УДК 681.325.5

ВЫЧИСЛЕНИЕ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ МЕТОДОМ "ЦИФРА ЗА ЦИФРОЙ"

Н.Н. ГЕРШУНИНА[1], А.А. БУЛАТНИКОВ[2]

[1]Кубанский государственный технологический университет,
350072, Российская Федерация, г. Краснодар, ул. Московская, 2
[2]ПАО "МТС"
350072, Российская Федерация, г. Краснодар, ул. Морская, 54 к.2

Предлагается и обосновывается целочисленный алгоритм вычисления трансцендентной функции ($y = e^x$). Производится расширение возможностей для случаев других функций (*sh x, ch x*).
Ключевые слова: быстродействующие целочисленные алгоритмы, микропроцессоры, алгоритмизация процессов локальной автоматики.

Стремление обеспечить простоту аппаратной реализации обработки данных от высокоскоростных источников информации (навигация, ракетные технологии, связь и т.п.) привело к созданию особых быстродействующих алгоритмов, в том числе называемых как "цифра за цифрой" (step by step – в англоязычной литературе) [1, 2, 3].

Они широко применялись в специализированных вычислителях, но исключительно как "hard wear", т.е. в структурах, дорогостоящих и реализуемых аппаратно.

В связи с появлением микропроцессоров (МП) интерес к таким алгоритмам возрос, но уже как к "soft wear", т.е. в программных комплексах [4].

Одним из таких направлений являются целочисленные (т.е. без умножения и деления) алгоритмы вычисления элементарных функций, наиболее удобные для высокоскоростных МП.

В данной статье предлагаются именно такие алгоритмы – типа "цифра за цифрой" – для вычисления e^x, shx, chx и других.

Суть предлагаемого усовершенствования заключается в преобразовании этапа псевдоделения алгоритма со знакопостоянными индикаторами (0;1) в эквивалентный этап псевдоделения со знакопеременными индикаторами (-1;1).

Этап псевдоумножения исходного алгоритма со знакопостоянными приращениями сохраняется без изменений.

Для определенности и обеспечения точности будем считать аргумент x функций – n-разрядным двоичным нормализованным числом - $x \in [0,2^{-1}]$.

1. Разработка алгоритма для функции $y = A \cdot e^x$

Этап знакопостоянного псевдоделения состоит в нахождении разложения заданного значения аргумента x в виде суммы

$$x = \sum_{i=1}^{n} q_i \cdot \ln(1 + 2^{-i}), \qquad (1)$$

где $q_i \in \{0,1\}$ – набор индикаторов для каждой i-й итерации, n – двоичная разрядность аргумента x.

На этапе знакопостоянного псевдоумножения по индикаторам q_i получается значение функции $y = A \cdot e^x$

$$y = A \cdot e^{\uparrow}(\sum_{\downarrow}(i = 1)^{\uparrow}n \blacksquare (q \blacksquare_{\downarrow} i \cdot \ln(1 + 2^{\uparrow}(-i)))) = A \cdot \prod_{\downarrow}(i = 1)^{\uparrow}n \blacksquare (1 + 2^{\uparrow}(-i)) \blacksquare^{\uparrow}(q_{\downarrow} i) \qquad (2)$$

Кроме известного [5] разложения (1) предлагаем этап знакопеременного псевдоделения, состоящий в нахождении для заданного аргумента x индикаторов из бинарного множества $s_i \in \{-1, +1\}$.

$$x = x_0 + \frac{1}{2} \cdot \sum_{i=1}^{n} s_i \cdot \ln(1 + 2^{-i}), \qquad (3)$$

где $x_0 = \frac{1}{2} \cdot \sum_{i=1}^{n} \ln(1 + 2^{-i})$ – число, не зависящее от x.

Теперь докажем, что по индикаторам s_i просто определить индикаторы q_i. Из (3) следует, что

(4)

После сравнения правой части (4) с (1) имеем

$$q_i = \frac{1 + s_i}{2} = \begin{cases} 1, & \text{если } s_i = 1, \\ 0, & \text{если } s_i = -1. \end{cases} \qquad (5)$$

Соединив этап знакопеременного псевдоделения, определенного выше (3), с этапом знакопостоянного умножения, заданного в (2), получаем следующий рекуррентный РИА

$$s_i = \begin{cases} 1, & \text{если } X_{i-1} \geq 0, \\ -1, & \text{если } X_{i-1} < 0. \end{cases}$$

$$Y_0 = A,\ Y_i = Y_{i-1} \cdot (1 + q_i \cdot 2^{-i}),\ Y_n \Rightarrow A \cdot e^x; \qquad (6)$$

$$X_0 = x - x_0,\ X_i = X_{i-1} - 0{,}5 \cdot s_i \cdot \ln(1 + 2^{-i}) \Rightarrow x_i$$

где $i=1, 2, \ldots, n$ – номер итерации, q_i определяется из (5), а $0{,}5 \cdot \ln(1 + 2^{-i})$ – константы (их n штук).

Обращаем внимание на то, что РИА (5) не содержит умножений и делений в составе своих рекуррентных соотношений. А присутствующие в них 2^{-i} умножения на заменяются сдвигами итерируемой величины Y_{i-1} на i разрядов в сторону младших разрядов.

2. Вычисление shx и chx

В основе алгоритма лежат формулы

$$Shx = \frac{e^x - e^{-x}}{2}, \qquad Chx = \frac{e^x + e^{-x}}{2} \qquad (7)$$

Вычисление по ним ведется параллельно во времени. Единственное отличие вычисления $y = e^{-x}$ от вышеприведенного РИА (6) состоит в наборе констант $\ln(1 - 2^{-i})$ и наличие минуса у $(1 - q_i \cdot 2^{-i})$ во втором соотношении РИА (6).

В окончательном виде РИА для вычисления $y = A \cdot e^x$ выглядит так

$$s_i = \begin{cases} 1, \text{если } X \geq 0, \\ -1, \text{если } X < 0. \end{cases}$$

$$Y_0 = A,\ Y_i = Y_{i-1} \cdot (1 - q_i \cdot 2^{-i}),\ Y_n \Rightarrow A \cdot e^{-x}; \qquad (8)$$

$$X_0 = x + x_0,\ X_i = X_{i-1} + 0{,}5 \cdot s_i \cdot \ln(1 - 2^{-i}),\ X_n \Rightarrow x;$$

где i, n, q_i – то же самое, что и в (6); $\ln(1 - z^{-i})$- константы (их *n* штук), x_0 – константа, не зависящая от *x*

$$x_0 = 0{,}5 \cdot \sum_{i=1}^{n} \ln\left(1 - 2^{-i}\right) \qquad (9)$$

3. Анализ быстродействия РИА

Преимущество предложенного РИА (6) с его этапом знакопеременного псевдоделения по сравнению с известным [5], где используется знакопостоянное псевдоделение состоит в двойном сокращении операций.

Рассмотрим это подробнее. Как и при обычном делении делитель отнимается от делимого. И если у остатка знак “минус”, то делитель снова прибавляется к делимому. Это называется “восстановлением остатка”. В этом случае индикатор q_i равен нулю. Процесс деления на данной итерации заканчивается.

В случае знакопеременного деления (псевдоделения) восстановление остатка не производится, т.е. выработка индикатора s_i сходит быстрее.

Сам же термин “псевдо” появляется из-за того, что делитель состоит не из $-i$

весомозначных числовых кодов , а из констант $\ln(1 \pm z^{-i})$.

ЛИТЕРАТУРА

1.Байков В.Д., Смолов В.Б. Специализированные процессы: итерационные алгоритмы и структуры./ М.: Радио и связь, 1985.-288с.

2.Оранский А.М. Аппаратные методы в цифровой вычислительной технике.-Минск: Изд-во БГУ, 1977.-208с.

3. Linhardt R.J., Miller H.S. Digit by digit transcendental function computation.- RCA Review, 1969, vol.30, №2, p.209-247.

4.Байков В.Д., Вашкевич С.Н. Решение траекторных задач в микропроцессорных системах ЧПУ.-Л.:Машиностроение, 1986.-106с.

5.Булатникова И.Н. Информационные технологии с использованием целочисленной арифметики//Аналитический н.-т. журнал “Геоинжиниринг”, НИПИ “ИнжГео”.-2011.- №2(11). – с.54-58.

REFERENCES

1. Bajkov V.D., Smolov V.D. Specializirovannye process: iteracionnye algoritmy I struktury. (Specialized processes: iterative algorithms and patterns) /M.:Radio i svyaz. 1985. -288 s.

2. Oranskij A.M. Apparatnye metody v cifrovoj vychislitelnoj tekhnike.(Instrumental methods in digital computing)/Minsk.izd-vo BGU.1977.208s

3. Linhardt R.J., Miller H.S. Digit by digit transcendental function computation/RCA Review, 1969, vol.30, №2, p.209-247.

4.Bajkov V.D. Vashkevich S. N. Reshenie traektornyh zadach v mikroprocessornyh sistemah ChPU (The solution of trajectory problems in microprocessor-based CNC systems)/L.:Mashinostroenie. 1986.-106s.

5.Bulatnikova I.N./ Analiticheskij n.- t. zhurnal “Geoinzhiniring” NIPI “Inzhgeo (Geoengineering)”-2011.-№2(11).-s. 54-58

THE COMPUTATION OF ELEMENTARY FUNCTIONS

USING THE “DIGIT AFTER”

N.N. GERSHUNINA[1], A.A. BULATNIKOV[2]

[1]Kuban State Technological University,
2, Moskovskaya st., Krasnodar, Russian Federation, 350072
[2]JSC "MTS"
54/2, Morskay st., Krasnodar, Russian Federation, 350072

Proposed and validated an integer algorithm for computing transcendental functions (y=e^x). Is empowering for cases other functions (*sh x, ch x*).

Key words: fast integer algorithms, microprocessors, algorithmic processes of local automation.