СХЕМОТЕХНИКА И ПРОЕКТИРОВАНИЕ

УДК 658.512.011.56

Применение компиляционного подхода к моделированию аналоговых схем

Д.А.Булах

Московский государственный институт электронной техники (технический университет)

Предложен новый способ построения программ автоматизации схемотехнического проектирования, основанный на компиляционном подходе к моделированию схем. Проведено сравнение скорости расчета ряда схем во временной области с результатами программы, использующей традиционный подход.

В основу работы большинства программ автоматизации схемотехнического проектирования (АСхТП) положена следующая последовательность этапов:

1) считывание схемы из описания на одном из схемотехнических языков;

2) трансляция описания схемы во внутреннее представление программы;

3) формирование векторов и матриц, участвующих в расчете математической модели схемы, инициализация начальных значений вектора переменных;

4) решение сформированной математической модели схемы в зависимости от режима моделирования и применяемых методов.

При проведении моделирования последний этап является наиболее ресурсоемким с точки зрения затрат вычислительного времени. Алгоритм этого этапа представлен на рис.1. В качестве примера рассмотрен алгоритм расчета схемы методом узловых потенциалов во временной области, в котором методом интегрирования является неявный метод Эйлера с фиксированным шагом, а для решения системы нелинейных уравнений используются методы Ньютона и Гаусса.

На данном этапе основное вычислительное время занимают процедуры формирования вектора токов схемы, формирования матрицы проводимостей ветвей схемы и решение системы уравнений методом Гаусса [1].

В современных схемотехнических программах-симуляторах используются различные алгоритмы, ускоряющие процесс решения системы нелинейных уравнений. Однако практически не существует эффективных подходов к ускорению процесса формирования математических моделей, т.е. заполнения значениями вектора токов и матрицы проводимостей схемы. На сегодняшний день существуют два наиболее удачных подхода, но каждый из них имеет свои недостатки.

Первый из них заключается в том, что предлагается заполнять вектор и матрицу значениями до начала выполнения внешнего цикла. Затем внутри цикла на каждом шаге перезаписывать только те значения, которые соответствуют нелинейным элементам, поскольку только их значения будут меняться при изменении значений потенциалов.

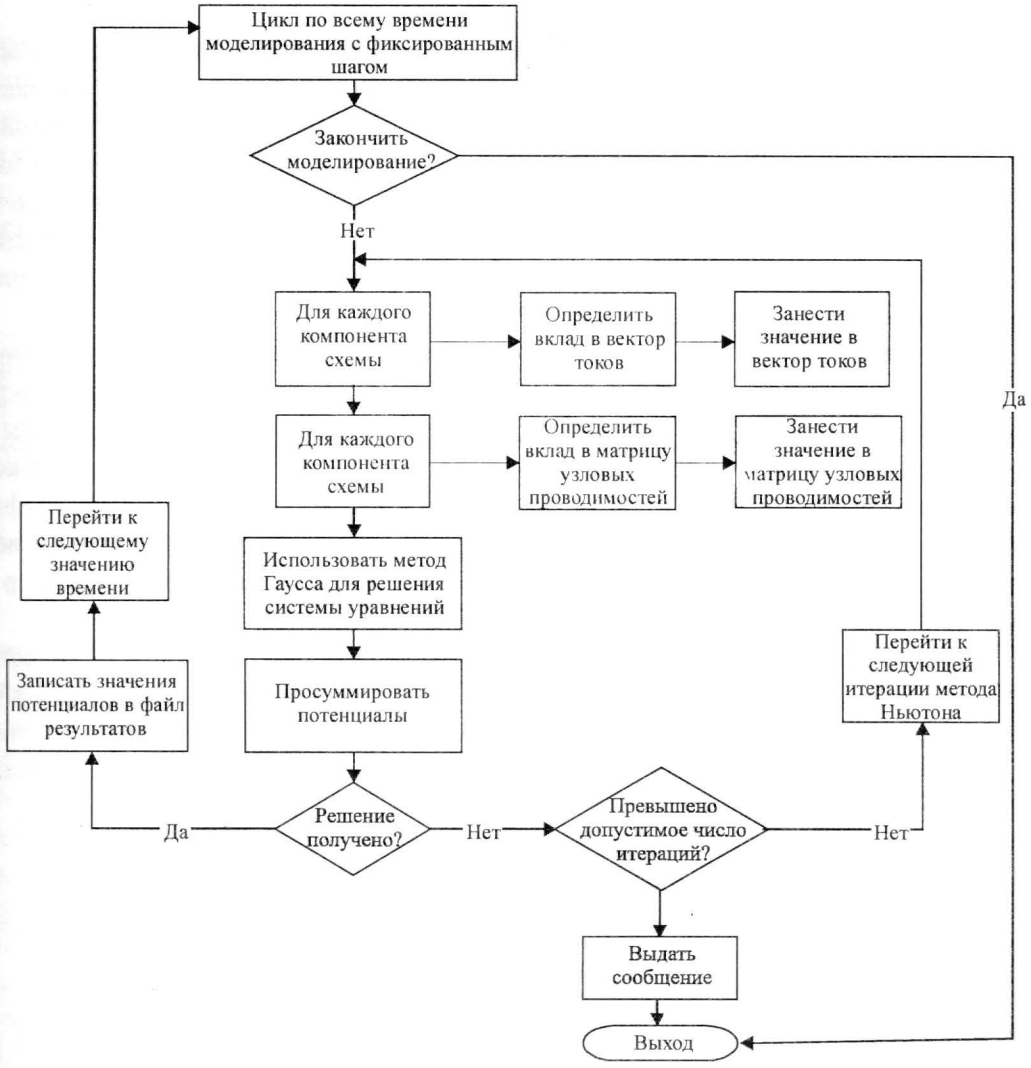

Рис.1. Алгоритм расчета временных характеристик

Такой метод имеет следующие недостатки:

- требуется дополнительная оперативная память компьютера для сохранения номеров нелинейных элементов, которые необходимо перезаписывать;
- поскольку при решении системы уравнений методом Гаусса изменяются значения как элементов исходной матрицы, так и исходного вектора токов, необходимо затрачивать дополнительную оперативную память на хранение двух матриц проводимостей – оригинальную и используемую при решении методом Гаусса, а также на хранение двух векторов токов;
- использование такого метода актуально только для схем, содержащих в процентном соотношении малое число нелинейных элементов.

Второй метод заключается в том, что в программном коде в объекте, отвечающем за каждый узел схемы, сохраняется информация о том, какие элементы подключены к этому узлу, например, в виде указателей на объекты соответствующих элементов. Недостатки этого подхода следующие:

- требуется дополнительная оперативная память на сохранение нескольких указателей для каждого узла;
- указатели для одного и того же элемента, подсоединенного к разным узлам, дублируются.

Использование второго подхода требует меньше времени для моделирования схем, однако у описанных подходов наблюдается один общий недостаток – при определении вклада элемента «резистор» в матрицу проводимостей в программе для получения значения проводимости приходится обращаться к методу объекта, реализующего данный элемент. При моделировании схем больших размерностей, содержащих в процентном соотношении значительное число резистивных элементов, таких, как схемы после экстракции паразитных элементов из топологии, эта процедура может внести значительный вклад в общее время расчета схемы.

Максимально ускорить заполнение значений вектора токов и матрицы проводимостей можно лишь полностью избавившись в программе-симуляторе от процедуры перебора узлов схемы. Аналитически это действие очевидно: составление системы топологических уравнений для всех рассматриваемых узлов схемы [2]. Однако при составлении математической модели схемы программе АСхТП неизвестна конфигурация схемы, поэтому единственным способом заполнения значений матрицы и вектора является прямой перебор ветвей и определение вклада каждой из ветвей в вектор токов и матрицу проводимостей [3].

Более эффективным для скорости работы программы моделирования является решение, исключающее (или сводящее к минимуму) затраты машинного времени на составление вектора токов и матрицы проводимостей. Предлагаемый подход компиляционного моделирования основан на реализации этого решения и предполагает выполнение следующих этапов:

1) считывание схемы из описания на одном из языков описания схем;

2) трансляция схемы во внутреннее представление программы;

3) анализ схемы, генерация программного кода, описывающего процесс моделирования данной схемы;

4) компиляция исходных кодов и запуск скомпилированной программы.

Отличие данного подхода заключается в том, что на этапе 3 выполняется анализ структуры схемы и генерируется исходный файл на языке программирования высокого уровня (например, C++), содержащий конструкции для моделирования требуемой схемы, причем для вектора токов и матрицы проводимостей записываются явные выражения. Помимо программного кода, служащего для составления математической модели и ее решения, в генерируемую программу включается также код для вывода сохраняемой информации в требуемом формате для последующего отображения в графическом постпроцессоре. На этапе 4 программа-симулятор компилирует полученный исходный код и запускает программу на исполнение, ожидая ее завершения.

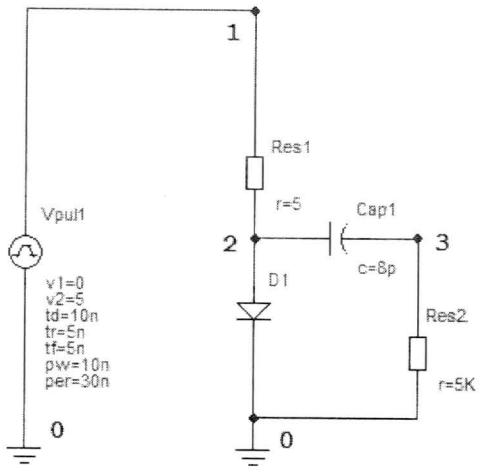

Рис.2. Анализируемая схема

Работа описываемого подхода продемонстрирована на примере расчета тестовой схемы (рис.2). Схема содержит импульсный источник напряжения, к которому подсоединяется определенное число подсхем, каждая из которых содержит диод, два резистора и конденсатор. На рисунке изображена одна подсхема.

При расчете данной схемы методом узловых потенциалов неизвестными являются потенциалы в узлах. Поскольку потенциалы в узлах «0» и «1» известны в любой момент времени, они в процессе моделирования не учитываются. Относительно оставшихся узлов (узлы «2» и «3») формируются математические модели.

Расчет во временной области методом узловых потенциалов сводится к решению на каждом шаге интегрирования системы нелинейных уравнений вида:

$$\mathbf{Ax} = \mathbf{b},$$

где **A** – матрица узловых проводимостей; **x** – вектор неизвестных (вектор потенциалов); **b** – вектор токов.

Размерность вектора токов **b** равна числу неизвестных потенциалов в схеме, каждый элемент вектора токов формируется следующим образом:

$$b[i] = \sum_{j=1}^{k} I_j,$$

где k – число элементов, подсоединенных к i-му узлу; I_j – ток, протекающий через соответствующий элемент.

Для сравнения компиляционного и традиционного способов моделирования написаны две программы: первая программа представляет собой классический SPICE-подобный симулятор, вторая программа – реализацию описываемого подхода. Разработанные программы абсолютно идентичны с точки зрения используемых математических методов (обе программы используют неявный метод Эйлера с фиксированным шагом, метод Ньютона и метод Гаусса) и различаются лишь способами формирования вектора токов и матрицы проводимостей. В качестве сравниваемого симулятора не был взят ни один из уже существующих по причине того, что в существующих симуляторах, например AVOSPICE, HSPICE и SPECTRE, заложены специальные алгоритмы, увеличивающие вычислительную производительность при решении математических моделей.

В общем виде процесс формирования вектора **b** можно выразить следующим участком псевдокода:

```
for(i = 0; i < netlist.nodes_count; i = i + 1) {
   for(j = 0; j < netlist.elements_count; j = j + 1) {
     if(netlist.elements[j].connected_with(netlist.nodes[i]))
        b[i] = b[i] + netlist.elements[j].get_current_trough();
   }
}
```

Матрица **A** представляет собой матрицу узловых проводимостей схемы. Процесс формирования ее элементов представляется в псевдокоде следующим образом (рассмотрен процесс формирования матрицы для двухполюсных элементов):

```
for(i = 0; i < netlist.nodes_count; i = i + 1) {
   for(j = 0; j < netlist.nodes_count; j = j + 1) {
     for(k = 0; k < netlist.elements_count; k = k + 1)
       if(netlist.elements[k].connected_with(netlist.nodes[i])&&
         netlist.elements[k].connected_with(netlist.nodes[j]))
            A[i][j] = A[i][j] + netlist.elements[k].get_conductance();
   }
}
```

Из приведенных листингов псевдокода видно, что процесс составления вектора токов и матрицы проводимостей требует многократного перебора всех элементов схемы, что при большой размерности схемы приводит к значительным затратам времени.

В результате работы программы, реализующий компиляционный подход, формируется код, описывающий явные выражения для всех элементов вектора **b** и матрицы **A**.

Ниже приведен участок кода на языке C++, демонстрирующий процесс формирования математической модели для схемы на рис.2.

```
I[0]= -/*r01*/((v1.v_at(time) - fi_n[0])/5.)
      +/*d01*/(1.2e-10)*(exp((fi_n[0])/0.025) - 1.)
      +/*c01*/((8.e-12)/euler_step)*((fi_n[0]-fi_n[1])-(fi_n_1[0]-
-fi_n_1[1]));
I[1]= +/*r02*/((fi_n[1])/(5.e+3))
      -/*c01*/((8e-12)/euler_step)*((fi_n[0]-fi_n[1])-(fi_n_1[0]-
-fi_n_1[1]));

J[0][0] = -/*r01*/(1./5.)
          -/*d01*/(1.2e-10)*(1./0.025)*exp((fi_n[0])/0.025)
          -/*c01*/((8.e-12)/euler_step);
   J[0][1]   =   +/*c01*/((8.000000e-
012)/euler_step);
   J[1][0]   =   -/*c01*/(-(8.000000e-
012)/euler_step);
   J[1][1] = -/*r02*/(1./5.e+3)
             -/*c01*/(8.000000e-
012)/time_step);
```

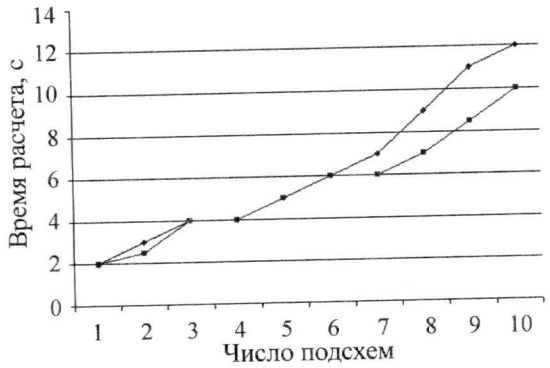

Рис.3. Результаты сравнения симуляторов:
SPICE-подобный симулятор;
компилирующий симулятор

Результаты сравнения двух программ представлены на рис.3.

Видно, что при использовании одинаковых методов, заложенных в основу вычислений, компиляционный подход к формированию математических моделей сокращает общее время расчета схем.

Литература

1. ***Albert Tatum Davis***. Implicit Mixed-Mode Simulation of VLSI circuits: PhD dissertation. 1991.

2. ***Казённов Г.Г.*** Основы проектирования интегральных схем и систем. - М.: «БИНОМ. Лаборатория знаний». – 2005.

3. ***Ильин В.Н., Коган В.Л.*** Разработка и применение программ автоматизации схемотехнического проектирования. – М.: Радио и связь. – 1984. – 368 с.

Статья поступила
12 февраля 2008 г.

Булах Дмитрий Александрович – старший преподаватель кафедры проектирования и конструирования интегральных микросхем МИЭТ. *Область научных интересов*: автоматизация проектирования, программирование алгоритмов САПР.