УДК 004.021

МЕТОД УСКОРЕННОГО УМНОЖЕНИЯ В СИСТЕМЕ ОСТАТОЧНЫХ КЛАССОВ С ИСПОЛЬЗОВАНИЕМ ЛОГАРИФМИЧЕСКИХ ИНТЕРВАЛЬНЫХ ХАРАКТЕРИСТИК

Коржавина А.С., Князьков В.С.

ФГБОУ ВО «Вятский государственный университет», Киров, e-mail: as_korzhavina@vyatsu.ru

Для решения многих задач вычислительной математики, математической физики, экономики, биохимии, криптографии требуется высокая, до 512–1024 бит и более, точность. Операция умножения – одна из наиболее часто выполняемых арифметических операций в таких расчетах. На сегодняшний день основным способом работы с длинными числами являются программные библиотеки позиционной длинной арифметики, главным недостатком которых является резкое снижение быстродействия вследствие возникающих единиц переноса между двоичными разрядами длинного позиционного числа. На практике даже самые быстрые способы умножения дают значительное снижение быстродействия на очень длинных числах, представленных в позиционных системах счисления, поэтому повышение скорости является основной целью при разработке методов умножения. В данной работе рассматривается метод умножения двух длинных чисел с плавающей точкой в гибридном модулярном интервально-логарифмическом формате представления. Мантисса представлена в системе остаточных классов (СОК), которая позволяет заменить последовательные методы выполнения арифметических операций над длинными целыми числами в позиционных системах счисления на параллельные методы выполнения арифметических операций над наборами коротких целых чисел, что существенно ускоряет выполнение операций сложения и умножения. В качестве метаданных формат содержит интервально-логарифмическую характеристику, необходимую для быстрого выполнения операций сравнения и масштабирования. Разработанный метод в среднем приблизительно в 3,4 раза быстрее метода, основанного на позиционной длинной арифметике.

Ключевые слова: система остаточных классов, умножение, логарифмическая система счисления, интервальная арифметика, формат с плавающей точкой

FAST MULTIPLICATION IN RESIDUE NUMBER SYSTEMS USING INTERVAL LOGARITHMIC CHARACTERISTIC

Korzhavina A.S., Knyazkov V.S.

Federal State Budgetary Educational Institution of Higher Education «Vyatka State University», Kirov, e-mail: as_korzhavina@vyatsu.ru

To solve many problems of computational mathematics, mathematical physics, economics, biochemistry, cryptography, a high precision up to 512-1024 bits or more, is required. The multiplication is one of the most frequently performed arithmetic operations in such computations. Nowadays one usually work with long numbers are software libraries of long positional arithmetic, the main drawback of which is a sharp decrease in performance due to the emerging transfer carries between binary digits of a long positional number. In practice, even the fastest methods of multiplication give a significant decrease in the performance over very long numbers represented in the positional number systems. Increasing the speed is the main goal in the multiplication methods. In this paper, the method of multiplying two long numbers with a floating point in a hybrid modular interval-log format of representation is introduced. The mantissa is represented by the residue number system (RNS), in which long integers is represented by the sets of independent digits. It allows replacing the successive methods of performing arithmetic operations over long integers in positional number systems on parallel methods of performing arithmetic operations on sets of short integers, which significantly speeds up the execution of the operations of addition and multiplication. The format contains as a meta-data the interval-logarithmic characteristic that required for fast comparison and scaling. The method is approximately 3.4 times faster than positional implementation.

Keywords: residue number system, multiplication, logarithmic number system, interval arithmetic, floating-point format

Проблема высокоточных вычислений стоит во многих областях вычислительной математики, математической физики, экономики, биохимии. Для решения таких задач, как прямое и обратное преобразования Лапласа [1], оптимальное управление [2], моделирование процессов метаболизма и конфигурации макромолекул [3], волновое рассеяние [4], требуется точность 256–2048 бит. Другой важной областью применения длинной арифметики являются задачи криптографии, в которых операция умножения длинных целых чисел является основной. При этом повышение скорости является основной целью при построении криптосистем, где длина чисел может достигать нескольких тысяч бит [5]. На сегодняшний день основным способом работы с длинными двоичными числами являются программные библиотеки позиционной длинной арифметики, главным недостатком которых является резкое снижение быстродействия вследствие возникающих единиц переноса между двоичными разрядами длинного позиционного числа. Однако даже самые быстрые способы умножения дают значительное снижение быстродействия на очень длинных числах [6]. Альтернативой

длинной позиционной арифметики является модулярная арифметика (системы счисления в остаточных классах, СОК).

СОК – это непозиционная система счисления, которая позволяет заменить последовательные методы выполнения арифметических операций над «длинными» целыми числами в позиционных системах счисления на параллельные методы выполнения арифметических операций над наборами «коротких» целых чисел [7]. Главным преимуществом СОК является высокое быстродействие таких операций, как сложение и умножение (так называемые модульные операции), поскольку нет необходимости фиксировать и распространять единицы переноса между разрядами; существенным недостатком – сложность выполнения немодульных операций, таких как операции масштабирования, сравнения и определения переполнения диапазона представления чисел. Учитывая особенности организации вычислений в СОК, их применение наиболее эффективно для решения задач, алгоритмизация которых реализована с существенным преобладанием модульных операций, особенно операций умножения.

Цель работы: повышение скорости выполнения операции умножения длинных чисел с плавающей точкой. Для повышения скорости вычислений предлагается использовать гибридное модулярное интервально-логарифмическое представление длинных чисел с плавающей точкой, в котором мантисса представлена в системе остаточных классов, а также содержащее в качестве метаданных интервально-логарифмическую характеристику, необходимую для быстрого выполнения операций сравнения и масштабирования.

Гибридный модулярный интервально-логарифмический формат данных

В интервальной арифметике вместо чисел, дискретных точек на числовой оси, используется пара чисел, представляющих собой границы закрытых интервалов, или отрезков числовой оси, что позволяет выполнять расчеты с учетом погрешностей в исходных данных и погрешностей, возникающих в процессе вычислений [8, 9]. При этом все вычисления выполняются таким образом, что результирующий интервал гарантированно содержит точный результат вычислений. В связи с этим интервальная арифметика позволяет повысить достоверность вычислений за счет учета в явном виде влияния ошибок округления [8]. Ширина интервала результата при выполнении арифметических операций в интервальной арифметике является не только критерием точности, но и критерием достоверности, поэтому уменьшение ширины интервала является одной из важных задач. В качестве одного из способов повышения точности интервальной арифметики могут быть применены логарифмические системы счисления (ЛСС) для представления границ интервала [10].

Число в интервально-логарифмическом формате представлено следующим образом:

в качестве одного из способов повышения точности интервала,

$$A \xrightarrow{\text{ИЛСС}} \left[\underline{L_A} = \underline{\log_b A}; \overline{L_A} = \overline{\log_b A}\right],$$

где log_b – логарифм числа по основанию b, вычисленный с округлением к минус и плюс бесконечности соответственно, A – модуль числа, представленный в позиционной системе счисления.

Результат умножения двух чисел, представленных в интервально-логарифмическом формате, определяется следующим образом:

$$\underline{L_Z} = \underline{L_A} + \underline{L_B}\ ;\ \overline{L_Z} = \overline{L_A} + \overline{L_B}\ .$$

Система остаточных классов является непозиционной системой счисления, в которой значения позиционного числа A представлены набором n остатков $\langle x_1, x_2, ..., x_n \rangle$ от деления числа A на каждый из n модулей $p_i \in \{p_1, p_2, ..., p_n\}$ [7] $A \xrightarrow{\text{СОК}} \langle x_1, x_2, ..., x_n \rangle$, где x_i вычисляется следующим образом:

$$x_i = |A|_{p_i} = A - \left\lfloor \frac{A}{p_i} \right\rfloor \cdot p_i, i = 1, 2, ..., n\ ,$$

где $\left\lfloor \frac{A}{p_i} \right\rfloor$ – целая часть частного $\frac{A}{p_i}$, $\{p_1, p_2, ..., p_n\}$ – набор оснований или базис СОК.

Произведение $P = p_1 \cdot p_2 \cdot ... \cdot p_n$ определяет диапазон представления чисел $A \in [0; P)$ в СОК, причем, если все числа $p_i \in \{p_1, p_2, ..., p_n\}$ являются попарно взаимно простыми, то между любым положительным целым числом A из диапазона [0; P) и числом, представленным в СОК, имеется взаимно однозначное соответствие.

Результат умножения двух чисел $\langle x_1, x_2, ..., x_n \rangle$ и $\langle y_1, y_2, ..., y_n \rangle$, представленных в СОК, определяется следующим образом:

$$\langle z_1, z_2, ..., z_n \rangle = \left\langle |x_1 \cdot y_1|_{p_1}, |x_2 \cdot y_2|_{p_2}, ..., |x_n \cdot y_n|_{p_n} \right\rangle .$$

В данной работе предлагается объединить преимущества обоих рассмотренных форматов представления, модулярного и интервально-логарифмического, в новый гибридный модулярный интервально-логарифмический (МИЛ) формат:

$$\left[\left\langle m_1, m_2, ..., m_n \right\rangle, L_l, L_h, E, \sigma \middle| m_i \in \{0,1,..., p_i - 1\}, L_l, L_h \in [0; L_P], \lambda \in [\lambda_{\min}, \lambda_{\max}], \sigma \in \{00, 01, 10, 11\}\right],$$

где σ – знак числа, $\sigma = \begin{cases} 01, a > 0 \\ 00, a = 0 \\ 11, a < 0 \end{cases}$, $\left\langle m_1, m_2, ..., m_n \right\rangle$ – мантисса числа, представленная в СОК, λ – масштаб (порядок) числа, L_l, L_h – интервально-логарифмическая характеристика (ИЛХ) мантиссы числа, $L = [L_l, L_h] = \left[\underline{\log_b |M|}, \overline{\log_b |M|}\right]$.

Метод выполнения операции умножения в модулярном интервально-логарифмическом формате

Пусть заданы два числа в МИЛ-формате

$$A = \left\langle M^A = \left\{m_i^A\right\}, \sigma^A, \lambda^A, L_{\min}^A, L_{\max}^A \right\rangle$$

и

$$B = \left\langle M^B = \left\{m_i^B\right\}, \sigma^B, \lambda^B, L_{\min}^B, L_{\max}^B \right\rangle.$$

Тогда результат

$$C = \left\langle M^C = \left\{m_i^C\right\}, \sigma^C, \lambda^C, L_{\min}^C, L_{\max}^C \right\rangle$$

будет вычислен следующим образом:

$$M^C = M^A \cdot M^B = \left\{\left| m_i^A, m_i^B \right|_{p_i}\right\},$$

$$\sigma^C = \sigma^A \cdot \sigma^B, \ \lambda^C = \lambda^A + \lambda^B.$$

Поскольку мантиссы чисел, представленных в МИЛ-формате, ограничены диапазоном [0; P – 1), то при выполнении умножения двух мантисс результат может выйти за пределы диапазона представления $M^C \geq P$. Для того, чтобы мантисса результата была представима в МИЛ-формате, необходимо выполнить операцию масштабирования $\frac{M^A \cdot M^B}{b^z}$, где $z = \left\lceil \log_b \frac{M^A \cdot M^B}{P} \right\rceil$.

Будем выполнять масштабирование обоих операндов до выполнения операции умножения таким образом, чтобы масштабированные сомножители не превышали величину $\sqrt{P}$, то есть

$$\tilde{M}^A = \frac{M^A - \left|M^A\right|_{b^{z_A}}}{b^{z_A}}, \ \tilde{M}^B = \frac{M^B - \left|M^B\right|_{b^{z_B}}}{b^{z_B}},$$

где z_A и z_B – коэффициенты, определяемые соотношениями значений модулей мантисс.

Таким образом, все необходимые вычисления (проверка переполнения, масштабирование) проводятся до непосредственного умножения мантисс.

На основании описанных выше выкладок сформулируем алгоритм умножения двух чисел, представленных в МИЛ-формате.

Шаг 1. Проверка операндов на ноль. Если хотя бы один из операндов равен нулю, то результат равен нулю: $\sigma^C = 0$, $\lambda^C = 0$, $M^C = \{0_i\}$, $L_{\min}^C = 0$, $L_{\max}^C = 0$, иначе перейти к следующему шагу.

Шаг 2. Проверка потери значащих разрядов: если $\lambda^A + \lambda^B + L_{\min}^A + L_{\min}^B \leq \min$, где $\min = \log_b(b^{\lambda_{\min}})$, то есть результат выполнения операции умножения двух чисел меньше, чем минимально представимое по модулю число, то результат равен нулю: $\sigma^C = 0$, $\lambda^C = 0$, $M^C = \{0_i\}$, $L_{\min}^C = 0$, $L_{\max}^C = 0$, иначе перейти к следующему шагу.

Шаг 3. Знак результата: $\sigma^C = \sigma^A \cdot \sigma^B$.

Шаг 4. Проверка переполнения: если $\lambda^A + \lambda^B + L_{\max}^A + L_{\max}^B \geq \max$, где $\max = \log_b(|m_1 \cdot m_2 \cdot ... \cdot m_n - 1| \cdot b^{\lambda_{\max}})$, то есть результат выполнения операции умножения двух чисел выйдет за границы представления чисел в формате IML, то присвоить результату значение «бесконечность»: $\lambda^C = \lambda_{\max}$, $M^C = \{0_i\}$, $L_{\min}^C = 0$, $L_{\max}^C = 0$, иначе перейти к следующему шагу.

Шаг 5. Проверка переполнения результата умножения модулярных мантисс. В случае, если результат умножения двух модулярных чисел выходит за границы представления, необходимо предварительно уменьшить один их сомножителей или оба сразу путем масштабирования степенью *b*. Если $L_{\max}^A + L_{\max}^B \geq L_P$, где $L_P = \log_b(|p_1 \cdot p_2 \cdot ... \cdot p_n - 1|)$, то вычислить значения масштабирующих коэффициентов: $L_1 = \left\lceil L_{\max}^A + L_{\max}^B - L_P \right\rceil$, $L_2 = \left\lceil L_{\max}^A - L_{\max}^B \right\rceil$, если $|L_1| \leq |L_2|$ и $L_2 \geq 0$, то $z_A \geq L_1$, $z_B = 0$; если $|L_1| \leq |L_2|$ и $L < 0$, то $z_A = 0$, $z_B = L_1$; если $|L_1| > |L_2|$ и $|L_1 + L_2|_2 = 1$, то $z_A = \frac{L_1 + L_2 + 1}{2}$, $z_B = \frac{L_1 - L_2 - 1}{2}$; если $|L_1| > |L_2|$ и $|L_1 + L_2|_2 = 0$, то $z_A = \frac{L_1 + L_2}{2} + 1$,

$z_B = \frac{L_1 - L_2}{2}$. Если $L^A_{\max} + L^B_{\max} < L_P$, то перейти к шагу 7.

Шаг 6. Вычислить значения скорректированных мантисс $\tilde{M}^A$ и $\tilde{M}^B$ операндов и скорректировать значение верхней и нижней границы интервальной логарифмической характеристики результата:

$$\tilde{M}^A = \frac{M^A - \left|M^A\right|_{b^{z_A}}}{b^{z_A}}, \quad \tilde{M}^B = \frac{M^B - \left|M^B\right|_{b^{z_B}}}{b^{z_B}}.$$

Скорректировать значения верхней и нижней границ ИЛХ чисел A и B:

$$l^A_{\min} = S\left(L^A_{\min}, \log_b \left|M^A\right|_{b^{z_A}}\right) - z_A;$$

$$l^A_{\max} = S\left(L^A_{\max}, \log_b \left|M^A\right|_{b^{z_A}}\right) - z_A;$$

$$l^B_{\min} = S\left(L^B_{\min}, \log_b \left|M^B\right|_{b^{z_B}}\right) - z_B;$$

$$l^B_{\max} = S\left(L^B_{\max}, \log_b \left|M^B\right|_{b^{z_B}}\right) - z_B,$$

где $S(x,y) = x + \log_b(1 - b^{y-x})$, если $y < x$, $S(x,y) = y + \log_b(1 - b^{x-y})$, если $y > x$.

Шаг 7. Выполнить модулярное умножение мантисс: $M^C = M^A \cdot M^B = \left\{\left|m^A_i, m^B_i\right|_{p_i}\right\}$ или $M^C = \tilde{M}^A \cdot \tilde{M}^B = \left\{\left|\tilde{m}^A_i, \tilde{m}^B_i\right|_{p_i}\right\}$, если мантиссы были скорректированы на шаге 6. Масштаб (порядок) результата: $\lambda^C = \lambda^A + \lambda^B + L_1$. ИЛХ результата: $L^C_{\min} = l^A_{\min} + l^B_{\min}$, $L^C_{\max} = l^A_{\max} + l^B_{\max}$.

Оценка быстродействия

Время выполнения предложенного метода (без учета исключений) равно

$$T_{IML} = p_{scl} \cdot t_{scl} + t_{MUL},$$

где p_{scl} – доля случаев, при которых требуется операция масштабирования – вероятность того, что произведение двух мантисс выйдет за пределы диапазона представления модулярных мантисс, t_{scl} – время выполнения операции масштабирования, t_{MUL} – время модулярного умножения.

Предположим, что числа из диапазона [0; P) могут появиться с равной вероятностью. Вероятность того, что произведение двух мантисс выйдет за пределы диапазона представления модулярных мантисс, то есть $M^A \cdot M^B \geq P$, равна

$$p_{scl} \approx \frac{P - \ln P}{P} \approx 1.$$

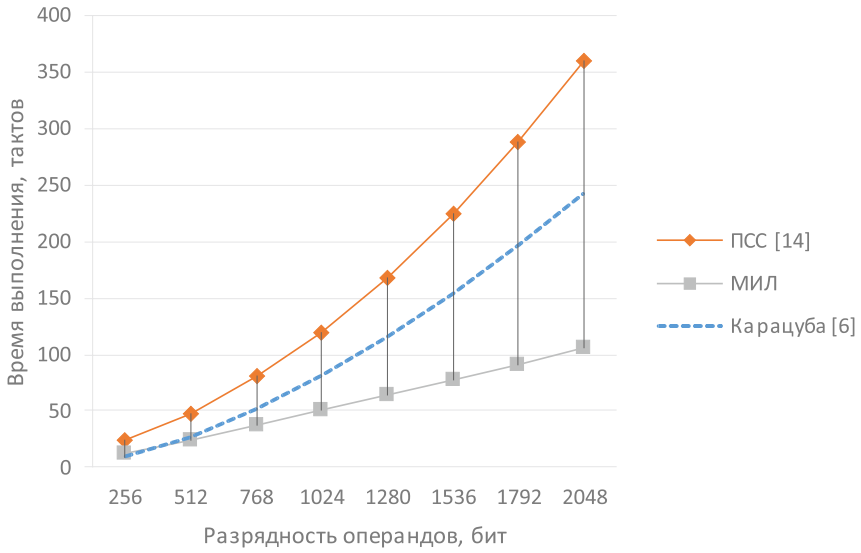
Время выполнения операции масштабирования равно [11]:

$$t_{scl} = \frac{n}{4}(t_{BEX} + t_{SUB} + t_{MUL}),$$

где t_{BEX} – время выполнения операции расширения базиса, t_{SUB} – время модулярного вычитания, t_{MUL} – время модулярного умножения.

Время расширения базиса при использовании самого быстрого алгоритма равно [11] $t_{BEX} = t_{MUL} + t_{ADD} \cdot (\log_2 n + 1)$.

Таким образом, общее время выполнения разработанного метода не превышает $T_{IML} = 2 + \frac{n}{4} \cdot (\log_2 n + 8)$ тактов.



Сравнение быстродействия разработанного метода и методов, основанных на позиционном представлении длинных чисел

Сравним разработанный метод и метод умножения длинных чисел с плавающей точкой [12], время выполнения которого составляет $\left(n^2/4+3\cdot n+8\right)$ тактов, а также с одним из асимптотически быстрых методов умножения, используемых для организации некоторых целочисленных двоичных умножителей (алгоритм Карацубы) [6]. Зависимость времени выполнения метода от разрядности операндов представлена на рисунке.

Разработанный метод в среднем приблизительно в 3,4 раза быстрее метода, основанного на позиционной длинной арифметике, и в 2,3 раза быстрее одного из асимптотически наиболее быстрых, но в явном виде практически не используемого алгоритма (алгоритм Карацубы) для разрядности мантисс 2048 бит.

Заключение

Разработан метод выполнения операции умножения с использованием модулярного интервально-логарифмического представления чисел с плавающей точкой. Отличительной особенностью представленного метода является использование интервальных логарифмических характеристик для быстрого выполнения немодульных операций сравнения и масштабирования. Разработанный метод в среднем приблизительно в 3,4 раза быстрее метода, основанного на позиционной длинной арифметике, и в 2,3 раза быстрее одного из асимптотически наиболее быстрых, но в явном виде практически не используемого алгоритма (алгоритм Карацубы) для разрядности мантисс 2048 бит.

Сложность разработанного метода выполнения умножения $O(n\log n)$, что быстрее, чем асимптотически быстрые алгоритмы умножения чисел в позиционной системе счисления: Шёнхаге – Штрассена, имеющего сложность $O(n\log n\log\log n)$, и Фюрера со сложностью $O(n\log n2^{O(\log^* n)})$.

Представленный метод защищен патентом РФ [13]. В качестве дальнейших направлений исследований предлагается разработка быстрых методов выполнения операций расширения базиса и масштабирования, что позволит уменьшить время выполнения операции умножения чисел в МИЛ-формате.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-37-00278 мол_а.

Список литературы

1. Krougly Z., Davison M., Aiyar S. The Role of High Precision Arithmetic in Calculating Numerical Laplace and Inverse Laplace Transforms. Applied Mathematics. 2017. vol. 8. no. 04. P. 562–589.

2. Pan B., Wang Y., Tian S. A High-Precision Single Shooting Method for Solving Hypersensitive Optimal Control Problems. Mathematical Problems in Engineering. 2018. vol. 2018. P. 1–11.

3. Yang L. et al. solveME: fast and reliable solution of nonlinear ME models. BMC bioinformatics, 2016. vol. 17. no. 1. P. 391.

4. Gergidis L.N. et al. Numerical investigation of the acoustic scattering problem from penetrable prolate spheroidal structures using the Vekua transformation and arbitrary precision arithmetic. Mathematical Methods in the Applied Sciences. 2018. P. 1–16.

5. Asif S., Kong Y. Highly parallel modular multiplier for elliptic curve cryptography in residue number system. Circuits, Systems, and Signal Processing. 2017. vol. 36. no. 3. P. 1027–1051.

6. Harvey D., Van Der Hoeven J., Lecerf G. Even faster integer multiplication. Journal of Complexity. 2016. vol. 36. P. 1–30.

7. Gérard B., Kammerer J.-G., Merkiche N. Contributions to the Design of Residue Number System Architectures. Proceedings of the 22nd IEEE International Symposium on Computer Arithmetic. France. 2015. P. 105–112.

8. Revol N. Introduction to the IEEE 1788-2015 Standard for Interval Arithmetic. International Workshop on Numerical Software Verification. Springer. Cham. 2017. P. 14–21.

9. Johansson F. Arb: Efficient arbitrary-precision midpoint-radius interval arithmetic. IEEE Transactions on Computers. 2017. vol. 66 (8). P. 1281–1292.

10. Коржавина А.С. Исследование эффективности реализации логарифмической интервальной арифметики на универсальных процессорах. Фундаментальные проблемы радиоэлектронного приборостроения. М.: Галлея-Принт. 2016. С. 165–168.

11. Коржавина А.С., Князьков В.С. Методы расширения базиса в системе остаточных классов: обзор и анализ вычислительной сложности // Современные наукоемкие технологии. 2017. № 12. С. 37–42.

12. Lei Y. et al. FPGA implementation of an exact dot product and its application in variable-precision floating-point arithmetic. The Journal of Supercomputing. 2013. vol. 64. no. 2. P. 580–605.

13. Князьков В.С., Коржавина А.С. Способ организации выполнения операции умножения двух чисел в модулярно-логарифмическом формате представления с плавающей точкой на гибридных многоядерных процессорах: 2666285 Рос Федерация: G06F7/32; заявл. 06.10.2017; опубл. 06.09.2018 бюл. № 25. 19 с.