в котором находится число, представленное в СОК, к группе положительных или отрицательных интервалов по заданному p_i, на которые разбит диапазон P. Число интервалов определяется величиной $\frac{P}{p_i}$. В случае если диапазон разбивается на нечетное число интервалов по выбранному модулю p_i, т. е. все основания нечетные, то имеется критический интервал, который является границей между положительными и отрицательными числами. В этом случае критический интервал делится на две части, а процесс определения знака числа при этом сводится к сравнению остатка по модулю p_i, что резко усложняет процесс определения знака числа.

С целью повышения эффективности вычисления ПХ предлагается новый приближенный метод определения ПХ, позволяющий реализовать практически все немодульные процедуры модулярного кода.

Исторически так сложилось, что поиск некоторого компромисса в удовлетворении требований, предъявляемых к ПХ, привел исследователей к введению таких характеристик модулярной алгебры, как ранг, след, нормированный ранг, неточный ранг, ядро числа и др. [1, 5, 7]. Анализ этих ПХ показал, что значение модулярной величины по ним определяется сложно и не всегда однозначно. Кроме того, при выполнении некоторых немодульных операций нет необходимости в точном их определении, а достаточно знать значения в пределах каких-то интервалов, т. е. при определении этих характеристик появляется избыточная информация, которая не используется. Эта идея и подтолкнула к поиску такой ПХ, которая бы не содержала избыточной информации, на нахождение которой требуются дополнительные вычислительные ресурсы.

С другой стороны широко используются методы выполнения проблемных операций, основанные на выборе модулей СОК специального вида [8–10]. Однако такой подход накладывает существенные ограничения на используемую СОК, что сильно затрудняет его использование в некоторых приложениях, особенно в тех, где необходимо динамическое изменение структуры СОК: диапазона и модулей. К таким приложениям относятся, например, криптографическая защита информации и проектирование отказоустойчивых вычислительных систем, функционирующих в СОК.

Анализ немодульных операций показал, что их можно представить точно или приближенно, поэтому методы вычисления ПХ можно разделить на две группы:

методы точного вычисления ПХ;

методы приближенного вычисления ПХ.

Подробное описание методов точного вычисления ПХ можно найти, например, в [1–3, 6]. В данной статье исследуются приближенные методы вычисления ПХ, позволяющие существенно сократить аппаратные и временные затраты, обусловленные операциями, выполняемыми над позиционными кодами уменьшенной разрядности. В связи с этим возникает задача использования приближенного метода при вычислении определенного ряда немодуль-

ных процедур: определения интервалов чисел, знака числа, сравнения, насколько одно число больше или меньше другого, в том случае, когда не требуется знания точного значения.

С целью упрощения процесса сравнения модулярных чисел рассмотрим приближенный метод, позволяющий реализовать основные классы процедур принятия решений: анализ наличия определенного значения в конкретном разряде; проверку равенства (неравенства) двух значений; сравнение двух значений (больше, меньше), обеспечивающих решение основного круга задач, возникающих при аппаратной или программной реализации реальных процессов.

Суть приближенного метода сравнения модулярных чисел основана на использовании относительных величин анализируемых чисел к полному диапазону, определяемому Китайской теоремой об остатках [1], которая связывает позиционное число A с его представлением в остатках (α_1, α_2, ..., α_n), где α_i – наименьшие неотрицательные вычеты числа, относительно модулей системы остаточных классов p_1, p_2, ..., p_n следующим выражением:

$$A = \left| \sum_{i=1}^{n} \frac{P}{p_i} \left| P_i^{-1} \right|_{p_i} \alpha_i \right|_P , \qquad (2)$$

где p_i – модули СОК; $P = \prod_{i=1}^{n} p_i$ – диапазон СОК; $P_i = \frac{P}{p_i} = p_1 p_2 ... p_{i-1} ... \; p_{+1} p_n$ – произведение всех модулей СОК, кроме i-го; $\left| P_i^{-1} \right|$ – мультипликативная инверсия P_i относительно p_i.

Если левую и правую части выражения (2) разделить на величину P, соответствующую диапазону чисел, то получим приближенное значение

$$\left| \frac{A}{P} \right|_1 = \left| \sum_{i=1}^{n} \frac{\left| P_i^{-1} \right|_{p_i}}{p_i} \alpha_i \right|_1 \approx \left| \sum_{i=1}^{n} k_i \alpha_i \right|_1 , \qquad (3)$$

где $k_i = \frac{\left| P_i^{-1} \right|_{p_i}}{p_i}$ – константы выбранной системы; α_i – разряды числа, представленного в СОК, при этом значение каждой суммы будет в интервале [0,1). Конечный результат суммы определяется после суммирования и отбрасывания целой части числа с сохранением дробной части суммы. Дробная часть $F(A) = \left| \frac{A}{P} \right|_1 \in [0,1)$ содержит информацию как о величине числа, так и о его знаке. Если $\left| \frac{A}{P} \right|_1 \in \left[0, \frac{1}{2} \right)$, то число A положительное и $F(A)$ равна величине A, разделенной на P. В противном случае A – отрицательное число и $1-F(A)$ показывает относительную величину числа A по отношению к P. Исследования показали, что функция $F(A)$ может использоваться при разработке алгоритмов вычисления основных проблемных операций в системе остаточных классов [11]. Целая часть числа представляет собой ранг числа, т. е. такую непозиционную характеристику, которая показывает, сколько раз диапазон системы P был превзойден при переходе от представления чисел в системе остаточных классов к его позиционному представлению. При необходимости определение ранга числа может производиться непосредственно в процессе выполнения операции суммирования констант k_i. Дробная часть может быть записана так же как Amod1, потому что $A = \lfloor A \rfloor + A\text{mod}1$ [4]. Количество разрядов дробной части числа определяется максимально возможной разностью между соседними числами. При необходимости точного сравнения необходимо вычислить значение (3), которое является эквивалентом преобразования из СОК в позиционную систему счисления. Для решения задач основных процедур принятия решения достаточно знать приблизительно значения чисел $\left| \frac{A}{P} \right|_1$ и $\left| \frac{B}{P} \right|_1$ по отношению к динамическому диапазону [0,1), которое выполняется достаточно просто, но при этом верно определяется соотношениями $A = B$, $A > B$ или $A < B$.

Итак, приближенный метод вычисления ПХ может описываться следующей последовательностью действий [4]:

1. Вычисление констант СОК $k_i = \frac{\left| P_i^{-1} \right|_{p_i}}{p_i}$ с требуемой точностью.

2. Вычисление приближенных значений $\alpha_i k_i$, где k_i – константы, найденные в п. 1, $1 \le \alpha_i \le p_i - 1$.

3. Вычисление приближенного значения ПХ $\left|\sum_{i=1}^{n} k_i \alpha_i\right|_1$ в интервале [0,1).

Рассмотрим использование информации, содержащейся в $F(A)$, для вычисления проблемных операций в СОК.

Конструируются некоторые правила Ψ_i, $i = 1, ..., 4$, согласно которым вычисляется i-я немодульная операция (определение знака числа, сравнение чисел, обнаружение ошибки и переполнения, а также локализация ошибочного разряда).

Правило Ψ_1. Определение знака числа в случае, если $p_1 = 2$:

если $\left|\frac{A}{P}\right|_1 < \frac{1}{2}$, то число положительное;

если $\left|\frac{A}{P}\right|_1 > \frac{1}{2}$, то число отрицательное.

Правило Ψ_2. Сравнение модулярных чисел A и B:

если $\left|\frac{A}{P}\right|_1 - \left|\frac{B}{P}\right|_1 = 0$, то $A = B$;

если $\left|\frac{A}{P}\right|_1 - \left|\frac{B}{P}\right|_1 > 0$, то $A > B$;

если $\left|\frac{A}{P}\right|_1 - \left|\frac{B}{P}\right|_1 < 0$, то $A < B$.

Правило Ψ_3. Обнаружение ошибки и переполнения динамического диапазона:

если $\left|\frac{\overline{A}}{P_{\text{изб}}}\right|_1 < \left|\frac{M}{P_{\text{изб}}}\right|_1$, то ошибки нет, где $\overline{A}$ – искаженное число; $P_{\text{изб}} = p_{n+1}p_{n+2}P$ – избыточный диапазон при двух избыточных модулях p_{n+1} и p_{n+2}; $M = P = \prod_{i=1}^{n} p_i$ – рабочий диапазон;

если $\left|\frac{\overline{A}}{P_{\text{изб}}}\right|_1 \ge \left|\frac{M}{P_{\text{изб}}}\right|_1$, то ошибка есть и установлено переполнение динамического диапазона.

Правило Ψ_4. Локализация неисправного канала:

если $\left|\frac{\overline{A_i}}{P_i}\right|_1 < \left|\frac{M_i}{P_i}\right|_1$, то в разряде i нет ошибки;

если $\left|\frac{\overline{A_i}}{P_i}\right|_1 \ge \left|\frac{M_i}{P_i}\right|_1$, то в разряде i есть ошибка, где $\overline{A_i} = (\alpha_1, \alpha_2, ..., \alpha_{i1}, \alpha_{i+1}, .., \alpha_n, \alpha_{n+}1, \alpha_{n+2})$ – проекция искаженного числа $\overline{A}$, $M_i = (m_1, m_2, ..., m_{i-1}, m_{i+1}, ..., m_n, m_{n+1}, m_{n+2})$ – проекция рабочего диапазона.

Алгоритм определения знака модулярного числа и его техническая реализация

Основой для выполнения немодульных операций, указанных в правилах Ψ_2–Ψ_4, является немодульная операция определения знака числа (правило Ψ_1). Пусть для вычислений используются константы $\overline{k}_i^{(N)}$, представленные в виде двоичных дробей, содержащих N двоичных знаков после запятой, полученные отбрасыванием всех остальных разрядов, начиная с N+1-го. Тогда точные значения констант K_i будут лежать в диапазоне

$$\overline{k}_i^{(N)} \le k_i \le \overline{k}_i^{(N)} + 2^{-N}. \qquad (4)$$

Для определения ПХ числа в СОК необходимо вычислять величину $\left|\sum_{i=1}^{n} k_i \alpha_i\right|_1$ из формулы (3).

Утверждение. Наименьшее значение $N_{\min}$, при котором восстановление позиционной величины числа из представления в СОК по формуле (3) будет корректным, равно

$$N_{\min} = \left\lceil \log_2 (P\rho) \right\rceil, \qquad (5)$$

где $\rho = -n + \sum_{i=1}^{n} p_i$.

Доказательство. Из (4) следует, что точное значение величины $\sum_{i=1}^{n} k_i \alpha_i$ из формулы (3) будет лежать в диапазоне

$$\sum_{i=1}^{n} \overline{k}_i^{(N)} \alpha_i \le \sum_{i=1}^{n} k_i \alpha_i \le \le \sum_{i=1}^{n} \overline{k}_i^{(N)} \alpha_i + 2^{-N} \sum_{i=1}^{n} \alpha_i. \qquad (6)$$

Так как величина $\left|\sum_{i=1}^{n} k_i \alpha_i\right|_1$ соответствует точному местоположению числа A на числовой оси, то для однозначного (точного) определения величины числа A не-

обходимо таким образом подобрать параметр N, чтобы в произвольный интервал $\left[\sum_{i=1}^{n}\overline{k}_i^{(N)}\alpha_i, \sum_{i=1}^{n}\overline{k}_i^{(N)}\alpha_i + 2^{-N}\sum_{i=1}^{n}\alpha_i\right]$ попадало лишь одно возможное число из диапазона СОК. Это требование равносильно условию $2^{-N}\sum_{i=1}^{n}\alpha_i < \frac{1}{P}$.

Если обозначить $\rho = -n + \sum_{i=1}^{n} p_i$, то наименьшее значение $N_{\min}$, при котором возможно точное восстановление позиционной формы числа с использованием формулы (3), определяется формулой $N_{\min} = \lceil \log_2(P\rho) \rceil$. Утверждение доказано.

При снижении точности вычислений (уменьшении величины N) диапазон (6) будет увеличиваться так, что в него будут попадать несколько чисел. Однако при решении задачи определения знака числа с использованием правила Ψ_1 возможно использование таких расширенных диапазонов, если учитывать появление зон ошибочного определения знака числа.

Пусть в процессе вычислений используются константы, округленные до N двоичных знаков после запятой, $1 \le N < N_{\min}$. Тогда в интервал $\left[\sum_{i=1}^{n}\overline{k}_i^{(N)}\alpha_i, \sum_{i=1}^{n}\overline{k}_i^{(N)}\alpha_i + 2^{-N}\sum_{i=1}^{n}\alpha_i\right]$ попадет не более $2^{-N}P\sum_{i=1}^{n}\alpha_i$ чисел СОК. Так как $0 \le \alpha_i \le p_{i-1}$, то

$$\max\left\{2^{-N}P\sum_{i=1}^{n}\alpha_i\right\} = 2^{-N}P\left(-n + \sum_{i=1}^{n}p_i\right) = 2^{-N}P\rho.$$

Ввиду увеличения интервала будет происходить наложение его границ на ту область числовой оси, где определенный по формуле $\left\{\sum_{i=1}^{n}\overline{k}_i^{(N)}\alpha_i\right\}$ и реальный знак числа в СОК из формулы $\left\{\sum_{i=1}^{n}k_i\alpha_i\right\}$ могут не совпадать. На рис. 1 *а* показаны возможные направления появления зон неопределенности при округлении констант [12]. На рис. 1 *б* показано появление зон неопределенности при отбрасывании разрядов вместо округления. В последнем случае формируются только две зоны неопределенности вместо четырех. Это позволяет в два раза уменьшить количество проверяемых условий в процессе выполнения алгоритма, что позволяет существенно упростить его формулировку и реализацию. Стрелками отмечены направления роста зон ошибочного определения знака числа при снижении точности. Зонам, изображенным на рис. 1 *б*, соответствуют следующие диапазоны:

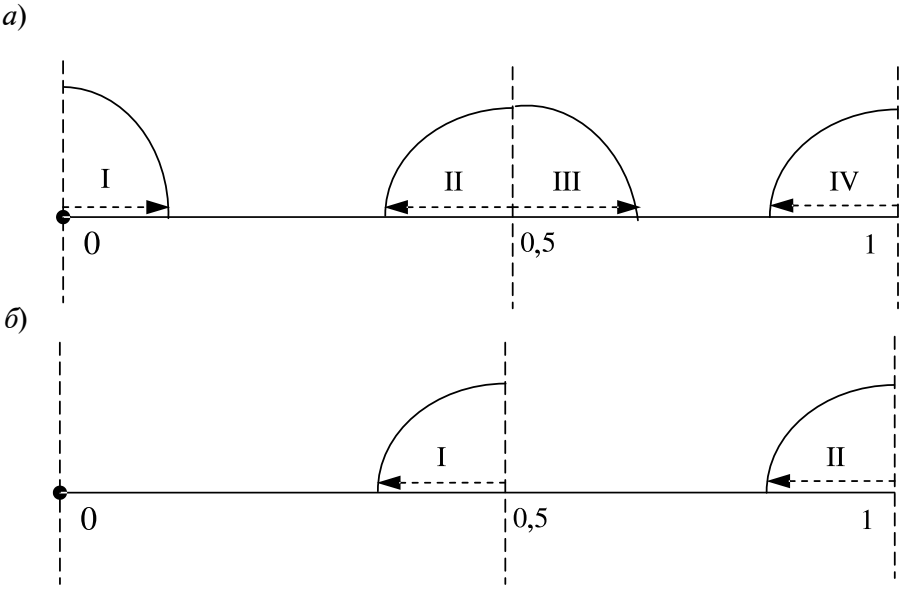

Рис. 1. Появление зон ошибочного определения знака числа в СОК:
а – при округлении констант; *б* – при отбрасывании разрядов

зона ошибочного определения знака I: $[0{,}5 - 2^{-N}\rho,\ 0{,}5)$;

зона ошибочного определения знака II: $[1 - 2^{-N}\rho,\ 1)$.

Для определения знака числа необходимо выполнить проверку двух условий:

если $0 < \frac{A}{P} < 0{,}5 - 2^{-N}\rho$, то число A – положительное;

если $0{,}5 < \frac{A}{P} < 1 - 2^{-N}\rho$, то число A – отрицательное.

В том случае, если не выполняется ни одно из указанных условий, число попадает в одну из зон неопределенности и требуется дополнительная итерация для получения точного результата с использованием $N_{\min}$ знаков после запятой (5).

На рис. 2 приведена схема для определения знака модулярного числа, представленного по четырем модулям p_i, $i = 1, 2, 3, 4$. Схема содержит входные регистры RG_i, $\forall i = [1...4]$ для временного хранения остатков чисел по соответствующим модулям, просмотровые таблицы LUT_i, $\forall i = [1...4]$ для хранения произведений $\left|\frac{P_i^{-1}}{p_i}\right|_{p_i} \cdot \alpha_i$ и параллельный сумматор.

Схема работает следующим образом. Код числа A, для которого необходимо определить интервал, что равносильно определению знака числа, поступает на входные регистры RG_i в двоичном коде (каждый разряд СОК кодируется двоичным кодом). Сигналы с выходов регистров поступают на входы просмотровых таблиц LUT. В просмотровых таблицах хранятся произведения констант k_i и остатков α_i, то есть $\frac{\left|P_i^{-1}\right|_{p_i}}{p_i}\alpha_i$, представленных в естественной форме двоичной дроби в дополнительном коде. Количество элементов памяти (N) просмотровых таблиц определяется выражением $N = \sum_{i=1}^{n} p_i$.

Выходные сигналы просмотровых таблиц в дополнительном двоичном коде поступают на вход сумматора, в котором уже записана константа $0{,}5–2^{-N}\rho$, во время начальной установки. (Дополнительный код используется для того чтобы операцию вычитания заменить операцией сложения). Знак результата сложения определяет интервал (первый или второй), что соответственно определяет знак числа.

Моделирование алгоритма

Для моделирования разработанного алгоритма определения знака числа с использованием ПХ на основе приближенного метода были выбраны следующие СОК:

1. COK_1 {2, 3, 5, 7}.
2. COK_2 {7, 17, 19, 29}.
3. COK_3 {2, 3, 5, 11, 13, 19, 23, 29, 79}.

Данный выбор объясняется тем, что COK_1 наиболее просто и наглядно отображает свойства вычислений в остаточных классах и используется многими авторами (например [6]); COK_2 и COK_3 позволяют

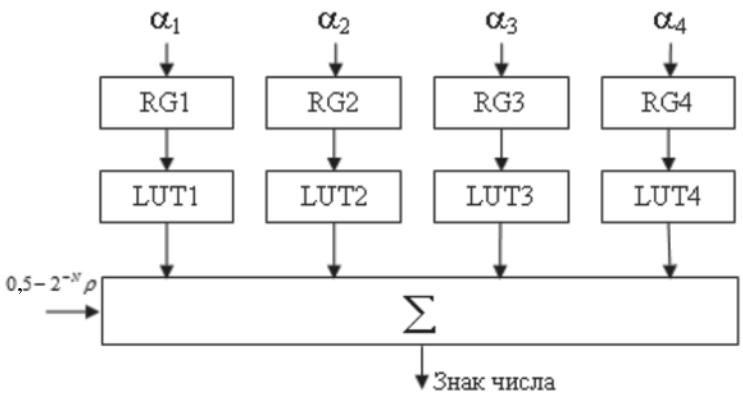

Рис. 2. Схема определения знака числа

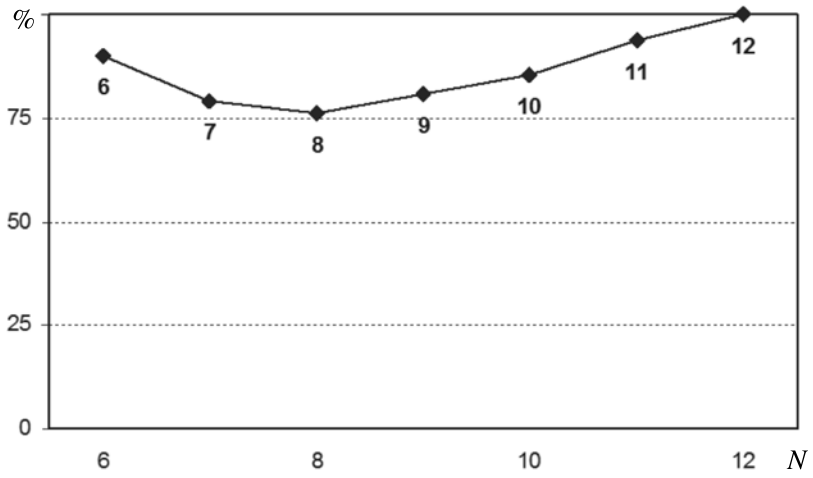

Рис. 3. График зависимости времени работы алгоритма (% от максимального) от точности вычислений для COK_1

представить числовые диапазоны в 16 бит и 32 бит соответственно [2].

При моделировании замерялось время выполнения операции определения знака числа при постепенном уменьшении величины $N \le N_{min}$. Для COK_1 имеем следующие характеристики: $P = 210$, $\rho = 16$, $N_{min} = 12$. Результаты моделирования разработанного алгоритма для COK_1 показаны на рис. 3. На рисунке изображено, насколько (в процентах) снижается время работы алгоритма при выборе различных значений N. При снижении величины N от $N_{min} = 12$ до $N = 8$ происходит уменьшение времени работы алгоритма за счет уменьшения разрядности обрабатываемых чисел. Однако при дальнейшем уменьшении N начинается увеличение времени выполнения алгоритма. Этот факт объясняется тем, что при $N < 8$ зоны неопределенности при определении знака числа увеличиваются настолько, что уже значительная доля чисел из диапазона СОК требует дополнительной (уточняющей) итерации алгоритма с максимальной точностью вычислений.

Наилучший показатель времени работы алгоритма достигается при $N = 8$, в этом случае время работы алгоритма составляет $\approx 76{,}2$ % от максимального времени работы алгоритма при $N_{min} = 74$. Таким образом, использование предложенного алгоритма для COK_1 позволяет сократить время определения знака числа примерно в 1,31 раз по сравнению с точными методами.

Для COK_2 имеем: $P = 65569$, $\rho = 68$,

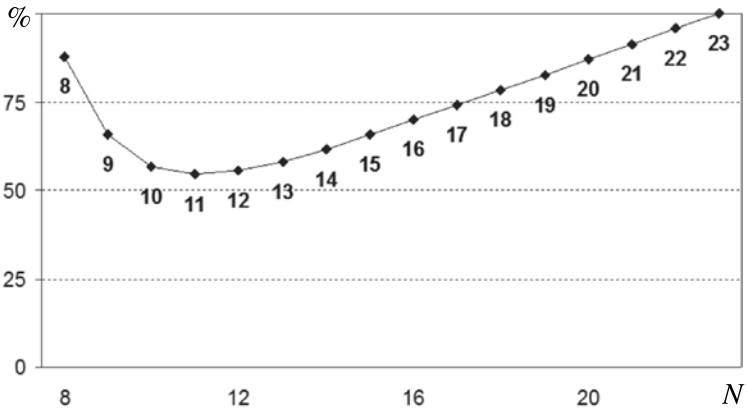

Рис. 4. График зависимости времени работы алгоритма (% от максимального) от точности вычислений для COK_2

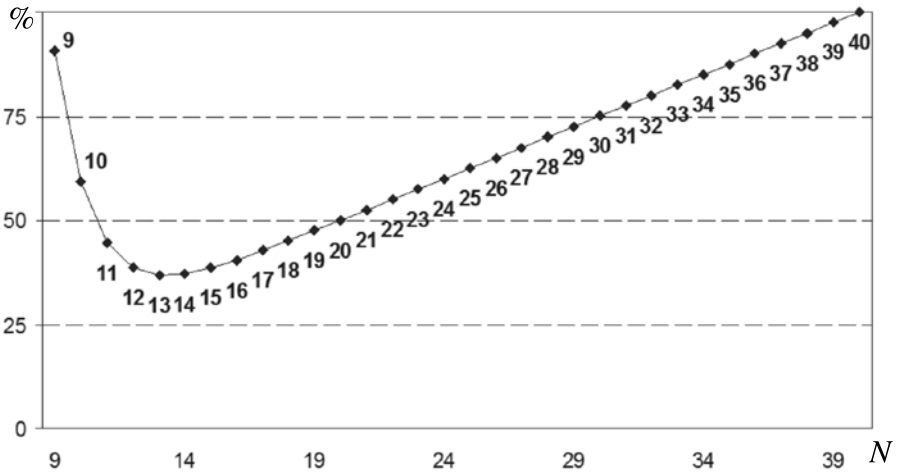

Рис. 5. График зависимости времени работы алгоритма (% от максимального) от точности вычислений для $СОК_3$

$N_{min} = 23$. Результаты моделирования разработанного алгоритма для $СОК_2$ показаны на рис. 4. Поведение кривой скорости алгоритма аналогично случаю $СОК_1$. Снижение величины N от $N_{min} = 23$ до $N = 11$ приводит к уменьшению времени работы алгоритма. При дальнейшем уменьшении N начинается увеличение времени выполнения алгоритма.

Самая высокая скорость работы алгоритма наблюдается при $N = 11$, в этом случае время работы алгоритма составляет $\approx 54{,}5$ % от максимального времени работы алгоритма при $N_{min} = 74$. Таким образом, использование предложенного алгоритма для $СОК_2$ позволяет сократить время определения знака числа примерно в 1,84 раз по сравнению с точными методами.

Наконец, $СОК_3$ имеет параметры: $P = 4295006430$, $\rho = 175$, $N_{min} = 40$. Результаты моделирования разработанного алгоритма для $СОК_3$ показаны на рис. 5. Снова наблюдается снижение времени работы алгоритма при снижении точности до $N = 13$. При дальнейшем снижении время работы алгоритма увеличивается.

Самая высокая скорость работы алгоритма наблюдается при $N = 13$, в этом случае время работы алгоритма составляет $\approx 36{,}8$ % от максимального времени работы алгоритма при $N_{min} = 40$. Таким образом, использование предложенного алгоритма для $СОК_2$ позволяет сократить время определения знака числа примерно в 2,72 раз по сравнению с точными методами.

Противоречие между вычислительной сложностью определения основных проблемных процедур в СОК и их быстродействием разрешено путем замены абсолютных величин их относительными значениями и простотой их вычисления, которая сохраняет адекватную связь числовых значений модулярных величин с их представлениям в СОК и позволяет существенно повысить скорость выполнения немодульных операций. Благодаря этому применение СОК может дать значительные преимущества не только в тех приложениях, в которых основная доля вычислений приходится на точное умножение, возведение в степень больших чисел в сочетании со сложением и вычитанием, но и в которых часто появляется необходимость в делении либо сравнении и определении знака числа, а также при проверке не «выходят» ли результаты за пределы допустимых значений и др.

Решена фундаментальная проблема реализации основных проблемных операций в СОК, которые ранее определяли наибольший вклад в алгоритмическую сложность и сдерживали широкое применение СОК при разработки новых классов вычислительных систем. Предложенная техническая реализация приближенного метода определения знака числа проста для применения

на практике, поэтому внедрение полученных результатов позволит расширить область применения модулярной арифметики. Полученные новые результаты эффективного выполнения немодульных процедур являются развитием теории математических основ разработки и проектирования высокопроизводительных и надежных вычислительных систем, функционирующих в системе остаточных классов. Применение предложенного алгоритма позволяет сократить время выполнения проблемной операции в 1,31–2,72 раза (в зависимости от диапазона СОК) по сравнению с известными методами.

Работа выполнена при финансовой поддержке РФФИ (проекты № 13-07-00478-а) и ФЦП № 14.B37.21.1128.

СПИСОК ЛИТЕРАТУРЫ

1. **Акушский, И.Я.** Машинная арифметика в остаточных классах [Текст] / И.Я. Акушский, Д.И. Юдицкий. –М.: Сов. радио, 1968. –440 с.

2. **Червяков, Н.И.** Модулярные параллельные вычислительные структуры нейропроцессорных систем [Текст] / Н.И. Червяков, П.А. Сахнюк, А.В. Шапошников, С.А. Ряднов. – М.: Физматлит, 2003. –288 с.

3. **Червяков, Н.И.** Нейрокомпьютеры в остаточных классах [Текст] / Н.И. Червяков, П.А. Сахнюк, А.В. Шапошников, А.Н. Макоха; под ред. А.И. Галушкина. –М.: Радиотехника, 2003. – 272 с.

4. **Червяков, Н.И.** Методы, алгоритмы и техническая реализация основных проблемных операций, выполняемых в системе остаточных классов [Текст] / Н.И. Червяков // Инфокоммуникационные технологии. –2011. –№ 4. –С. 4–12.

5. **Burgess, N.** Scaling an RNS Number Using the Core Function [Text] / N. Burgess // XVI IEEE Symp. Computer Arithmetic, Santiago de Compostella. –2003. –P. 262–271.

6. **Omondi, A.** Residue Number Systems: Theory and Implementation [Text] / A. Omondi, B. Premkumar. –Imperial College Press, 2007. –296 p.

7. **Амербаев, В.М.** Теоретические основы машинной арифметики [Текст] / В.М. Амербаев. – Алма-Ата: Наука, 1976. –324 с.

8. **Gallaher, D.** The digit parallel method for fast RNS to weighted number system conversion for specify moduli (2^k–1, 2^k, 2^k+1) [Text] / D. Gallaher, F.E. Petry, P. Sirnivasan // IEEE Trans. on Circuits and System II: Analog and Digital Signal Processing. – 1997. –Vol. 44. –№ 1. –P. 53–57.

9. **Tomczak, T.** Fast sign detection for rns (2^n–1, 2^n, 2^n+1) [Text] / T. Tomczak // IEEE Trans. on Circuits and Systems-I: Regular papers. – 2008. – Vol. 55. – № 6. – P. 1502–1511.

10. **Wang, Y.** Adder based residue to binary numbers converters for (2^n–1, 2^n, 2^n+1) [Text] / Y. Wang, M. Aboulhamid, H. Shen // IEEE Trans. Signal Processing. – 2002. – Vol. 50. – № 7. – P. 1772–1779.

11. **Hung, C.Y.** An Approximate Sign Detection Method for Residue Numbers and its Application to RNS Division [Text] / C.Y. Hung, B. Parhami // Computers Math. Applic. – 1994. – Vol. 27. – № 4. – P. 23–35.

12. **Червяков, Н.И.** Метод определения знака числа в системе остаточных классов на основе приближенных вычислений [Текст] / Н.И. Червяков, П.А. Ляхов. – Нейрокомпьютеры: разработка, применение. – 2012. – № 12. –С. 56–64.

REFERENCES

1. **Akushskii I.Ia., Iuditskii D.I.** Mashinnaia arifmetika v ostatochnykh klassakh. – Moscow: Sov. radio, 1968. – 440 s. (rus).

2. **Cherviakov N.I., Sakhniuk P.A., Shaposhnikov A.V., Riadnov S.A.** Moduliarnye parallel'nye vychislitel'nye struktury neiroprotsessornykh sistem. – Moscow: Fizmatlit, 2003. – 288 s. (rus)

3. **Cherviakov N.I., Sakhniuk P.A., Shaposhnikov A.V., Makokha A.N.** Neirokomp'iutery v ostatochnykh klassakh; pod red. A.I. Galushkina. – Moscow: Radiotekhnika, 2003. – 272 s. (rus)

4. **Cherviakov N.I.** Metody, algoritmy i tekhnicheskaia realizatsiia osnovnykh problemnykh operatsii, vypolniaemykh v sisteme ostatochnykh klassov / Infokommunikatsionnye tekhnologii. – 2011. – № 4. – S. 4–12. (rus)

5. **Burgess N.** Scaling an RNS Number Using the Core Function / XVI IEEE Symp. Computer Arithmetic, Santiago de Compostella. – 2003. – P. 262–271.

6. **Omondi A., Premkumar B.** Residue Number Systems: Theory and Implementation. – Imperial College Press, 2007. – 296 p.

7. **Amerbaev V.M.** Teoreticheskie osnovy mashinnoi arifmetiki. – Alma-Ata: Nauka, 1976. – 324 s. (rus)

8. **Gallaher D., Petry F.E., Sirnivasan P.** The digit parallel method for fast RNS to weighted

number system conversion for specify moduli (2k-1, 2k, 2k+1)/ IEEE Trans. on Circuits and System II: Analog and Digital Signal Processing. – 1997. – Vol. 44. – № 1. – P. 53–57.

9. **Tomczak T.** Fast sign detection for rns (2n-1, 2n, 2n+1)/ IEEE Trans. on Circuits and Systems-I: Regular papers. – 2008. – Vol.55. – № 6. – P. 1502–1511.

10. **Wang Y., Aboulhamid M., Shen H.** Adder based residue to binary numbers converters for (2n-1, 2n, 2n+1)/ IEEE Trans. Signal Processing. – 2002. – Vol.50. – № 7. – P. 1772–1779.

11. **Hung C.Y., Parhami B.** An Approximate Sign Detection Method for Residue Numbers and its Application to RNS Division / Computers Math. Applic. – 1994. – Vol. 27. – № 4. – P. 23–35.

12. **Cherviakov N.I., Liakhov P.A.** Metod opredeleniia znaka chisla v sisteme ostatochnykh klassov na osnove priblizhennykh vychislenii / Neirokomp'iutery: razrabotka, primenenie. – 2012. –№ 12. –S. 56–64. (rus)

ЧЕРВЯКОВ Николай Иванович – *заведующий кафедрой прикладной математики и математического моделирования Института математики и естественных наук Северо-Кавказского федерального университета, доктор технических наук, профессор.*
355000, Россия, г. Ставрополь, пр. Кулакова, д. 2.
E-mail: k-fmf-primath@stavsu.ru

CHERVYAKOV, Nikolay I. *North-Caucasian Federal University.*
355000, prosp. Kulakova 2, Stavropol, Russia.
E-mail: k-fmf-primath@stavsu.ru

БАБЕНКО Михаил Григорьевич – *доцент кафедры прикладной математики и математического моделирования Института математики и естественных наук Северо-Кавказского федерального университета, кандидат физико-математических наук.*
355000, Россия, г. Ставрополь, пр. Кулакова, д. 2.
E-mail: whbear@yandex.ru

BABENKO, Michael G. *North-Caucasian Federal University.*
355000, prosp. Kulakova 2, Stavropol, Russia.
E-mail: whbear@yandex.ru

ЛЯХОВ Павел Алексеевич – *доцент кафедры прикладной математики и математического моделирования Института математики и естественных наук Северо-Кавказского федерального университета, кандидат физико-математических наук.*
355000, Россия, г. Ставрополь, пр. Кулакова, д. 2.
E-mail: ljahov@mail.ru

LYAKHOV, Pavel A. *North-Caucasian Federal University.*
355000, prosp. Kulakova 2, Stavropol, Russia.
E-mail: ljahov@mail.ru

ЛАВРИНЕНКО Ирина Николаевна – *доцент кафедры высшей алгебры и геометрии Института математики и естественных наук Северо-Кавказского федерального университета, кандидат физико-математических наук.*
355017, Россия, г. Ставрополь, пр. Кулакова, д. 2.

LAVRINENKO, Irina N. *North-Caucasian Federal University.*
355000, prosp. Kulakova 2, Stavropol, Russia.