

СЕРГИЕНКО А.М.

ПРИМЕНЕНИЕ РАЦИОНАЛЬНЫХ ДРОБЕЙ В СПЕЦИАЛИЗИРОВАННЫХ ВЫЧИСЛИТЕЛЯХ

В статье рассмотрено нетрадиционное представление данных в виде рациональных дробей, которое позволяет, не применяя чисел с плавающей запятой, выполнять вычисления с повышенной точностью.

An untraditional rational factor data representation in the application specific processors is considered. This data representation provides high computational precision and helps to do without floating point numbers.

Введение

Традиционно, вычисления с повышенной точностью выполняются с плавающей запятой. Однако, например, в цифровой обработке сигналов наибольшее распространение получили сигнальные микропроцессоры с фиксированной запятой, в которых плавающая запятая реализуется неэффективно. Также большие вычислительные возможности предоставляют современные программируемые логические интегральные схемы (ПЛИС), архитектура которых ориентирована на эффективную реализацию устройств с фиксированной запятой. При этом обеспечивается высокая точность вычислений, но за счет существенного увеличения разрядности данных и уменьшения быстродействия. Поэтому актуален поиск способов реализации вычислений с повышенной точностью, основанный на представлении целых чисел.

Для минимизации погрешностей, задержек и аппаратных затрат при реализации вычислений в специализированных устройствах предлагается использовать нетрадиционное представление данных в виде дробей. Такое представление приведено в [1] для целочисленного решения задач с полиномами.

Рациональные дроби

Рациональная дробь – это числовой объект, состоящий из целочисленных числителя и знаменателя. Судя по ее названию, эта дробь представляет рациональное число, т.е. число, полученное как решение системы целочисленных линейных уравнений или как частное от деления целочисленных полиномов. Рациональное число n_x/d_x может эффективно аппроксимировать заданное иррациональное или трансцендентное число x . Если нецелое число x представлено $2n$ разрядами с погрешностью ξ_1 , то оно может быть представлено дробью $x = n_x/d_x$ с погрешностью $\xi_2 \approx \xi_1$, причем

числа n_x и d_x будут иметь не более чем n разрядов в своем представлении [2].

Представление чисел рациональными дробями имеет ряд преимуществ. Во-первых, используемые в ЭВМ двоичные дроби приближенно представляют вещественные числа. Так, например, дробь $1/9 = 1/1001_2$ является точным рациональным числом в любой системе исчисления и представляется в виде десятичной дроби 0.1111 или двоичной дроби 0.11100011100011₂ с погрешностью, равной отбрасываемым разрядам бесконечной периодической десятичной или двоичной дроби.

Во-вторых, рациональные дроби упрощают нахождение аппроксимации с заданной точностью иррациональных или трансцендентных чисел. Многие элементарные функции эффективно вычисляются по соответствующим формулам рациональной аппроксимации [3].

И в-третьих, рациональные дроби предоставляют простой аппарат арифметических действий. При этом умножение, деление и сложение выглядят так:

$$\begin{aligned}x \cdot y &= n_x n_y / (d_x d_y); & x/y &= n_x d_y / (d_x n_y); & (1) \\x+y &= (n_x d_y + n_y d_x) / (d_x d_y).\end{aligned}$$

При этом собственно деление не выполняется. Для сравнения чисел достаточно вычислить

$$n_x d_y - n_y d_x.$$

При сопоставлении сложности операций следует учесть, что разрядность числителя и знаменателя, по крайней мере, вдвое меньше, чем у целых чисел при той же точности представления. Поэтому сложность аппаратных сумматоров дробей приближается к сложности аппаратных умножителей целых чисел, а сложность умножителей дробей оказывается вдвое меньшей, чем для целых чисел.

Недостаток вычислений с рациональными дробями состоит в быстром росте разрядности числителя и знаменателя при выполнении серии точных

последовательных действий. Поэтому для компенсации этого явления приходится периодически искать в дробях наибольшее общее кратное и делить на него числитель и знаменатель [4].

При вычислениях с рациональными дробями, если не требуются точные результаты, следует использовать округление и нормализацию числителя и знаменателя. При этом точность вычислений приближается к точности вычислений с плавающей запятой при малых значениях ее экспоненты, что будет показано ниже. Но сложность вычислений существенно меньше, так как не нужно выполнение операций с порядками и выравнивание порядков при сложении [5].

Однако, при использовании рациональных дробей остается открытым вопрос преобразования данных в традиционном представлении в такие дроби и обратно. Преобразование целого числа в дробь – тривиально – число подставляется в числитель, а знаменатель принимается равным 1. Обратный процесс затруднен тем, что для него требуется выполнение полноразрядного деления с получением результата двойной разрядности. Следующие доводы смягчают это обстоятельство.

Вычисления с повышенной точностью – это, как правило, сложные, многоитерационные вычисления. Поэтому все вычисления на первых $N-1$ итерациях, например, решения системы уравнений, предлагается выполнять с данными в виде рациональных дробей. И только в конце последней итерации алгоритма решения задачи числители дробей делятся на знаменатели обычным делением с получением результатов алгоритма.

В результате, представление данных в виде рациональных дробей обеспечивает как малые погрешности вычислений, так и расширенный динамический диапазон в сравнении с арифметикой чисел с фиксированной запятой, а также простоту реализации в ПЛИС и высокое быстродействие в сравнении с устройствами с плавающей запятой. При большой разрядности дробей (больше 32) точность вычислений может превышать точность вычислений с плавающей запятой, что будет показано ниже.

Следует также отметить, что представление данных рациональными дробями естественно для задач линейной алгебры (ЛА). Промежуточные и выходные результаты решения таких задач точно равны некоторым рациональным дробям. Таким образом, если исходные данные представляют обусловленную задачу ЛА, то при достаточной разрядности дробей получаются точные результаты.

Процессор для разложения Холецкого с арифметикой рациональных дробей

Разложение Холецкого обычно применяется для LL^T - разложения матрицы A таким образом, что

$$A = L \cdot L^T,$$

где $L_{(N,N)}$ – нижняя треугольная матрица. Этот алгоритм может быть представлен следующей программой

```

do i = 1, N
  l(i, i) = SQRT(a(i, i));
  do j = i+1, N
    l(j, i) = a(j, i)/l(i, i);
    do k = i+1, j
      a(j, k) = a(j, k) - a(j, i)*a(k, i);
    end do
  end do
end do.

```

Преимущество алгоритма Холецкого в том, что у него в 2 раза меньше объем вычислений, чем у алгоритма Гаусса. Алгоритм стабилен и не требует выбора ведущего элемента, благодаря чему он подходит для реализации в специализированных вычислительных системах. Поэтому спецпроцессор для выполнения этого алгоритма, в котором реализована арифметика дробей является ярким примером такой реализации.

Критический участок алгоритма – это циклическое вычисление умножения с вычитанием. Также повторяющейся операцией является деление. Эти операции эффективно выполнять над рациональными дробями. Наконец, необходимо N раз вычислить квадратный корень. Для реализации этой функции традиционно используется метод Ньютона. Он основан на итерационном уточнении аппроксимированного значения t^1 по формуле [3]:

$$t^{k+1} = (t^k + p/t^k)/2, \quad (2)$$

где $k = 1, \dots, K$ – номер итерации. Исходное значение t^1 можно принять равным 1. Моделирование вычислений рациональных дробей показало, что при таком t^1 итерационный процесс (2) сходится с погрешностью представления данного за $K = 5$ итерации для разрядности дроби менее 64.

Для уменьшения числа итераций до $K = 3$ предлагается генерировать начальное приближение как $t^1 = t_n/t_d$, где t_n, t_d равны константе $a=1$ или 3, умноженной на $2^{-\lfloor n/2 \rfloor}$ или $2^{-\lfloor d/2 \rfloor}$ (т.е. это a , сдвинутое на $\lfloor n/2 \rfloor$ или $\lfloor d/2 \rfloor$ разрядов вправо), где n и d – число нулей перед старшей значащей единицей в числителе и знаменателе числа p , причем $a=3$, если n или d – нечетное.

Арифметическое устройство (АУ) спецпроцессора имеет структуру, показанную на рис.1. В ней выполняется базовая операция $P=AX \pm Y$, в которой сложение и умножение выполняются согласно (1). Для выполнения деления A/X операнд $Y=0$, а числитель и знаменатель X меняются местами. Блоки нормализации БН1, БН2 выполняют сдвиг влево числителя и знаменателя на одинаковое число разрядов, которое не превосходит половины их разрядности.

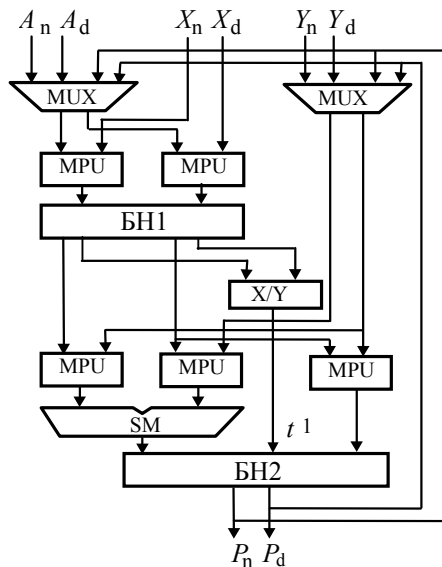


Рис.1 Структура АУ для вычисления алгоритма Холецкого

Обратные связи в структуре АУ предназначены для реализации аккумулятора парных произведений и вычислений квадратного корня по уравнению (2). При этом числа разрядов n и d для получения приближения t^1 в схеме X/Y поступают из блока нормализации БН1. В результате, АУ выполняет вычисления в конвейерном режиме с глубиной конвейера 9, причем вычисления с накоплением $P = P \pm AX$ выполняются с периодом 4 такта, а время вычисления квадратного корня составляет 36 тактов.

Характеристики полученного АУ при его реализации на ПЛИС разных серий представлены в табл. В ней аппаратная сложность выражена в количестве эквивалентных конфигурированных логических блоков (ЭКЛБ) и блоков 18-разрядных умножителей DSP48. Для сравнения там же показаны характеристики АУ, составленного из блоков операций с плавающей запятой одинарной точности (разрядность мантиссы – 23), которые обеспечивают такую же функциональность и доступны в САПР Xilinx Coregen [6].

Таблица. Характеристики АУ для реализации алгоритма Холецкого.

Тип ПЛИС и АУ	XC 4VSX25-12		XC 5VLX50-3
	Xilinx Coregen	Предлагаемое	Предлагаемое
Апп. затраты, ЭКЛБ	1350 (13%),	1089 (10%),	729 (15%)
Апп. затраты, DSP48	8 (6%),	15 (11%),	15(46%)
Максимальная тактовая частота, МГц	165	195	225

Как видим, предлагаемое АУ имеет на 30% меньшие затраты в числе ЭКЛБ. Хотя количество умножителей увеличено почти вдвое, но аппаратные затраты оказываются сбалансированными, т.е. при полной загрузке кристалла число задействованных как умножителей, так и ЭКЛБ будет минимальным.

Анализ погрешностей алгоритма Холецкого

Для доказательства эффективности арифметики рациональных дробей при реализации разложения Холецкого был выполнен ряд вычислений с различной разрядностью дробей и с плавающей запятой одинарной точности. На рис. 2 показаны зависимости погрешностей, помноженных на 10^7 , от разрядности операндов и размерности задачи N .

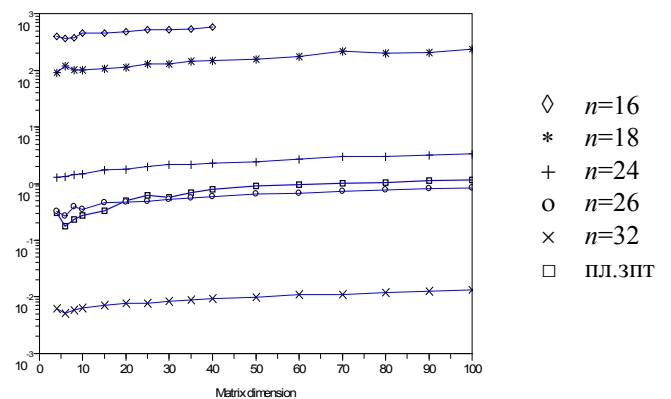


Рис. 2. Погрешности разложения Холецкого

Анализ графиков показывает, что арифметика рациональных дробей с разрядностью 26 обеспечивает такую же погрешность вычислений, как и арифметика плавающей запятой одинарной точности; увеличение разрядности на единицу уменьшает погрешность вдвое.

Эксперименты с разложением Холецкого с различными данными показали, что при изменении масштаба входной матрицы (умножении ее на коэффициент M) существенно меняется ошибка вычислений. Графики зависимости ошибки вычислений от масштаба M показаны на рис. 3.

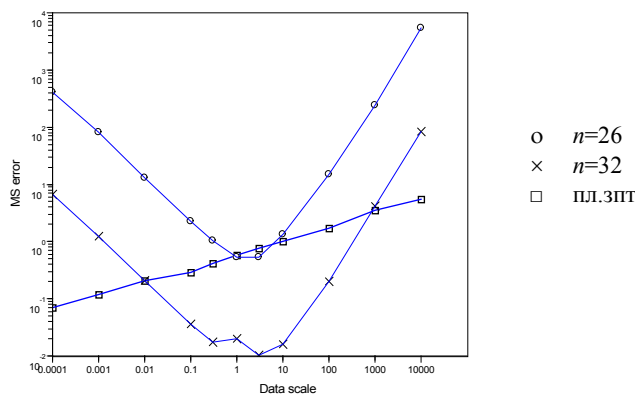


Рис. 3. Зависимость погрешности от масштаба входных данных

Как видим, минимальная погрешность вычислений достигается при $M=1$, т.е. при матожидании данных, равном 1. Это объясняется тем, что при отклонении M от единицы или в числителе, или в знаменателе дробей число значащих цифр уменьшается на $\log_2 M$ разрядов. Критическим становится масштаб $M=2^{n/2}$, когда при выполнении операций может получиться нулевой или максимальный результат.

Также было установлено, что рост ошибки при вычислениях в арифметике дробей – приблизительно в $2,3 \approx \sqrt[3]{10}$ раза при увеличении размерности задачи в 10 раз. При этом при вычислении с

плавающей запятой с одинарной точностью это отношение несколько больше – 4,3 [7].

Выводы

Таким образом, показана высокая эффективность реализации метода Холецкого с помощью арифметики рациональных дробей. Погрешность разложения Холецкого при применении n -разрядных дробей оценивается как $2^{-n} MN^{1/3}$. Кроме того, если размерность матрицы не превосходит $N=100$, то разложение Холецкого дает среднеквадратическую ошибку около $2 \cdot 10^{-5}$ при вычислении с 18-разрядными дробями. Такой точности достаточно для решения многих задач цифровой обработки сигналов, например в адаптивных фильтрах. При этом эффективно используются ресурсы современных ПЛИС.

Высокая эффективность применения рациональных дробей также была проверена при решении систем уравнений с треугольной матрицей [8], методом сопряженных градиентов [5], выполнении дискретного преобразования Фурье [9].

Представление и обработка данных в виде рациональных дробей может быть также получена эффективную программную реализацию, например, в сигнальных микропроцессорах.

Список литературы

1. Кнут Д. Искусство программирования. Т.2. –М.:Мир. –1979. –556с.
2. Хинчин А.Ю. Цепные дроби. – М.: Наука, 3-е изд., –1978. –112с.
3. Попов Б.А., Теслер Г.С. Вычисление функций на ЭВМ. –Киев: Наукова думка. –1984. –599с.
4. Irvin M.J., Smith D.R. A rational arithmetic processor // Proc. 5-th Symp. Comput. Arithmetic. –1981. –P.241-244.
5. Сергиенко А.М. Применение арифметики рациональных дробей для реализации метода сопряжения градиентов. //Электрон. моделирование. –2006. –Т.28. –№ 1. – С. 33–41.
6. Sergiyenko A., Maslennikov O. Ratushniak P. Implementation of Linear Algebra Algorithms in FPGA-based Fractional Arithmetic Units. //Proc. 9-th Int. Conf. “The Experience of Designing and Application of CAD Systems in Microelectronics”, CADSM’2007. –Lviv-Polyana –20–24 Feb. –2007. – p.228–234.
7. Sergiyenko A., Maslennikov O., Lepekha V., Tomas A., Wyrzykowski R. Parallel Implementation of Cholesky LL^T Algorithm in FPGA-Based Processor //Lecture Notes in Computer Science. —Berlin: Springer. —2008. –V. —p. 137 —147.
8. Maslennikov O., Shevtshenko Ju., Sergiyenko A. Configurable Microprocessor Array for DSP Applications.// Lecture Notes in Computer Science. –V.3019. –2004. –P.36–41.
9. Сергиенко А.М., Мелковська В.М. Спосіб демодуляції сигналів з багаточастотною модуляцією // Вісник Національного технічного університету України. –Сер. Інформатика, управління та обчислювальна техніка. –2008, № 48. — с. 82–84.