УДК 004.02

МОДИФИКАЦИЯ МЕТОДА ВРАЩЕНИЯ ПРИ АППАРАТНОМ РЕШЕНИИ СЛАУ

Т.А. Едемская
Д.В. Бельков, доц.
Донецкий национальный технический университет, 6otba@list.ru

Аппаратная реализация известных классических методов решения систем уравнений затруднена из-за сложности основных вычислительных операторов. Необходима разработка новых модифицированных алгоритмов решения систем уравнений, ориентированных на реализацию в параллельных вычислительных структурах на основе ПЛИС. В данной работе для реализации на ПЛИС предлагается модифицированный метод вращения. Программа реализована в среде Delphi.

Hardware representation of known classic methods of decision of the systems of equalizations is labored because of complication of basic calculable operators. Development of new modified algorithms of decision of the systems of equalizations, oriented on realization in the parallel calculable structures on the basis of FPGA is necessary. A modified method of rotation is offered to the given work for realization on FPGA. The program is realized in the Delphi environment.

Апаратна реалізація відомих класичних методів рішення систем рівнянь утруднена через складність основних обчислювальних операторів. Необхідна розробка нових модифікованих алгоритмів рішення систем рівнянь, орієнтованих на реалізацію в паралельних обчислювальних структурах на основі ПЛІС. В даній роботі для реалізації на ПЛІС пропонується модифікований метод обертання. Програма реалізована в середовищі Delphi.

СИСТЕМА УРАВНЕНИЙ, ПРОГРАММИРУЕМАЯ ЛОГИЧЕСКАЯ ИНТЕГРАЛЬНАЯ СХЕМА, МЕТОД ВРАЩЕНИЯ, ПРОЦЕССОР CORDIC.

Решение систем линейных алгебраических уравнений (СЛАУ) имеет большое значение в практике вычислений. Это объясняется тем, что линейное приближение многих математических моделей реальных объектов приводит к системам линейных алгебраических уравнений. Задачи решения таких систем возникают при обработке экспериментальных данных, приближенном решении линейных интегральных и дифференциальных уравнений и в других областях. Практические способы решения зависят от структуры исходных данных, порядка матрицы коэффициентов, а также типа используемых вычислительных средств.

В настоящее время большое внимание уделяется пересмотру методов численного анализа и созданию новых алгоритмов, позволяющих наиболее эффективно использовать параллельные вычисления. В значительной мере это связано с развитием параллельных вычислительных систем с большим числом одновременно работающих процессоров [1].

При формальном подходе решение систем уравнений не встречает затруднений. Если задана система $AX = B$, где A - квадратная матрица коэффициентов размером $n \times n$, а B - вектор свободных членов размером $n \times 1$, то решение сводится к нахождению вектора неизвестных X размером $n \times 1$. В общем случае $X = A^{-1}B$ и при любом векторе B решение существует, если $\Delta_A \neq 0$. Однако, этот метод практически непригоден при больших значениях n из-за чрезмерного объема вычислений. Общее число операций для формирования обратной матрицы равно $n^2 \cdot n! - 1$.

Реализация известных классических методов решения СЛАУ затруднена из-за сложности основных вычислительных операторов. Необходима разработка новых модифицированных алгоритмов решения систем уравнений, ориентированных на аппаратную реализацию в параллельных вычислительных структурах на основе Программируемых Логических Интегральных Схем (ПЛИС).

Методы решения алгебраических задач разделяют на прямые и итерационные. Классы задач, для решения которых обычно применяют методы этих групп, можно соответственно назвать классами задач с малым и большим числом неизвестных. Изменение объема и структуры памяти компьютеров, увеличение их быстродействия и развитие численных методов приводят к смещению границ применения методов в сторону систем более высоких порядков. В настоящее время прямые методы обычно применяются для решения систем до порядка 10^4, итерационные – до порядка 10^7.

Большинство прямых методов основано на переходе от заданной системы $AX = B$ к новой системе $CAX = CB$ такой, что система $CAX = CB$ решается проще, чем исходная [1]. При решении системы методом Крамера важным является применение рациональных методов вычисления определителя. В общем случае нахождение определителя требует большого объема вычислений, но определитель треугольной матрицы вычисляется легко, он равен произведению диагональных элементов. Одним из методов, основанным на приведении матрицы системы к треугольному виду, является метод исключения Гаусса. Для приведения матрицы к треугольному виду может использоваться QR-разложение матрицы.

В работе [2] предложен спецпроцессор, ориентированный на QR-разложение матрицы алгоритмом RLS (Recursive Least Square), основанный на методике CORDIC (Coordinate Rotation by Digital Computer). Процессор может быть реализован на ПЛИС Altera Stratix, Xilinx Vertex4-xc4vlx200.

Для системы уравнений XC=Y алгоритм преобразует заданную матрицу X в треугольную матрицу R и вектор Y преобразует в вектор U, такой, что RC=U. Искомый вектор C вычисляется с помощью процедуры обратной подстановки по формулам: $C_n = U_n / R_{nn}$; $C_i = \frac{1}{R_{ii}}(U_i - \sum_{j-i+1}^{n} R_{ij}C_j)$, i=n-1,…,1.

Структура процессора состоит из двух частей, показанных на рисунке 1.

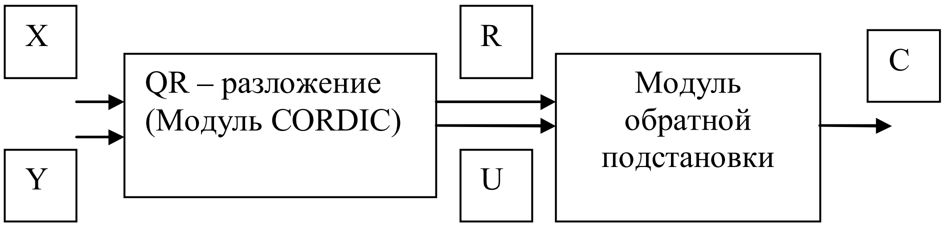

Рисунок 1. – Структура процессора

В данной работе для реализации на ПЛИС решения СЛАУ предлагается модифицированный метод вращения.

Метод вращения является разновидностью метода Гаусса. Исходная система n уравнений за n(n-1)/2 шагов приводится к системе с треугольной матрицей следующим образом [3]:

1. Сформировать матрицу A, такую, что a_{ij} – элементы матрицы коэффициентов, i=1,2,…,n, $a_{i,0} = b_i$, b_i – элементы вектора свободных членов.
2. Вычислить значение $m = \sqrt{a_{ii}^2 + a_{ki}^2}$.
3. Если $a_{ii} = a_{ki} = 0$, то *m:=1* , *l:=0*, иначе – вычислить значения $l = -a_{ki} / m = -\sin(\theta)$, $m = a_{ii} / m = \cos(\theta)$, где θ – угол поворота вектора $\{a_{ii}, a_{ki}\}$.
4. Выполнить преобразование системы по формулам $my_i - ly_k = mb_i - lb_k$, $ly_i - my_k = lb_i - mb_k$, y_i и y_k – левые части i-го и k-го уравнений, b_i и b_k – правые части i-го и k-го уравнений, i=1,2,…,n+1, k=i+1, i+2,…,n.

При известной треугольной матрице корни системы уравнений определяются процедурой обратной подстановки. Самым трудоемким является шаг 2.

Для аппаратного решения СЛАУ методом вращения применяют спецпроцессоры CORDIC, реализованные на ПЛИС [4]. Они работают соответственно в векторном режиме (при определении угла поворота), режиме вращения (при повороте вектора), линейном режиме (при выполнении обратной подстановки). Алгоритм работы в линейном режиме приведен в работе [4] (алгоритм (14)), алгоритм работы в режиме вращения приведен в работе [5] (алгоритм (23)), алгоритм работы в векторном режиме приведен в работе [5] (алгоритм (28)).

Предлагаемая модификация метода вращения заключается в том, что сферическая норма вектора $\{a_{ii}, a_{ki}\}$, вычисляемая на шаге 2 по формуле $m = \sqrt{a_{ii}^2 + a_{ki}^2}$, заменяется кубической нормой вектора: $m = \max\{|a_{ii}|, |a_{ki}|\}$. Вместо вычисления корня из суммы квадратов значений a_{ii}, a_{ki} требуется найти максимум среди модулей этих значений. Таким образом, в векторном режиме определение величины m упрощается, за счет чего скорость аппаратного решения системы может быть повышена. Кроме кубической нормы можно использовать октаэдрическую норму вектора: $m = |a_{ii}| + |a_{ki}|$. В алгоритме векторного режима строка $\begin{pmatrix} x \\ y \end{pmatrix}' = \begin{pmatrix} \sqrt{x^2 + y^2} \\ 0 \end{pmatrix}$ должна быть заменена строкой $\begin{pmatrix} x \\ y \end{pmatrix}' = \begin{pmatrix} \max(|x|, |y|) \\ 0 \end{pmatrix}$ или строкой $\begin{pmatrix} x \\ y \end{pmatrix}' = \begin{pmatrix} |x| + |y| \\ 0 \end{pmatrix}$.

Метод вращения был реализован в среде Delphi:

```
procedure TForm1.Button1Click(Sender: TObject);
label
 m1,m2;
var
  a: array[0..3, 0..3] of real;
    n,i,j,k : integer;
    l,m,r : real;
    {МЕТОД ВРАЩЕНИЯ}
begin
n:=3;
{Первый вариант СЛАУ
 a[1,0]:=4;   a[2,0]:=1; a[3,0]:=3;
    a[1,1]:=3;   a[1,2]:=2; a[1,3]:=1;
    a[2,1]:=1;   a[2,2]:=1; a[2,3]:=-1;
    a[3,1]:=1;   a[3,2]:=-2; a[3,3]:=1;}
    {Второй вариант СЛАУ}
 a[1,0]:=8;   a[2,0]:=9; a[3,0]:=20;
    a[1,1]:=4;   a[1,2]:=0.24; a[1,3]:=-0.08;
    a[2,1]:=0.09;   a[2,2]:=3; a[2,3]:=-0.15;
    a[3,1]:=0.04;   a[3,2]:=-0.08; a[3,3]:=4;
    for i:=1 to n-1 do
      begin {i}
       for k:=i+1 to n do
        begin  {k}
      if a[i,i]<>0 then goto m1;
      if a[k,i]<>0 then goto m1;
     m:=1;
 l:=0;
goto m2;
 m1:
 {НОРМА}
                if a[i,i]>a[k,i] then m:=a[i,i] else  m:=a[k,i];
   l:=-a[k,i]/m; m:=a[i,i]/m;
      for j:=1 to n do
```

```
        begin {j}
     r:=m*a[i,j]-l*a[k,j];
  m2:
     a[k,j]:=l*a[i,j]+m*a[k,j];
     a[i,j]:=r;
        end; {j}
      r:=m*a[i,0]-l*a[k,0];
      a[k,0]:=l*a[i,0]+m*a[k,0];
      a[i,0]:=r;
       end; {k}
        end; {i}
      for i:=n downto 1  do
       begin {i}
         m:=0;
      for k:=0 to n-i-1 do
       m:=m+a[0,n-k]*a[i,n-k];
     a[0,i]:=(a[i,0]-m)/a[i,i];
        end; {i}
     edit1.Text:=floattostr(a[0,1]);
     edit2.Text:=floattostr(a[0,2]);
     edit3.Text:=floattostr(a[0,3]);
end;
```

При тестовом решении двух вариантов систем трех уравнений модифицированным методом получены результаты, совпавшие с результатами исходного метода. Для дальнейшего изучения модифицированного метода необходимо разработать схему устройства, реализующего алгоритм CORDIC, например, в среде Quartus.

Литература

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. Москва: БИНОМ, 2006. – 636 с.
2. Andraka R. A survey of CORDIC algorithms for FPGA based computers. [Электронный ресурс], 2003. – Режим доступа: http://www.andraka.com/files/crdcsrvy.pdf
3. Дьяконов В.П. Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ. Москва.- 1987.- 240 с.
4. “Устройство для решения систем линейных алгебраических уравнений”. [Электронный ресурс], 2003. – Режим доступа: http://vstuhelp.narod.ru/lab/specproc.html.
5. “Изучение процессора CORDIC”. [Электронный ресурс], 2003. – Режим доступа: http://vstuhelp.narod.ru/lab/specproc.html.