

КЛАСС АППАРАТУРНО-ОРИЕНТИРОВАННЫХ АЛГОРИТМОВ ДИСКРЕТНЫХ ЛИНЕЙНЫХ ПРЕОБРАЗОВАНИЙ

Егунов В.А.

ФГБОУ ВПО Волгоградский государственный технический университет, Волгоград, Россия (400131, г. Волгоград, пр. Ленина, 28), e-mail: vegunov@mail.ru

Приведен базовый алгоритм класса аппаратурно-ориентированных алгоритмов дискретных линейных преобразований (ДЛП). Рассмотрены модификации базового алгоритма, предназначенные для программно-аппаратной реализации на средствах вычислительной техники, использующие различные способы повышения скорости сходимости базового алгоритма. Данные модификации базового алгоритма требуют, с одной стороны, различного числа итераций для достижения необходимой точности вычислений, с другой стороны, для их реализации необходимы различные объемы дополнительных вычислений, что в итоге приводит к росту операционной сложности отдельной итерации, зависящему от конкретной модификации базового алгоритма. В результате предложен обобщенный класс алгоритмов дискретных линейных преобразований. Полученный обобщенный ДЛП-класс может быть использован для построения алгоритмов, предназначенных для программно-аппаратной реализации и использующих различные вычислительные схемы для реализации отдельных итераций – отражение, вращение и т.д.

Ключевые слова: аппаратурно-ориентированные алгоритмы, дискретные линейные преобразования, CORDIC.

CLASS HARDWARE-ORIENTED ALGORITHMS OF DISCRETE LINEAR TRANSFORMATIONS

Egunov V.A.

Volgograd state technical university, Volgograd, , Russia (400131, Volgograd, avenue of Lenin, 28), e-mail: vegunov@mail.ru

Is the basic algorithm class hardware-oriented algorithms of discrete linear transformations (DLP). Considered modifications of the basic algorithm intended for the hardware and software for the computers that use different ways to increase the speed of convergence of the basic algorithm. These modifications of the basic algorithm requires on the one hand a different number of iterations to achieve the necessary accuracy of the calculations, on the other hand, for their implementation need different amounts of additional computations, which ultimately leads to increased operational complexity separate iteration, depending on the specific modification of the basic algorithm. As a result of the proposed generalized class of algorithms of discrete linear transformations. The obtained generalized DLP-class can be used to construct algorithms intended for the hardware and software implementation and using different computational scheme for the implementation of individual iterations - reflection, rotation, etc.

Keywords: hardware-oriented algorithms, discrete linear transformations, CORDIC.

Одним из подходов к созданию аппаратурно-ориентированных алгоритмов является синтез дискретных линейных преобразований (ДЛП)[4]. Под ДЛП-реализацией линейного оператора A понимается разложение исходного линейного оператора в бесконечное произведение матриц

$$A = \prod_{i=0}^n B_i, \quad (1)$$

таких, что произведение $\prod_{i=0}^n B_i$ стремится к исходной матрице A при $n \rightarrow \infty$. Элементы матриц B_i выбираются таким образом, что умножение на очередную матрицу ДЛП сводится к последовательности сложений и сдвигов, то есть элементы матриц B_i выбираются равными нулю или целой степени основания системы счисления (в двоичной системе – целой степени двух). Это приводит к понижению аппаратно-временной сложности алгоритма, откуда следует снижение стоимости вычислительных устройств, реализованных на базе этого алгоритма.

Под базовым алгоритмом ДЛП-класса будем понимать базовую реализацию ДЛП-алгоритма с минимально необходимым числом итераций.

$$\left. \begin{array}{l}
 i = 0; \\
 \text{Если } _ \text{точность_не_достигнута, то} \\
 \{ \text{ДЛП} - \text{оператор, } i = i + 1 \}, \\
 \text{иначе } _ \text{выход}
 \end{array} \right\} \text{ДЛП} - \text{итерация} \quad (2)$$

Преобразование начинается с нулевой итерации ($i=0$). Далее на каждой итерации проверяется, не достигнута ли требуемая точность (например, сравнением с эталоном). Если не достигнута, выполняется ДЛП-оператор, инкрементируется номер итерации и далее выполняется следующая ДЛП-итерация. Очевидно, что при базовой реализации ДЛП-алгоритма число итераций для различных наборов входных данных будет отличаться. В частности, для наборов, которые совпадают с результатом, никаких итераций производить вообще не надо. Для других наборов число итераций будет отличаться от нуля. Максимально необходимое число итераций базового алгоритма обозначим через k . Очевидно, что базовый алгоритм может быть использован только при моделировании для определения минимально необходимого числа итераций, т.к. при выполнении реальных вычислений эталонное (точное) значение вычисляемого параметра неизвестно (собственно, оно и должно быть вычислено). Однако базовый алгоритм может быть полезен для определения потенциальных возможностей алгоритмов данного ДЛП-класса.

При программно-аппаратной реализации необходимо изменить базовый алгоритм таким образом, чтобы он мог быть использован для решения конкретных задач. Для этого представляется несколько вариантов.

ДЛП-алгоритм со строго определенным количеством итераций

В данном случае аналитическим путем или путем моделирования определяется число итераций k_1 , при которых ДЛП-алгоритм гарантированно сходится на любом наборе входных данных [2].

$$\left. \begin{array}{l} i = 0; \\ \text{Если } i < k_1, \text{ то} \\ \{ \text{ДЛП} - \text{оператор}, i = i + 1 \}, \\ \text{иначе } - \text{выход} \end{array} \right\} \text{ДЛП} - \text{итерация} \quad (3)$$

Преобразование в данном случае выполняется всегда, в т.ч. и для набора, который совпадает с результатом, а затем, путем последовательного уточнения на каждой итерации, приближает к искомому результату. Значение k_1 зависит от ряда параметров, в т.ч. от требуемой точности вычислений, но, очевидно, что $k_1 > k$, т.е. для данного варианта реализации ДЛП-алгоритма число итераций будет заведомо больше числа итераций базового алгоритма.

Очевидным недостатком данного варианта является выполнение ненужных итераций. В частности, в приведенном выше примере, когда набор входных данных уже совпадает с результатом, итерация все равно будет выполнена, и вычисляемые параметры будут удалены от искомым. Данный недостаток учтен в базовом алгоритме, однако, в нем производится сравнение вычисляемых параметров с эталонным значением, которое при конкретной реализации неизвестно на этапе вычислений. Однако уменьшить количество ненужных итераций можно, если по значениям вычисляемых параметров косвенно оценивать текущую точность вычислений.

ДЛП-алгоритм с оценкой текущей точности вычислений

$$\left. \begin{array}{l} i = 0; \\ \text{Если } _ \text{текущая } _ \text{погрешность } _ \text{больше} \\ \text{необходимой, то} \\ \{ \text{ДЛП} - \text{оператор}, i = i + 1 \}, \\ \text{иначе } - \text{выход} \end{array} \right\} \text{ДЛП} - \text{итерация} \quad (4)$$

Число итераций в данном случае будет равно k_2 , причем $k < k_2 < k_1$.

Еще одним способом, уменьшающим общее число итераций по сравнению с ДЛП-алгоритмом со строго определенным количеством итераций, является отказ от выполнения «неудачных» итераций в процессе преобразования [5]. Не все итерации приближают вычисляемые параметры к искомому результату, т.е. являются «удачными». Для того же

примера с набором входных данных, совпадающим с результатом, любая итерация будет являться «неудачным». Таким образом, выполнение только «удачных» итераций поможет снизить число итераций относительно k_1 [3].

ДЛП-алгоритм с выполнением только «удачных» итераций

$$\left. \begin{array}{l}
 i = 0; \\
 \text{Если } i < k_3, \text{ то} \\
 \{ \\
 \text{Если } _ \text{итерация } _ \text{удачна, то} \\
 \{ \text{ДЛП} - \text{оператор} \} \\
 i = i + 1 \\
 \}, \\
 \text{иначе } _ \text{выход}
 \end{array} \right\} \text{ДЛП} - \text{итерация} \quad (5)$$

Число итераций в данном случае будет равно k_3 , причем $k < k_3 < k_1$. Соотношение итераций k_2 и k_3 будет зависеть от конкретного ДЛП-оператора (отражение, вращение и т.д.).

К наименьшему числу итераций должно привести совмещение технологии оценки текущей точности вычислений и выполнения только «удачных» итераций.

ДЛП-алгоритм с оценкой текущей точности вычислений и выполнением только «удачных» итераций

$$\left. \begin{array}{l}
 i = 0; \\
 \text{Если } _ \text{текущая } _ \text{погрешность } _ \text{больше} \\
 \text{необходимой, то} \\
 \{ \\
 \text{Если } _ \text{итерации } _ \text{удачна, то} \\
 \{ \text{ДЛП} - \text{оператор} \} \\
 i = i + 1, \\
 \text{иначе } _ \text{выход}
 \end{array} \right\} \text{ДЛП} - \text{итерация} \quad (6)$$

Число итераций в данном случае будет равно k_4 .

Очевидно, что

$$\left. \begin{array}{l}
 k < k_4 < k_1 \\
 k_4 \leq k_2 \\
 k_4 \leq k_3
 \end{array} \right\} \quad (7)$$

Учитывая все вышеизложенное, обобщенный класс ДЛП-алгоритмов будет выглядеть следующим образом (табл.1).

Таблица 1. Варианты реализации базового алгоритма дискретного линейного преобразования

№ п/п	Наименование реализации базового алгоритма	Косвенная оценка текущей точности	Определение необходимости выполнения текущей итерации
1	ДЛП-алгоритм со строго определенным количеством итераций	-	-
2	ДЛП-алгоритм с оценкой текущей точности вычислений	+	-
3	ДЛП-алгоритм с выполнением только «удачных» итераций	-	+
4	ДЛП-алгоритм с оценкой текущей точности вычислений и выполнением только «удачных» итераций	+	+

Из анализа табл.1 очевидно, что выполнение базового алгоритма с использованием первой схемы приводит к самой простой реализации отдельной итерации, т.к. нет необходимости постоянно оценивать текущую точность вычислений или оценивать необходимость выполнения текущей итерации. Однако данный вариант приводит к наибольшему числу итераций. Остальные схемы реализации приводят к потенциально более сложной реализации отдельной итерации, однако, это приводит к снижению общего числа итераций.

Наиболее перспективной схемой вычислений, очевидно, является ДЛП-алгоритм с оценкой текущей точности вычислений и выполнением только «удачных» итераций, т.к. он потенциально приводит к наименьшему количеству итераций. Однако для конкретных ДЛП-операторов (отражение, вращение и т.д.) наиболее эффективной может оказаться другая схема, даже вариант со строго определенным количеством итераций, если оценка текущей точности и определение «удачной» итерации является затрудненным или имеет высокую операционную сложность.

Из приведенных выше алгоритмов заданного ДЛП-класса выбор конкретного алгоритма будет зависеть от ДЛП-оператора, а также требований, предъявляемых к вычислительной системе, реализующей алгоритм [1].

Список литературы

1. Андреев А.Е., Стрельников О.И., Егунов В.А. Программное обеспечение реконфигурируемых вычислителей на базе ПЛИС // Информационные технологии в образовании, технике и медицине: материалы международной конференции / ВолгГТУ. – Волгоград, 2006. – С.125-126.

2.Егунов В.А., Андреев А.Е., Стрельников О.И. Моделирование алгоритмов дискретных линейных преобразований // Известия Волгоградского государственного технического университета: межвуз. сб. науч. Ст. №1(27) / ВолгГТУ. – Волгоград, 2007. - С.51-53 [Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. Вып.1].

3.Егунов В.А., Андреев А.Е., Стрельников О.И. Определение условия оптимальности процесса дискретного линейного преобразования отражения // Известия Волгоградского государственного технического университета: межвуз. сб. науч. Ст. №1(27) / ВолгГТУ. – Волгоград, 2007. – С. 53-54 [Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. Вып.1].

4. Егунов В.А. Аппаратные методы решения задач линейной алгебры: монография/ В.А.Егунов, В.С. Лукьянов/ВолгГТУ, Волгоград, 2007. – 152 с.

5. Лукьянов В.С. Алгоритм ускоренного дискретного линейного преобразования отражения / В.С.Лукьянов, В.А.Егунов // Открытое образование: прил. к журн.: по матер. XXXIII межд. конф. к IV межд. конф. мол. уч. IT+S&E'06 (Ялта – Гурзуф, Крым). – 2006. – Б/н (май). – С.105-106.

Рецензенты:

Шилин А.Н., д.т.н., профессор, заведующий кафедрой «Энергообеспечение предприятий» филиала ФГБОУ ВПО «Национальный исследовательский университет (МЭИ)», г. Волжский.

Кравец А.Г., д.т.н., профессор кафедры САПР и ПК, ФГБОУ ВПО Волгоградский государственный технический университет, Волгоград.