УДК 004.04:004.415.23 DOI: 10.17587/it.24.782-790

С. А. Инютин, д-р техн. наук, проф. e-mail: inyutin_int@mail.ru,
Московский авиационный институт (Национальный исследовательский университет) (МАИ)

Комплексирование систем счисления для многоразрядных вычислительных процессов

Введены конструкции и обоснованы варианты сочетаний внешней и внутренней систем счисления — математические основы методов вычислений в больших компьютерных диапазонах (многоразрядных вычислений). Систематизированы и обоснованы форматы данных для многоразрядных вычислений. Детально исследована конструкция, содержащая внутреннюю позиционную и внешнюю степенную модулярную системы. На основе кодонезависимых алгоритмов Pepen и Lucas-Lehmer введены кодозависимые модулярные модификации алгоритмов, имеющих меньшую временную сложность за счет больших затрат памяти. Исследованы линейные пространства модулярных векторов. Выявлены особенности и обоснованы преимущества модулярной системы с квадратичным диапазоном и соответствующих параллельных компьютерных форматов данных для модулярных реконфигурируемых вычислительных систем. Показанные преимущества обосновывают использование внешней квадратичной модулярной и внутренней позиционной системы счисления в пределах машинного диапазона для разрабатываемого программного инструментария организации вычислительных процессов в больших компьютерных диапазонах на компьютерах SIMD-архитектуры.

Ключевые слова: *многопроцессорные реконфигурируемые системы SIMD-архитектуры, многоразрядный вычислительный процесс, кодозависимые алгоритмы, линейная и квадратичная сложность алгоритмов, параллельные компьютерные форматы квадратичной модулярной системы*

Введение

При численных расчетах в быстро развивающейся прикладной теории чисел, при математическом моделировании в космологии, а также при проектировании точных и сложных технологических процессов в ряде отраслей техники и производства появляются отдельные вычислительные задачи, требующие целочисленных многоразрядных вычислений или вычислений в больших компьютерных диапазонах. На современном уровне развития вычислительной техники под этой проблематикой понимаются вычисления с числовыми величинами, содержащими более миллиона десятичных разрядов.

Важной научной и практической задачей является разработка программно-алгоритмических основ вычислительного инструментария для многоразрядных вычислений, относящихся к задачам с большой алгоритмической сложностью. Для организации этих вычислений на суперкомпьютерах и серийных вычислительных системах требуется специальная организация вычислительных процессов. Справиться с возникающими проблемами и выполнить числовые расчеты невозможно без специальной арифметики и производных типов данных на базе соответствующей компьютерной алгебры [1, 2]. В этих процессах для ряда вычислительных задач значения операндов операций (переменных функций), в частности целочисленных, значительно, на много (сто, тысячу и более) порядков превышают максимум типового компьютерного диапазона серийной вычислительной техники.

С длиной машинного слова связан типовой компьютерный диапазон представления целых и рациональных числовых величин. В машинных словах или их совокупностях отображаются числовые данные в различных внутренних форматах. Данные в этих форматах обрабатываются в операционных регистрах процессора. Обработка форматов данных в регистрах поддерживается аппаратными и микропрограммными способами. Машинная арифметика для обработки этих форматов числовых данных, не превышающих максимума типового компьютерного диапазона, имеет максимальную скорость выполнения и является основной в компьютере [1].

1. Базовая терминология вычислений в больших диапазонах

Для корректного анализа проблем организации вычислительных процессов в диапазонах, значительно превышающих типовой компью-

терный диапазон серийной вычислительной техники, введем специальную терминологию.

Диапазонные числа — числовые величины для машинных расчетов, отображаемые в разрядной сетке процессора компьютера. Например, арифметический целочисленный компьютерный диапазон для 32-разрядного процессора есть отрезок $[-2^{31}, 2^{31} - 1]$.

Большие числа — конечные вектора с компонентами, являющимися диапазонными числами. *Сверхбольшие числа* — конечные или бесконечные последовательности больших чисел, полученные на этапах проблемных алгоритмов, как правило, в виде вычетов от больших чисел по модулю большого числа. Программная поддержка чисел двойной разрядности (или точности) не меняет приведенной классификации.

Для многоразрядных вычислений или расчетов в больших компьютерных диапазонах выделим внешнюю и внутреннюю системы счисления, учитывая, что в вычислительной технике нашли применение позиционные и модулярные системы счисления [1, 2].

Внутреннюю систему счисления можно выбрать позиционную или модулярную и использовать для представления диапазонных чисел, отображаемых в 32 разрядах (4 байтах) или в 64 разрядах (8 байтах).

Внешнюю систему счисления также можно выбрать позиционную или модулярную и использовать для представления больших чисел.

Представляет интерес поиск и обоснование оптимального сочетания внешней и внутренней систем счисления. Затем для них строят соответствующие арифметики и производные форматы числовых данных. Базовые операции в различных арифметиках имеют разную сложность.

Рассмотрим кратко варианты комплексирования внешней и внутренней систем счисления, приведенные в таблице.

Вариант	Внешняя система	Внутренняя система
1	Позиционная	Позиционная
2	Позиционная	Модулярная
3	Модулярная	Позиционная
4	Модулярная	Модулярная

1. ПозПоз — типовой вариант, исследованный в работах [1, 2].

2. ПозМод — проблематичный вариант, имеющий большую алгоритмическую сложность при реализации [1].

3. МодПоз — вариант представляет наибольший интерес, в нем для внутреннего представления используется одинарная или квадратичная модулярная система счисления, что позволяет организовать оптимальную эмуляцию на серийных машинах с аппаратно-поддерживаемой быстрой позиционной целочисленной 16-, 32-, 64-разрядной арифметикой на компьютерах соответствующей разрядности.

4. МодМод — вариант представляет интерес для модулярных систем со сжатием (перераспределением избыточности) внутреннего представления. Для этого варианта, требующего отдельного описания, ориентируясь на стандартный двоичный размер байта, можно предложить взаимно простые модули — основания модулярной арифметики: $\{p_1, p_2, p_3, p_4, p_5\} = \{253, 255, 256, 257, 259\}$.

2. Область приложения многоразрядных вычислений

Детально проанализируем вариант 3, наиболее перспективный на наш взгляд. Целесообразно рассмотреть особенности комплексирования систем счисления, опираясь на один из классов алгоритмов, предназначенных для операций с операндами из сверхбольших диапазонов. К этому классу относятся алгоритмы тестирования на простоту чисел специального вида — числа Ферма и Мерсенна, мультипликативное каноническое разложение которых неизвестно. Задачи тестирования чисел на простоту имеют важные приложения, но большую алгоритмическую и временную сложность [2, 3]. Соответствующие критерии простоты дают теоремы.

Теорема 1. Pepen. Необходимое и достаточное условие простоты числа Ферма $F_n = 2^{2^n} + 1$ есть выполнение сравнения:

$$3^{\frac{F_n-1}{2}} \equiv -1 (\mathrm{mod}\, F_n).$$

Теорема 2. Lucas-Lehmer. Число Мерсенна $M_q = 2^q - 1$, $q \in N$, — простое тогда и только тогда, когда M_q делит нацело W_{q-2} член последовательности $\{W_n\}$, построенной следующим образом:

$$W_0 = 4, W_{n+1} = W_n^2 - 2, n \geqslant 0.$$

Приведем тексты алгоритмов *Pepen* и *Lucas-Lehmer* в кодонезависимой постановке с указанием сложности на алгоритмическом псевдоязыке (в нотации П. Нодена [3]). Алгоритмы можно считать типовыми для вычислительных процессов, использующих большие компьютерные диапазоны.

Algorithm A-1. *Pepen.*
1. $m \leftarrow 0$, $A \leftarrow 3$
2. $A \leftarrow A^2 (mod\ F_n), m \leftarrow m + 1$
3. *if* $2^n - m = 0$ *then end else goto* 2
4. *end: if* $A = 2^{2^n}$ *then* F_n *— prime else* F_n *— not prime*

Complexity: time $= 2^n \cdot (t_m + t_{\text{mod}}) = 2^n \cdot (n \cdot 2^n + 2 \cdot 2^n)$

Algorithm A-2. *Lucas-Lehmer.*
1. $m \leftarrow 0$, $A \leftarrow 4$
2. $A \leftarrow A^2 - 2 (\text{mod}\ M_n)$, $m \leftarrow m + 1$
3. *if* $n - 2 - m = 0$ *then end else goto* 2
4. *end: if* $u \equiv 0 (\text{mod}\ M_n)$ *then* M_n *— prime else* M_n *— not prime*

Complexity: time $= (n - 2) \cdot (t_m + t_{\text{mod}}) = (n - 2)(n \cdot \log_2 n + 3 \cdot n)$

Характерной особенностью этих алгоритмов является однотипная последовательность операций с применением арифметики больших чисел на каждом шаге алгоритма:

— возведение в квадрат большого числа, полученного на предыдущем шаге алгоритма;

— выполнение при необходимости арифметических операций с квадратом большого числа;

— вычисление вычета от результата предыдущего шага алгоритма, т. е. квадрата большого числа по большому модулю с неизвестным каноническим разложением.

Критерий останова этих алгоритмов заключается в сравнении больших чисел — результатов, полученных на каждом шаге алгоритма, с константой или повторение однотипных шагов алгоритма фиксированное число раз и затем сравнение большого числа с константой.

Перспективным путем решения проблем, возникающих при вычислениях в больших компьютерных диапазонах, является выбор варианта 3, т. е. использование для внешней системы модулярного представления данных и соответствующей модулярной арифметики. Внутренняя система является аппаратно поддерживаемой позиционной. При этом возникает проблема конвертации кодонезависимых проблемных вычислительных алгоритмов в форму модулярных проблемных алгоритмов [4, 5]. Введем модификации алгоритмов *Pepen* и *Lucas-Lehmer* тестирования на простоту чисел специального вида Ферма и Мерсенна, ориентированные на использование модулярной компьютерной арифметики [2]. Из оценок следует, что это позволяет уменьшить их временную сложность, хотя организация вычислительного процесса усложняется и затраты специальным образом организованной памяти увеличиваются [6].

Algorithm A-3. *Pepen for residue codes.*
1. $|P - F_n| \rightarrow \min$, $P = \prod_{i=1}^{k} p_i > F_n$, p_i *— prime*
2. $F_n (\text{mod}\ P^2) = 2^{2^n} + 1 (\text{mod}\ P^2)$
3. $m \leftarrow 0$, $A (\text{mod}\ P^2) \leftarrow 3 (\text{mod}\ P^2)$
4. $A \leftarrow |A^2|_{F_n} (\text{mod}\ P^2)$, $m \leftarrow m + 1$
5. *if* $2^n - m = 0$ *then end else goto* 4
6. *end: if* $A = F_n - 1 (\text{mod}\ P^2)$ *then* F_n *— prime else* F_n *— not prime*

Complexity (one processor): $time_1 = 2^n \cdot (t_m + t_{\text{mod}}) = (2^n)(O_1(k) + O_2(k)) + O_3(k)$

Complexity (k — modular processor): $time_k = O(2^n)$

Algorithm A-4. *Lucas-Lehmer for residue codes.*
1. $|P - M_n| \rightarrow \min$, $P = \prod_{i=1}^{k} p_i > M_n$, p_i *— prime*
2. $M_n = 2^n - 1 (\text{mod}\ P^2)$
3. $m \leftarrow 0$, $A (\text{mod}\ P^2) \leftarrow 4 (\text{mod}\ P^2)$
4. $A (\text{mod}\ P^2) \leftarrow |A^2 (\text{mod}\ P^2) - 2 (\text{mod}\ P^2)|_{M_n} (\text{mod}\ P^2)$, $m \leftarrow m + 1$
5. *if* $n - 2 - m = 0$ *then end else goto* 4
6. *end: if* $A = 0 (\text{mod}\ P^2)$ *then* M_n *— prime else* M_n *— not prime*

Complexity (one processor): $time_1 = (n - 2) \cdot (t_m + t_{mod}) = (n - 2)(O_1(k) + O_2(k)) + O_3(k))$

Complexity (k — modular processor): $time_k = O(n - 2)$

3. Свойства инструментальной вычислительной алгебры

Переход к использованию модулярной арифметики требует уточнения терминологии применительно к математическим и алгоритмическим основам программного инструментария на базе модулярной арифметики.

Вычислительная проблема — формулировка математической расчетной задачи или комплекса взаимосвязанных задач в кодонезависимой постановке, т. е. независимой от системы счисления, необходимой для представления операндов и арифметики для оперирования с ними.

Вычислительная задача — формулировка математической расчетной задачи или комплекса взаимосвязанных задач в кодозависимой постановке, в частности, модулярная вычислительная задача ориентирована на вычисления с модулярными величинами и на операции модулярной арифметики.

В общем случае под *модулярной степенной системой* (k-й степени) будем понимать множество типов модулярных данных, способ ввода арифметического аддитивного и мультипликативного диапазонов, алгоритмы выполнения модульных операций, выбор количественной характеристики отношения порядка и связанных с ней алгоритмов выполнения немодульных операций. *Степень модулярной системы* при указанных условиях определяет число четырехбайтных машинных слов, необходимых для хранения одного степенного вычета.

Конкретизируем некоторые используемые понятия, полагая, что модулярные системы одинарного (первой степени) диапазона исследованы в достаточной степени [1, 2, 5].

Модулярная система, ориентированная на эмуляцию на серийном компьютере, должна удовлетворять некоторым требованиям [8]:

- иметь максимально возможные значения модулей для минимизации присущей модулярной арифметике информационной избыточности, неизбежной при отображении на двоичные конечные разрядные сетки компьютера;
- значения модулей системы должны лежать в пределах, "удобных" для вычислений с модулярными компонентами на 32-разрядных компьютерах;
- использовать знаковый арифметический целочисленный диапазон стандартной компьютерной позиционной арифметики 32-разрядных компьютеров, являющейся "быстрой" благодаря аппаратной или микропрограммной поддержке.

Под термином "удобства" вычислений будем понимать минимизацию затрат операционного оборудования или машинного времени для рассматриваемой задачи на выполнение машинных операций, а также последующую минимизацию затрат оперативной и долговременной памяти. Для уменьшения информационной избыточности модулярной системы, отображаемой в двоичных разрядных сетках 32-разрядных компьютеров, целесообразно применение простых модулей, лежащих в диапазоне $(2^{30}, 2^{31} - 1]$. Оценка Чебышева дает более 50 млн простых чисел в этом диапазоне [2].

Целесообразно назвать числовой величиной элемент множества целых чисел. Для рассматриваемых задач значение числовой величины на много порядков превышает максимум целочисленного машинного диапазона современных компьютеров с ограниченной разрядной сеткой.

Целочисленная числовая величина отображается (представлена) в любой системе счисления вектором с конечным числом компонент:

$$A \leftrightarrow (a_1, \ldots, a_i, \ldots, a_n),$$

где $A \in \mathbf{Z}|\mathbf{Z}_P$ принадлежит подмножеству целых чисел или кольцу по модулю P — произведению оснований системы счисления; компоненты $a_i \in p|p_i$ принадлежат кольцу по модулю p или p_i — одиночным основаниям системы счисления.

Для биективности отображения необходимо ограничение значений числовой величины некоторым конечным диапазоном и фиксацией системы счисления, так как из представления значение числовой величины однозначно можно вычислить только в соответствующей системе счисления.

Обратное отображение компонент вектора представления в значение числовой величины однозначно определяет разложение числовой величины в сумму компонент представления с некоторыми числовыми коэффициентами, детерминированными выбранной системой счисления.

В модулярной системе числовые величины представлены модулярными векторами с компонентами-вычетами по простым (взаимно простым) модулям для системы одинарного диапазона или по степеням простых модулей для систем степенных диапазонов:

$$A \ (\mathrm{mod}\ P^m) \leftrightarrow (a_1(\mathrm{mod}\ p_1^m), \ldots, a_i(\mathrm{mod}\ p_i^m), \ldots, a_n(\mathrm{mod}\ p_n^m)),$$

где $a_i \equiv |A|_{p_i^m} (\mathrm{mod}\ p_i^m)$ — степенной вычет числовой величины по одному из модулей (оснований) степенной модулярной системы.

Рассмотрим метрическое пространство векторов с модулярными компонентами. Для удобства рассмотрения ограничимся случаем $m = 2$, т. е. квадратичной модулярной системой, которую будем обозначать МС(P^2).

Для векторов с модулярными компонентами известно остаточное расстояние (аналог метрики Хэмминга), введенное для задач помехозащитного кодирования и учитывающее характер модульной ошибки [1].

Определение. Остаточное расстояние $d(A, B)$ между двумя модулярными векторами $A(\mathrm{mod}\ P^2)$, $B(\mathrm{mod}\ P^2)$ есть остаточный вес модульной разности двух модулярных векторов:

$$d(A, B) = \sum_{i=1}^{n} \delta\left(|a_i - b_i|_{p_i^2}\right),$$

где символ Кронекера

$$\delta\left(|a_i - b_i\ |_{p_i{}^2}\right) = \begin{cases} 1 \text{ при } a_i \neq b_i, \\ 0 \text{ при } a_i = b_i. \end{cases}$$

Остаточное расстояние является метрикой, для него выполняются соответствующие аксиомы.

Введем аналоги евклидового расстояния на множестве n-мерных векторов через их скалярное произведение, применимые для анализа модулярных вычислительных процессов.

Определение. Первое скалярное произведение двух модулярных величин — векторов A, $B(\mathrm{mod}\ P^2)$ из $\mathrm{MC}(P^2)$ есть число

$$s_1 = \sum_{i=1}^{n} a_i \cdot b_i = (A, B).$$

Замечание. $s_1 \leqslant n(p_i^2 - 1)^2$.

Определение. Второе скалярное произведение двух модулярных величин — векторов A, $B(\mathrm{mod}\ P^2)$ из $\mathrm{MC}(P^2)$ есть число

$$s_2 = \sum_{i=1}^{n} |a_i|_{p_i^2} \cdot |b_i|_{p_i^2}.$$

Замечание 1. $s_2 \leqslant n(p_i^2 - 1)^2$.

Замечание 2. $s_2 = \sum_{i=1}^{n} |a_i|_{p_i^2}^2$ при $A = B(\mathrm{mod}\ P^2)$.

Введем два типа расстояний на множестве модулярных векторов, связанных со скалярными произведениями.

Определение. Евклидово расстояние между двумя модулярными величинами — векторами A, $B(\mathrm{mod}\ P^2)$ из $\mathrm{MC}(P^2)$ есть число

$$\varepsilon(A, B) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} = \sqrt{(A - B, A - B)}.$$

Евклидово расстояние есть метрика, так как не отличается от соответствующей математической конструкции в евклидовом пространстве векторов с целочисленными компонентами.

Определение. Модулярное расстояние между двумя модулярными величинами — векторами A, $B(\mathrm{mod}\ P^2)$ из $\mathrm{MC}(P^2)$ есть число

$$l(A, B) = \sqrt{\sum_{i=1}^{n} |a_i - b_i|_{p_i^2}^2} = \sqrt{(|A - B|_{p_i^2}, |A - B|_{p_i^2})}.$$

Для модулярного расстояния докажем следующую теорему.

Теорема Т-3. Модулярное расстояние между двумя модулярными векторами является метрикой.

Доказательство. Используя арифметическое значение квадратного корня, по определению получим:

1. $l(A, B) \geqslant 0$.
2. $l(A, B) = 0$, если $A \equiv B(\mathrm{mod}\ P^2)$.
3. $l(A, B) = l(B, A)$.
4. Неравенство треугольника рассмотрим подробнее.

Действительно, выполняется следующее равенство:

$$|x_i - y_i|_{p_i^2} = \|x_i - z_i|_{p_i^2} + |z_i - y_i|_{p_i^2}|_{p_i^2} =$$

$$= |x_i - z_i|_{p_i^2} + |z_i - y_i|_{p_i^2} - 0|p_i^2.$$

Следовательно

$$|x_i - z_i|_{p_i^2} + |z_i - y_i|_{p_i^2} \geqslant |x_i - y_i|_{p_i^2}.$$

Возведем обе части неравенства в квадрат и просуммируем:

$$\sum_{i=1}^{n} (|x_i - z_i|_{p_i^2}^2 + 2|x_i - z_i|_{p_i^2}|z_i - y_i|_{p_i^2} + |x_i - y_i|_{p_i^2}^2) \geqslant$$

$$\geqslant \sum_{i=1}^{n} |x_i - y_i|_{p_i^2}^2.$$

Применим неравенство Коши—Буняковского, которое для соотношений по модулям имеет следующий вид:

$$\sum_{i=1}^{n} (|x_i - z_i|_{p_i^2}|z_i - y_i|_{p_i^2}) \leqslant$$

$$\leqslant \sqrt{\sum_{i=1}^{n} |x_i - z_i|_{p_i^2}^2} \sqrt{\sum_{i=1}^{n} |z_i - y_i|_{p_i^2}^2}.$$

Получим:

$$\sqrt{\sum_{i=1}^{n} |x_i - z_i|_{p_i^2}^2} + \sqrt{\sum_{i=1}^{n} |z_i - y_i|_{p_i^2}^2} \geqslant$$

$$\geqslant \sqrt{\sum_{i=1}^{n} |x_i - y_i|_{p_i^2}^2}.$$

Следствие. Первое и второе скалярные произведения определяют метрики, а не нормы [7].

Для кодозависимых алгоритмов, ориентированных на модулярную арифметику, важно соотношение между модульными и немодульными операциями в линейных и циклических частях алгоритмов. Степень сложности модульных и немодульных операций может меняться в различных модулярных системах, что ставит задачу выбора оптимальной степени модулярной системы. Причем эта сложность зависит как от свойств самой модулярной системы, так и от компьютерной базы реализации программного инструментария [2, 9].

Покажем целесообразность выбора квадратичной модулярной системы, опираясь на удобство программно-алгоритмической реализации на 32-разрядных IBM-совместимых серийных компьютерах.

Рассмотрим проблему эффективности выполнения немодульных операций, опираясь на внутренние свойства самой квадратичной модулярной системы.

Числовая величина в квадратичной модулярной системе имеет следующее P-разложение:

$$A(\mathrm{mod}\, P^2) = |A|_P + K \cdot P(\mathrm{mod}\, P^2),$$

где $0 < |A|_P, K < P$.

Пусть для числового значения модулярной величины выполняется неравенство $0 < A < P^2$, тогда для модулярной величины

$$A(\mathrm{mod}\, P^2) \leftrightarrow (\ldots, a_i(\mathrm{mod}\, p_i^2), \ldots).$$

В модулярной системе квадратичного диапазона введем три представления и разложение модулярной величины, а также рассмотрим связи между ними.

Определение. Первое (естественное) представление модулярной величины есть выражение следующего вида:

$$A(\mathrm{mod}\, P^2) \leftrightarrow (\ldots, a_i(\mathrm{mod}\, p_i^2), \ldots),$$
$$\text{или } A(\mathrm{mod}\, P^2) \leftrightarrow a_i(\mathrm{mod}\, p_i^2),$$

где $a_i \equiv A(\mathrm{mod}\, p_i^2)$.

Будем при необходимости использовать краткую запись, отображая знаком $\leftrightarrow$ биективность отображения числовой величины в пространство образов — модулярных представлений числовой величины.

Определение. Второе (каноническое) представление модулярной величины есть выражение следующего вида:

$$A(\mathrm{mod}\, P^m) \leftrightarrow (\ldots, a_{i0} + a_{i1} \cdot p_i(\mathrm{mod}\, p_i^2), \ldots)$$
$$\text{или } A(\mathrm{mod}\, P^2) \leftrightarrow a_{i0} + a_{i1} \cdot p_i(\mathrm{mod}\, p_i^2),$$

где

$a_{i0} \equiv A(\mathrm{mod}\, p_i)$,

$a_{i0} \equiv a_i(\mathrm{mod}\, p_i) \equiv |a_i|_{p_i}(\mathrm{mod}\, p_i^2)$,

$a_{i1} = [a_i/p_i] = (a_i - |a_i|_{p_i})/p_i$.

Каноническое представление применяется для выполнения арифметических операций, в частности деления в МС(P^2). Для канонического представления двухкомпонентный вектор (a_{i0}, a_{i1}) хранится в двух машинных словах серийного компьютера при выборе оснований модулярной системы, меньших максимального значения машинного диапазона.

Каноническому представлению соответствует разложение модулярной величины $A(\mathrm{mod}\, P^2) = A_0 + A_1 \cdot P(\mathrm{mod}\, P^2)$, причем неравенство $A_0 < P$ в общем случае не выполняется.

Определение. Третье представление (согласованное с P-разложением) модулярной величины есть

$$A(\mathrm{mod}\, P^2) \leftrightarrow (\ldots, s_i + k_i \cdot p_i(\mathrm{mod}\, p_i^2), \ldots)$$
$$\text{или } A(\mathrm{mod}\, P^2) \leftrightarrow s_i + k_i \cdot p_i(\mathrm{mod}\, p_i^2),$$

где

$s_i \equiv |A|_P(\mathrm{mod}\, p_i^2) \equiv s_{i0} + s_{i1} \cdot p_i(\mathrm{mod}\, p_i^2)$;

$s_i \neq |a_i|_{p_i}(\mathrm{mod}\, p_i^2)$, $s_{i0} = |a_i|_{p_i}(\mathrm{mod}\, p_i^2)$;

$P(\mathrm{mod}\, p_i^2) \leftrightarrow g_i \cdot p_i(\mathrm{mod}\, p_i^2) = \overline{p}_i(\mathrm{mod}\, p_i^2)$;

$A_1(\mathrm{mod}\, P) = K(\mathrm{mod}\, P) = (A - |A|_P)/P(\mathrm{mod}\, P)$;

$K(\mathrm{mod}\, P) \leftrightarrow k_i(\mathrm{mod}\, p_i) \neq g_i\overline{k}_i + s_{i1}(\mathrm{mod}\, p_i) \equiv$

$\equiv a_{i1}(\mathrm{mod}\, p_i)$;

$g_i = |P/p_i|_{p_i}(\mathrm{mod}\, p_i)$, $h_i = |P/p_i|_{p_i}^{-1}(\mathrm{mod}\, p_i)$;

$\overline{p}_i = g_i \cdot p_i(\mathrm{mod}\, p_i^2)$, $\overline{k}_i = h_i \cdot k_i(\mathrm{mod}\, p_i^2)$.

Согласованное представление применяется в алгоритмах вычисления количественных характеристик отношения порядка для модулярных величин, в частности, с использованием алгоритма исчерпывания [2].

При рассмотрении мы ограничились случаем $m = 2$, но все предыдущее обобщается

на случай произвольной модулярной системы m-степени.

Сказанное выше является основой методов преобразования канонического в согласованное представление модулярной величины. Такое преобразование уменьшает алгоритмическую сложность в алгоритмах вычисления количественных характеристик отношения порядка, а это, в свою очередь, уменьшает сложность процедур выполнения немодульных операций в проблемных вычислительных алгоритмах, ориентированных на применение модулярной арифметики.

4. Преимущества модулярной квадратичной арифметики

Покажем ряд преимуществ модулярной системы квадратичного диапазона при выполнении операций модулярной арифметики.

При выполнении эмулируемых арифметических операций аддитивного алгебраического сложения и мультипликативной операции умножения двух модулярных операндов нормальным этапом является переполнение типового машинного диапазона. Переполнение вызывает аппаратное прерывание. Временные затраты на обработку прерывания минимальные, если факт переполнения однозначно формирует числовой результат без анализа кратности переполнения. Такой анализ требует дополнительных операций и временных затрат на выполнение алгоритма для обработки прерывания.

Пусть квадратичный диапазон $P^2 = \prod_{i=1}^{n} p_i^2$ задан совокупностью простых оснований $\{\ldots, p_i, \ldots\}$, а $A(\mathrm{mod}\ P^2), B(\mathrm{mod}\ P^2)$ — модулярные величины в квадратичном диапазоне, представленные векторами с соответствующими компонентами, которые являются наименьшими неотрицательными вычетами от числовой величины по квадратам модулей-оснований модулярной системы.

Рассмотрим аддитивную операцию алгебраического сложения двух модулярных величин, имеющих разложения в канонические представления:

$$A(\mathrm{mod}\ P^2) = A_0 + A_1 \cdot P(\mathrm{mod}\ P^2) \leftrightarrow$$
$$\leftrightarrow a_i^0 + a_i^1 \cdot p_i (\mathrm{mod}\ p_i^2);$$
$$B(\mathrm{mod}\ P^2) = B_0 + B_1 \cdot P(\mathrm{mod}\ P^2) \leftrightarrow$$
$$\leftrightarrow b_i^0 + b_i^1 \cdot p_i (\mathrm{mod}\ p_i^2).$$

Модулярные величины в одинарном модулярном диапазоне, являющиеся коэффициентами позиционного аддитивного разложения, имеют следующий вид:

$$A_0(\mathrm{mod}\ P) \leftrightarrow a_i^0 (\mathrm{mod}\ p_i), A_1(\mathrm{mod}\ P) \leftrightarrow$$
$$\leftrightarrow a_i^1 (\mathrm{mod}\ p_i).$$

Результат алгебраического сложения двух модулярных величин должен иметь вид:

$$A + B(\mathrm{mod}\ P^2) \equiv C(\mathrm{mod}\ P^2) \equiv$$
$$\equiv C_0 + C_1 \cdot P(\mathrm{mod}\ P^2) \leftrightarrow c_i^0 + c_i^1 \cdot p_i (\mathrm{mod}\ p_i^2).$$

Операции целочисленной знаковой арифметики в процессоре 32-разрядного компьютера, учитывая ограниченность разрядной сетки, выполняются по модулю $m = 2^{31}$. Оценим число прерываний по переполнению при суммировании максимальных значений компонент векторов:

$$c_i^0 + c_i^1 \cdot p_i \leqslant p_i - 2 + 1 \cdot p_i + (p_i - 2)p_i + 1 \cdot p_i^2,$$
$$\left|a_i^0 + b_i^0\right|_m + r_0 m, \left|a_i^1 + b_i^1\right|_m + r_1 m,$$

где r_0, $r_1 = 0|1$.

Эти покомпонентные соотношения в модулярной системе примут вид:

$$\left\|a_i^0 + b_i^0\right|_m\Big|_{p_i} + r_0 |m|_{p_i} + e_1 p_i + e_2 p_i;$$
$$\left\|a_i^1 + b_i^1\right|_m + r_1 m\Big|_{p_i} = \left\|\left\|a_i^1 + b_i^1\right|_m\Big|_{p_i} + r_1 |m|_{p_i}\right|_{p_i},$$

где e_0, $e_1 = 0|1$.

Переполнения, вызывающие прерывания, отражены в коэффициентах r_0, $r_1 = 0|1$. Кратности переполнений имеют единичные значения, следовательно, соответствующие числовые значения определяются фактом переполнения.

В модулярных системах больших степеней сложность операций растет, например, в МС(P^3) число процедур обработки переполнений удваивается при выполнении операций над соответствующими представлениями:

$$A(\mathrm{mod}\ P^3) = A_0 + A_1 \cdot P + A_2 \cdot P^2 (\mathrm{mod}\ P^3) \leftrightarrow$$
$$\leftrightarrow a_i^0 + a_i^1 \cdot p_i + a_i^2 \cdot p_i^2 (\mathrm{mod}\ p_i^3);$$
$$B(\mathrm{mod}\ P^3) = B_0 + B_1 \cdot P + B_2 \cdot P^2 (\mathrm{mod}\ P^3) \leftrightarrow$$
$$\leftrightarrow b_i^0 + b_i^1 \cdot p_i + b_i^2 \cdot p_i^2 (\mathrm{mod}\ p_i^3).$$

Рассмотрим операцию умножения двух модулярных величин, заданных каноническими представлениями в модулярной системе квадратичного диапазона. Результат должен быть получен в виде:

$$A \times B(\mathrm{mod}\ P^2) \equiv C(\mathrm{mod}\ P^2) \equiv$$
$$\equiv C_0 + C_1 \cdot P(\mathrm{mod}\ P^2) \leftrightarrow c_i^0 + c_i^1 \cdot p_i(\mathrm{mod}\ p_i^2).$$

Рассмотрим процесс формирования результата:

$$(a_i^0 + a_i^1 \cdot p_i(\mathrm{mod}\ p_i^2)) \times (b_i^0 + b_i^1 \cdot p_i(\mathrm{mod}\ p_i^2)) \leftrightarrow$$
$$\leftrightarrow c_i^0 + c_i^1 \cdot p_i + c_i^2 \cdot p_i^2 + c_i^3 \cdot p_i^3,$$

где $c_i^0, c_i^1, c_i^2, c_i^3$ — коэффициенты разложения модулярных компонент произведения с учетом переполнений.

Результат умножения в стандартной компьютерной арифметике операндов, имеющих машинный формат целых чисел, заносится в два последовательных операционных регистра или соседних машинных слова, что позволяет последующим делением получить целую часть и наименьший неотрицательный вычет от результата произведения. В этом случае аппаратно формируемый сигнал переполнения сигнализирует о выходе за пределы удвоенного машинного диапазона без информации о кратности переполнения.

Оценим влияние отдельных компонент операндов на число переполнений при формировании коэффициентов $c_i^0, c_i^1, c_i^2, c_i^3$, используя при умножении максимальные значения компонент векторов-операндов:

$$a_i^0 \cdot b_i^0(\mathrm{mod}\ p_i^2) \equiv c_i^0 + c_i^{01} \cdot p_i(\mathrm{mod}\ p_i^2) \leqslant$$
$$\leqslant 1 + (p_i - 2)p_i \leqslant m^2;$$
$$a_i^0 \cdot b_i^1 + a_i^1 \cdot b_i^0(\mathrm{mod}\ p_i^2) \equiv c_i^{10} + c_i^{11} \cdot p_i +$$
$$+c_i^{12} \cdot p_i^2(\mathrm{mod}\ p_i^2) \leqslant 2 + (p_i - 4)p_i + 1 \cdot p_i^2 \geqslant m^2;$$
$$a_i^1 \cdot b_i^1(\mathrm{mod}\ p_i^2) \equiv c_i^{22} + c_i^3 \cdot p_i(\mathrm{mod}\ p_i^2) \leqslant$$
$$\leqslant 1 + (p_i - 2)p_i \leqslant m^2,$$

где $c_i^0, c_i^{01}, c_i^{10}, c_i^{11}, c_i^{12}, c_i^{22}, c_i^3$ — коэффициенты разложения слагаемых модулярных компонент произведения с учетом переполнений.

Для квадратичного модулярного диапазона отдельные из соотношений, имеющих, соответственно, множители 1, p, p^2, лишь косвенно отражаются на результате операции умножения. Переполнение, подлежащее обработке, появляется во втором соотношении рассматриваемой совокупности, причем коэффициент кратности переполнения единичный.

Заключение

Целесообразность применения для вычислений в больших компьютерных диапазонах в качестве внешней модулярной системы квадратичного диапазона обусловлена рядом причин, из которых основными являются:

— минимальные временные затраты на обработку прерываний при выполнении основных арифметических операций целочисленной арифметики при эмуляции в процессоре серийного компьютера с использованием знаковой целочисленной арифметики;

— возможность получения для многоразрядных вычислительных процессов соотношений, имеющих меньшую временную сложность для вычисления позиционных характеристик модулярного представления и некоторых важных для проблемных алгоритмов операций, например, вычисления вычета по большому модулю без операции деления в явном виде и ряда других.

Список литературы

1. **Амербаев В. М.** Теоретические основы машинной арифметики. Алма-Ата: Наука, 1976. 320 с.

2. **Инютин С. А.** Основы модулярной алгоритмики. Ханты-Мансийск: Полиграфист, 2009. 237 с.

3. **Ноден П., Китте К.** Алгебраическая алгоритмика. М.: Мир, 1999. 720 с.

4. **Инютин С. А.** Анализ сложности многоразрядных вычислительных процессов // Научные труды МАТИ. 2014. Вып. 22 (94). С. 154—159.

5. **Амербаев В. М., Стемпковский А. Л., Соловьев Р. А.** Принципы рекурсивных модулярных вычислений // Информационные технологии. 2013. № 2. С. 22—27.

6. **Шилов В. В., Столярский Е. З.** Организация и работа кэш-памяти // Информационные технологии. 2000. № 7. С. 2—8.

7. **Inyutin S. A.** Parallel Square Modular Computer Algebra // Transaction of Parallel Processing and Applied Mathematics PPAM-2003. Poland-Denmark: Springer, 2003. P. 117—123.

8. **Инютин С. А**. Метод вычисления количественной характеристики модулярной величины // Информационные технологии. 2016. Т. 22, № 5. С. 343—347.

9. **Инютин С. А.** Метод вычисления характеристики отношения порядка для параллельных форматов данных // Информационные технологии. 2017. Т. 23, № 8. С. 569—574.

S. A. Inyutin, PhD, Professor, e-mail: inyutin_int@mail.ru,
Moscow Aviation Institute (Nation Research University) (MAI)

The Aggregation of Number Systems for Multi-Digit Computational Processes

Designs and variants of combinations of external and internal number systems — mathematical bases of calculation methods in large computer ranges (multi-digit calculations) are introduced. Data formats for multi-digit calculations are systematized and substantiated. Studied design, containing internal and external positional power of the modular system. On the basis of code-independent Pepen and Lucas-Lehmer algorithms, code-dependent modular modifications of algorithms with less time complexity due to high memory costs are introduced. Linear spaces of modular vectors are investigated. The features and advantages of a modular system with a quadratic range and the corresponding parallel computer data formats for modular reconfigurable computing systems are revealed. The shown advantages justify the use of an external quadratic modular and internal positional number system within the machine range for the developed software tools for the organization of computational processes in large computer ranges on SIMD architecture computers.

Keywords: *multiprocessor reconfigurable systems SIMD architecture, multi-row computational process, code-dependent algorithms, linear and quadratic complexity of algorithms, parallel computer formats of quadratic modular system*

DOI: 10.17587/it.24.782-790

References

1. **Amerbaev V. M.** *Teoreticheskie osnovyi mashinnoy arifmetiki* (Theoretic base computer arithmetic), Alma-Ata: Nauka, 1976. 320 p. (in Russian).

2. **Inyutin S. A.** *Osnovyi modulyarnoy algoritmiki* (Base at modular algorithmic), Hantyi-Mansiysk, Poligrafist, 2009. 237 p. (in Russian).

3. **Noden P., Kitte K.** *Algebraicheskaya algoritmika* (Algebra algorithmic). Moscow, Mir, 1999, 720 p. (in Russian).

4. **Inyutin S. A.** Analiz slojnosti mnogorazryadnyih vyichislitelnyih protsessov (Analysis many digital calculation process), *Nauchnyie trudyi MATI*, 2014, vol. 22 (94), pp. 154—159 (in Russian).

5. **Amerbaev V. M., Stempkovsky A. L., Solovjev R. A.** Printsipyi rekursivnyih modulyarnyih vyichisleniy (Principles of recursive modular computing), *Informatcionnye tehnologii,* 2013, no. 2, pp. 22—27 (in Russian).

6. **Shilov V. V., Stolyarskiy E. Z.** Organizatsiya i rabota kesh-pamyati (Planning work Kech-memory), *Informatcionnye tehnologii*, 2000, no. 7, pp. 2—8 (in Russian).

7. **Inyutin S. A.** Parallel Square Modular Computer Algebra, *Transaction of Parallel Processing and Applied Mathematics PPAM-2003,* Poland-Denmark: Springer, 2003. pp. 117—123.

8. **Inyutin S. A.** Metod vyichisleniy kolichestvenoy harakteristic modulyarnoy velichina (Method calculation quantitative characteristic for computer modular value), *Informatcionnye tehnologii*, 2016, vol. 22, no. 5, pp. 343—347 (in Russian).

9. **Inyutin S. A.** Metod vyichisleniy harakteristic otnoshenij porjadka dliy parallel formatov dannih (Method calculation characteristic the order for parallel computer format date), *Informatcionnye tehnologii,* 2017, vol. 23, no. 8, pp. 569—574 (in Russian).

17—19 июня 2019 года в Москве состоится

XIII Всероссийское совещание по проблемам управления,

посвященное 80-летию Института проблем управления имени В. А. Трапезникова РАН

Направления работы Совещания:

- Теория систем управления
- Управление подвижными объектами и навигация
- Интеллектуальные системы в управлении
- Управление в промышленности и логистике
- Управление системами междисциплинарной природы
- Средства измерения, вычислений и контроля в управлении
- Системный анализ и принятие решений в задачах управления
- Информационные технологии в управлении
- Проблемы образования в области управления: современное содержание и технологии обучения

Подробную информацию о Совещании смотрите на сайте
http://vspu2019.ipu.ru