УДК 519.6

СТРУКТУРА ИНКРЕМЕНТНЫХ АЛГОРИТМОВ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

О.В. Гомозов
Донецкий национальный технический университет

В докладе описывается группа итерационных алгоритмов решения систем линейных алгебраических уравнений (СЛАУ), названных ”инкрементными”, перспективных для аппаратной реализации, исследуется их быстродействие, сходимость и ресурсоемкость на основе приведенных данных и структур.

Научно-исследовательские, технические и другие задачи часто связаны с решением СЛАУ больших размерностей, описывающих тот или иной процесс. Нахождение результатов решения такой системы является трудоемкой задачей, которая под силу лишь вычислительным устройствам. Для этого существует немало алгоритмов и методов, прямых и итерационных. В докладе исследуются специализированные итерационные алгоритмы[1], целью которых является снижение аппаратных затрат на свою реализацию, а также большое быстродействие, сравнимое с другими методами.

Если преобразовать классическое СЛАУ вида $A \cdot \vec{x} = \vec{b}$ к интегрированию эквивалентной системы линейных дифференциальных уравнений, а именно, $\frac{dx_i}{dt} = b_i - \sum_{i,j=1}^{n} a_{ij} x_i,\ i, j = 1,2....n,\ n - \text{порядок СЛАУ}$, и обозначить левую часть через $\varepsilon_i(x)$, тогда $\varepsilon_i(x) = b_i - \sum_{i,j=1}^{n} a_{ij} x_i$. Используя метод Эйлера[1] получим обобщенные формулы для инкрементных алгоритмов:

$$x_i^{k+1} = x_i^k + \Delta x_i^{k+1}; \qquad \Delta x_i^{k+1} = 2^{-p} \cdot sign(\varepsilon_i^k),$$

$$\varepsilon_i^{k+1} = \varepsilon_i^k + \Delta\varepsilon_i^{k+1}; \quad \Delta\varepsilon_i^{k+1} = -\sum_{j=1}^{n} a_{ij} \Delta x_i^{k+1}, \ \ x_i^0 = 0, \varepsilon_i^0 = b_i,$$

где a_{ij} *– коэффициент матрицы* A, b_i *– свободный член* (1)

x_i *– неизвестная величина*, Δx_i *– приращение неизвестного*

ε_i *– невязка*, $\Delta\varepsilon_i$ *– приращение невязки*

i *– номер неизвестного*, k *– номер итерации*,

p *– разрядность величин*

Под невязкой $\varepsilon_i(x)$ подразумевают промежуточную величину, показывающую отклонение текущего значения неизвестного от истинного на числовой оси. Исходя из формул видно, что эти алгоритмы предполагают использование простых инкрементных операций и операций сдвига вместо сложных операций сложения и умножения. Различие в скорости сходимости определяется только формулами нахождения приращения неизвестного. Существует 4 метода [2]:

Метод №1:

$$\Delta x_i^{k+1} = 2^{-p} \cdot sign(\varepsilon_i^k),\ где\ \ p - разрядность \tag{2}$$

Метод №2:

$$\Delta x^{k+1} = 2^{-r} \cdot sign(\varepsilon^k), где\ \ r = 1,2...p-1, p \tag{3}$$

Метод №3:

$$\Delta x^{k+1} = 2^{-\frac{p}{S}} \cdot sign(\varepsilon^k), где\ S = r, \frac{r}{2}, \frac{r}{3}, ...1, \tag{4}$$

$$каждые\ 2^{\frac{p}{r}}\ итерации\ выбирается\ новое\ S$$

$$(r - количество\ участков, на\ которые\ разбивается\ p)$$

Метод №4:

$$\Delta x^{k+1} = 2^{-z} \cdot U_i \cdot sign(\varepsilon_i^k); \quad z = \min_{i=1..n} m_i;$$

$$2^{-m_i} \le \varepsilon_i^k < 2^{-m_i+1}; \quad U_i = \begin{cases} 1, \varepsilon_i^k \ge 2^{-z} \\ 0, \varepsilon_i^k < 2^{-z} \end{cases} \tag{5}$$

Критерием же сходимости может быть $\sum_{i=1}^{n} (\varepsilon_i^k)^2 \le E$, где E – некоторая заданная погрешность[2].

Количество итераций k зависит от разрядности данных. Для исследования этих методов было проведено моделирование на нескольких видах матриц (диагональ-доминирующая, трехдиагональная, матрица Гильберта), результаты которого представлены в таб. 1.

Таблица 1. Количество итераций в инкрементных методах

Разрядность	16	24	32	48	64
Метод №1	65536	16777216	$>10^9$	$>10^9$	$>10^9$
Метод №2	16	24	32	48	64
Метод №3	512	8192	65536	16777216	$>10^9$
Метод №4	64	128	256	512	1024

Как видно из таблицы, наиболее эффективными являются метод №2 и №4, однако первый из них накладывает ограничения на матрицу А.

Следует также отметить, что исходя из формул (1) возможно параллельное вычисление каждого неизвестного и его промежуточных данных на каждой итерации. Обобщенная функциональная операционного устройства, решающего инкрементными методами представлена на рис.1.

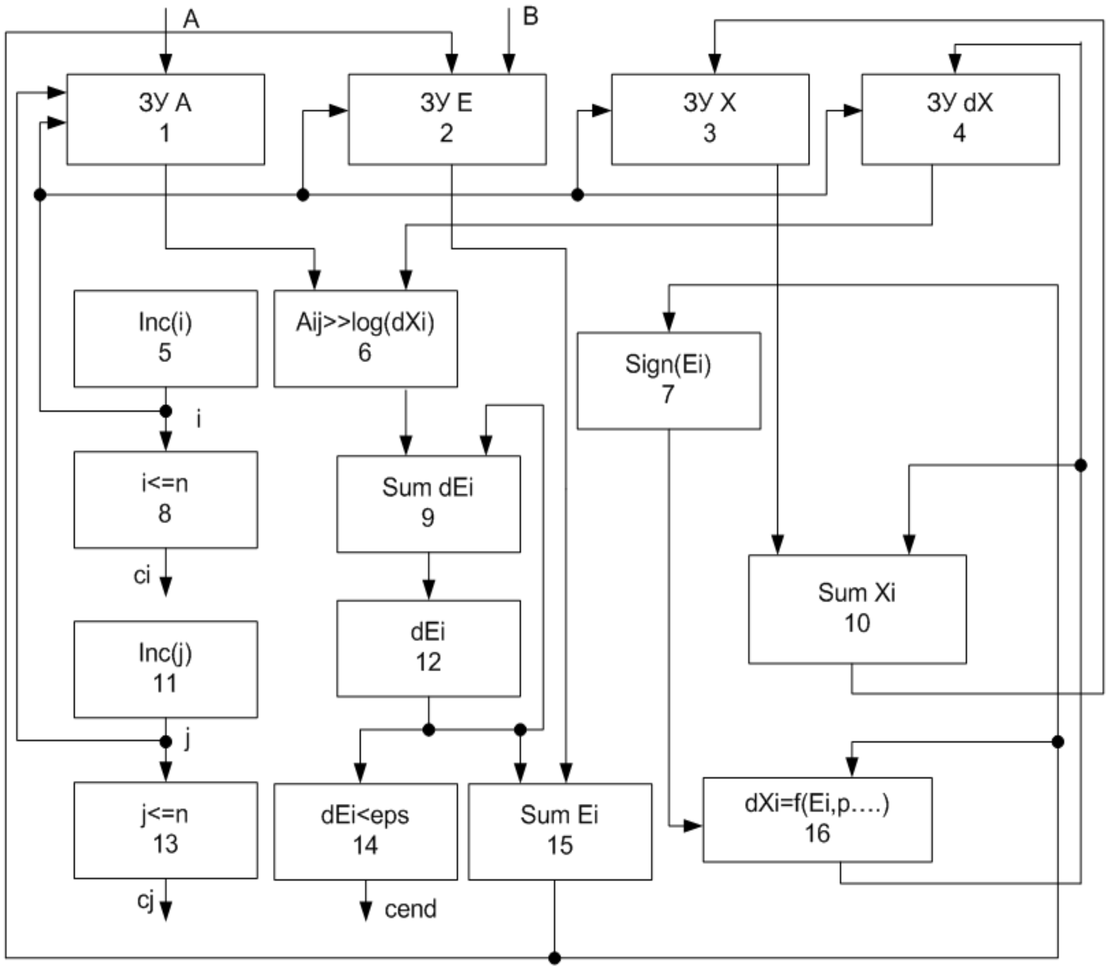

Рис. 1. Функциональная схема операционного устройства

Блоки 1,2,3 и 4 – запоминающие устройства для хранения матрицы А, вектора невязок Е, вектора неизвестных Х и вектора приращений неизвестных dX соответственно. В начале работы в блоки 1 и 2 загружаются матрица А и вектор свободных членов В. Блоки 5 и 8 отвечают за приращение индекса текущего уравнения i и анализа на достижение порядка n СЛАУ. Аналогично работают блоки 11 и 13 для индекса столбца j матрица А. Блок 6 производит операцию сдвига коэффициента $a_{i,j}$ на количество разрядов, равное $\log_2 dX_i$, что равнозначно операции $a_{i,j} * dX_i$. Блоки 9 и 12 представляют из себя сумматор и аккумулирующий регистр для вычисления значения приращения невязки dE_i. Блок 14 производит анализ на конец решения, что означает сходимость невязок к некоторой величине погрешности eps. В 15 происходит вычисление текущей невязки через формулу $E_i = E_i - dE_i$. Знак невязки, полученный в блоке 7, используется в блоке 16 для выполнения одного из алгоритмов нахождения приращения неизвестного, в основе которого также лежит операция сдвига невязки. Последняя операция нахождения неизвестного происходит в блоке 10 по формуле $x_i = x_i + dx_i$.

В данной структуре самыми сложными являются блоки суммирования, а все операции умножения заменены простыми и быстрыми операциями сдвига. Для параллельной реализации блоки 5 и 8 будут отсутствовать, однако количество таких структур увеличиться в n раз. Также возможны другие модификации, например, использование одного сумматора, мультиплексора данных и группы накапливающих регистров вместо трех сумматоров, приведенных на рис. 1.

В условиях вышеописанных свойств инкрементных алгоритмов перспективным направлением для их аппаратной реализации являются интегральные схемы FPGA[3]. Они содержат отдельные программируемые блоки памяти и шины данных, связывающие их. Такая структура позволяет сконфигурировать и запрограммировать микросхемы, наиболее эффективно используя достоинства инкрементных алгоритмов, особенно, распараллеливание процесса вычисления[3]. Реализация на FPGA может быть масштабируема и расширяема по разрядности, количеству решающих блоков и другим параметрам. Кроме того современные FPGA содержат готовые блоки оперативной и постоянной памяти, блоки ввода/вывода через различные интерфейсы, что позволит внедрить подобные устройства во многие вычислительные системы в качестве отдельных модулей. Еще одним достоинством является их дешевизна по сравнению с универсальными или специализированными процессорами, способными решать подобные задачи.

Исходя из вышесказанного, сформированы выводы об основных достоинствах инкрементных алгоритмов:

- высокая скорость решения и простота арифметических операций;
- эффективная реализация на базе микросхем FPGA;

- устойчивость и отсутствие ограничений применимо ко многим классам задач.

Литература

1. Б.Н.Малиновский, В.П.Боюн, Л.Г.Козлов. Алгоритмы решения систем линейных алгебраических уравнений, ориентированные на структурную реализацию. – “Управляющие системы и машины”. Вып. №5. 1977, с.79-84.

2. Боюн В.П., Козлов Л.Г., Малиновский Б.Н., Третьяков С.И. Устройства для решения систем линейных алгебраических уравнений. Автор. свид. № 543943 – БИ, 1977, № 3.

3. Максфилд К. Проектирование на ПЛИС. – М.: ”Додека-XXI”, 2007, 408с.

Получено 29.05.2009 г.