УДК 681.322

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, большинство микроконтроллеров, а также программируемые логические интегральные схемы (ПЛИС) вычисляют, как правило, с фиксированной запятой, так как в них плавающая запятая реализуется неэффективно. Поэтому актуален поиск способов реализации вычислений с повышенной точностью, основанный на представлении целых чисел.

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

Рациональные дроби

Рациональная дробь – это числовой объект, состоящий из числителя и знаменателя. 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 $1/9 = 1/1001_2$ – это точное рациональное число в любой системе счисления, но представляется десятичной дробью 0.1111 или двоичной дробью 0.111000111000011_2 с погрешностью, равной отбрасываемым разрядам бесконечной периодической дроби.

Во-вторых, рациональные дроби упрощают аппроксимацию иррациональных или трансцендентных чисел. Элементарные функции эффективно вычисляются по формулам рациональной аппроксимации [3].

И в-третьих, рациональные дроби имеют простой аппарат арифметических действий. При этом основные операции выглядят так:

$$x \cdot y = n_x\, n_y / (d_x\, d_y)\,; \qquad x/y = n_x\, d_y / (d_x\, n_y)\,; \quad (1)$$
$$x+y = (n_x\, d_y + n_y\, d_x)/(d_x\, d_y).$$

При этом собственно деление не выполняется. Для сравнения чисел вычисляется

$$n_x\, d_y - n_y\, d_x\,.$$

При сопоставлении сложности операций нужно 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 росте разрядности числителя и знаменателя при выполнении серии точных последовательных действий. Для компенсации этого явления приходится 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 плавающей запятой при малых значениях ее экспоненты, что будет показано ниже. Но сложность вычислений при этом существенно меньше, так как не нужно выполнение операций с порядками и выравнивание порядков при сложении [5].

Однако остается открытым вопрос преобразования данных в традиционном представлении в рациональные дроби и обратно. Преобразование целого в дробь – тривиально, число подставляется в числитель, а в знаменатель подставляется общий масштабный множитель. Обратный процесс затруднен тем, что для него требуется полноразрядное деление с результатом двойной разрядности.

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 числители дробей делятся на знаменатели обычным делением с получением результатов.

Поэтому представление данных рациональными дробями 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 (более 32) точность вычислений может превышать точность плавающей запятой.

Следует также отметить, что представление данных рациональными дробями естественно для задач линейной алгебры (ЛА). Если исходные данные представляют обусловленную задачу ЛА, то при достаточной разрядности дробей получаются точные результаты. Эффективность рациональных дробей покажем на примере разработки спецпроцессора для разложения Холецкого.

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 $L_{(\mathrm{N,N})}$ – нижняя треугольная матрица. Этот алгоритм может быть представлен так:

```
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.
```

У этого алгоритма в 2 раза меньше объем вычислений, чем у известного алгоритма Гаусса. Алгоритм стабилен и не требует выбора ведущего элемента, благодаря чему он удачно подходит для реализации в специализированных вычислительных системах.

Критический участок алгоритма – это циклическое вычисление умножения с вычитанием. Также повторяющейся операцией является деление. Наконец, необходимо N раз вычислить квадратный корень. Эта функция в арифметике дробей эффективно вычисляется по формуле Ньютона[3]:

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

где $k = 1,\ldots,K$. Моделирование вычислений рациональных дробей показало, что при начальном приближении $t^1 = 1$ итерационный процесс (2) сходится с погрешностью представления исходного данного p за $K = 5$ итерации для разрядности дроби менее 64.

Для уменьшения числа итераций до $K = 3$ предлагается $t^1 = t_n / t_d$, где t_n, t_d, равны константе a=1 или 3, умноженной на $2^{-]n/2[}$ или $2^{-]d/2[}$ (т.е. это a, сдвинутое на $]n/2[$ или $]d/2[$ 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) спецпроцесора имеет структуру, показанную на рис.1. В ней выполняется базовая операция $P=AX\pm Y$, в которой действия выполняются согласно (1). Для деления A/X операнд Y=0, числитель и знаменатель X меняются местами. Блоки нормализации БН1, БН2 сдвигают влево числитель и знаменатель на одинаковое число разрядов, которое не превосходит половины их разрядности.

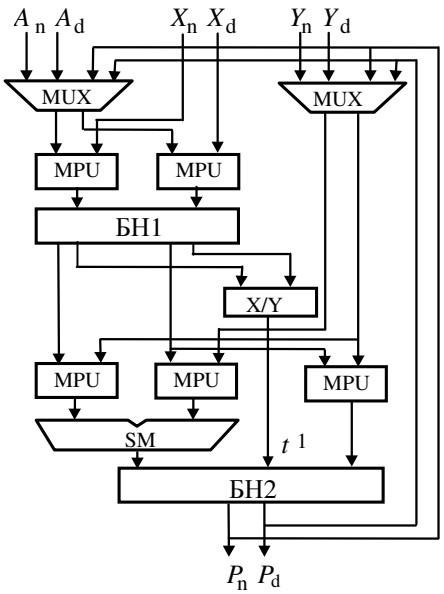

Рис.1 Структура АУ для вычисления алгоритма Холецкого

Обратные связи в структуре АУ предназначены для реализации аккумулятора парных произведений и вычислений квадратного корня по (2). При этом коды разрядов n и d для получения приближения t^1 в схеме X/Y поступают из блока нормализации БН1. АУ выполняет вычисления в конвейерном режиме с глубиной конвейера 9, причем вычисления с накоплением $P=P\pm AX$ имеют период 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 аппартная 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	1450 (14%),	1089 (10%),	729 (15%)
Апп. затраты, DSP48	8 (6%),	15 (11%),	15(46%)
Макс. тактовая частота	165 МГц	195 МГц	225 МГц

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

Погрешности алгоритма Холецкого

Был выполнен ряд вычислений разложения Холецкого с различной разрядностью дробей и для сравнения – с плавающей запятой одинарной точности. На рис. 2 показаны зависимости погрешностей, помноженных на 10^7, от разрядности операндов и размерности задачи.

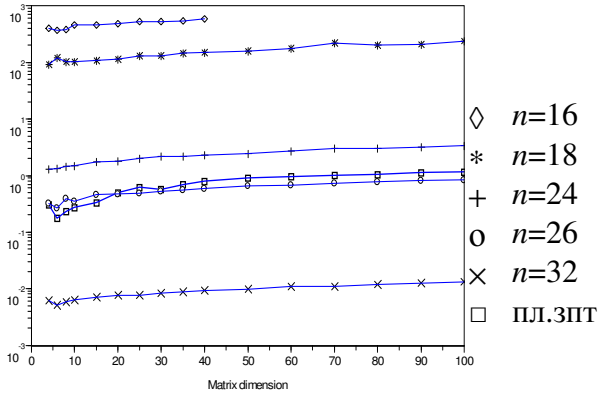

Рис. 2. Погрешности разложения Холецкого

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 показали, что при изменении масштаба входной матрицы (умножении ее на коэффициент M) существенно меняется ошибка выч-лений. Графики зависимости ошибки вычислений от масштаба M показаны на рис. 3.

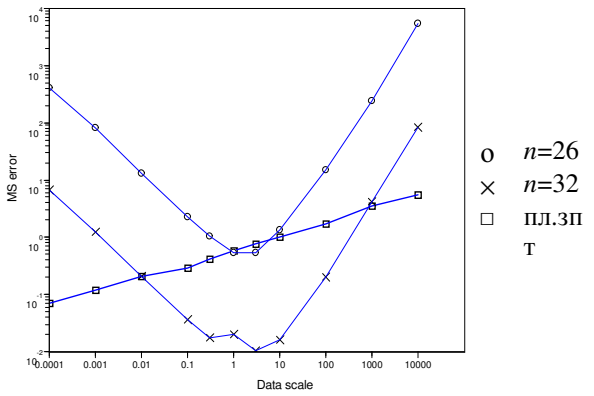

Рис. 3. Зависимость погрешности от масштаба входных данных

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 $\log_2 M$. Критическим становится масштаб $M=2^{n/2}$, когда при выполнении операций может получиться нулевой или максимальный результат.

Также замечено, что при увеличении размерности задачи в 10 раз ошибки вычислений растут в $2{,}3 \approx \sqrt[3]{10}$ раза. При вычислении с плавающей запятой это отношение несколько больше – 4,3 [7].

Выводы

Таким образом, показана высокая эффективность арифметики рациональных дробей на примере спецпроцессора для разложения Холецкого. 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 $2 \cdot 10^{-5} \cdot M$ 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.

Эффективность применения рациональных дробей также была проверена при решении систем уравнений с теплицевой матрицей [8], реализации метода сопряженных градиентов [5].

Представление и обработка данных в виде рациональных дробей может также получить 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. Sergyienko 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., Maslennikow 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., Sergyienko A. Configurable Microprocessor Array for DSP Applications.// Lecture Notes in Computer Science. –V.3019. –2004. –P.36–41.

Поступила в редакцию 27.11.2009