

Gladkov Leonid Anatol'evich – Southern Federal University; e-mail: leo@tgn.sfedu.ru; 44, Nekrasovskiy lane, Taganrog, 347928, Russia; phone: +78634371625; CAD department; associated professor.

Gladkova Nadezhda Viktorovna – e-mail: leo_gladkov@mail.ru; phone: +78634393260; DM&MO department; senior teacher.

Leiba Sergey Nikolaevich – e-mail: lejba.sergej@mail.ru; phone: +78634371625; CAD department; postgraduate student.

УДК 681.5.01:658.512.2

DOI 10.23683/2311-3103-2018-4-24-38

В.Н. Гридин, В.И. Анисимов, М.М. Абухазим

МЕТОДЫ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ МОДЕЛИРОВАНИЯ БОЛЬШИХ ИНТЕГРАЛЬНЫХ СХЕМ НА ОСНОВЕ КОМПАКТНОЙ ОБРАБОТКИ РАЗРЕЖЕННЫХ МАТРИЦ

Рассматриваются методы повышения производительности систем моделирования больших интегральных схем на основе компактной обработки разреженных матриц. Показывается, что вследствие наличия в математическом описании разреженных матриц возникает задача изменения традиционных подходов к формированию и решению систем уравнений на основе полного описания используемых матриц. Отмечается, что известные методы компактной обработки разреженных матриц значительно отличаются друг от друга по своей эффективности, а также по трудовым затратам при разработке программного обеспечения, реализующего сжатие данных на этапе формирования описания компонентов и в процессе решения уравнений. Приводится детальный сопоставительный анализ возможных методов компактной обработки разреженных матриц с целью установления рекомендаций по разработке высокопроизводительного программного обеспечения систем моделирования больших интегральных схем на основе сжатия данных. Делается вывод, что наибольшую эффективность с точки зрения экономии памяти представляет метод структурно-симметричных матриц. Вместе отмечается практическая значимость использования метода индексно-адресных матриц, который позволяет при создании высокопроизводительных систем автоматизированного проектирования наиболее простым путем реализовать построение программного обеспечения системы моделирования. Отмечается, что классическая форма всех методов фиксированного формата не позволяет перечислять ненулевые элементы в произвольном порядке и включать дополнительные элементы в компактное описание. Поэтому в классической форме метод структурно-симметричных матриц не позволяет непосредственно применять его для обработки компактного описания при решении систем уравнений вследствие неизбежного появления новых ненулевых элементов в процессе этого решения. В основе предложенной в статье методики сжатия данных на основе структурно-симметричных матриц лежит двухэтапная технология построения программного обеспечения системы, при этом на первом, топологическом этапе решается задача определения формата описания с учетом возможного появления новых ненулевых элементов в процессе решения уравнений моделируемой системы, а на втором этапе формируется математическое описание задачи в полученном на первом этапе расширенном формате.

Системы автоматизированного проектирования; автоматизации схемотехнического проектирования; сжатие данных; моделирование систем; компактная обработка; разреженные матрицы.

V.N. Gridin, V. I. Anisimov, M.M. Abukhazim

METHODS FOR IMPROVING THE PERFORMANCE OF LARGE-SCALE INTEGRATED CIRCUIT SIMULATION SYSTEMS BASED ON COMPACT PROCESSING OF SPARSE MATRICES

The methods of increasing the productivity of large-scale integrated circuit simulation systems based on compact processing of sparse matrices are considered. It turns out that due to the presence of sparse matrices in the mathematical description, arises the problem of changing the traditional approaches to the formation and solution of equation systems based on a complete description of the matrices used. It is noted that the well-known methods of compact processing of sparse matrices differ significantly in their efficiency, as well as in labor costs in the development of software that implements data compression at the stage of the description of components and in the process of solving equations. A detailed comparative analysis of possible methods for compact processing of sparse matrices is given with the aim of establishing recommendations for the development of high-performance software for modeling large-scale integrated circuits based on data compression. It is concluded that the most effective from the point of view of saving memory is the method of structurally symmetric matrices. It is noted that the use of the index-address matrices method is of practical importance, which makes it possible, in creating high-performance computer-aided design systems, to implement software modeling software in the simplest way. It is noted that the classical form of all methods of fixed format does not allow to list non-zero elements in an arbitrary order and include additional elements in a compact description. Therefore, in the classical form, the method of structurally symmetric matrices does not allow one to apply it directly to the processing of a compact description when solving systems of equations due to the inevitable appearance of new nonzero elements in the process of this solution. At the heart of the data compression technique proposed on the basis of structurally symmetric matrices lies the two-stage technology for constructing the software of the system, while the first, topological stage solves the problem of determining the format of the description, taking into account the possible appearance of new non-zero elements in the process of solving the equations of the modeled system, and the second step is the formation of a mathematical description of the problem in the extended format obtained at the first stage.

Systems of computer-aided design; circuit design automation; data compression; system modeling; compact processing; sparse matrixes.

Введение. При построении систем моделирования больших интегральных схем становятся весьма актуальными задачи увеличения производительности таких систем. Вследствие присутствия в математическом описании моделируемых систем разреженных матриц возникает задача изменения традиционных подходов к формированию и решению систем уравнений на основе полного описания используемых матриц. Это вызвано, с одной стороны, требованием экономии памяти, которую нежелательно использовать для хранения нулевых элементов, а с другой стороны, необходимостью устранения арифметических операций с нулевыми элементами с целью повышения быстродействия системы моделирования.

Степень разреженности матрицы α оценивается отношением числа ненулевых элементов разреженной матрицы m к общему числу n^2 элементов в полностью заполненной матрице n -го порядка, при этом всегда $m \ll n^2$. Существующие методы компактной обработки разреженных матриц значительно отличаются по своей эффективности и по дополнительным трудовым затратам, которые неизбежно возникают при разработке усложненного программного обеспечения, реализующего компактную обработку разреженных матриц как на этапе формирования описания компонентов, так и в процессе решения уравнений [1–3].

Поэтому для правильного выбора наиболее рационального метода компактной обработки разреженных матриц при разработке программного обеспечения систем моделирования больших интегральных схем необходим детальный сравнительный анализ эффективности возможных методов сжатия данных [4–7].

В настоящей статье приводится детальный сопоставительный анализ возможных методов повышения производительности систем моделирования больших интегральных схем на основе компактной обработки разреженных матриц. При этом учитывается эффективность методов как на этапе формирования математического описания компонентов, так и в процессе решения систем уравнений моделируемой схемы.

Методы связанных списков. Для основного метода связанных списков [8] необходимо создать следующие компактные массивы:

WZ – для значений ненулевых элементов W_{ij} исходной матрицы, которые располагаются в этом массиве по относительному адресу a , имеющему значение порядкового номера элемента в массиве WZ;

WI – для номеров строк ненулевых элементов W_{ij} ;

WJ – для номеров столбцов ненулевых элементов W_{ij} ;

NR – для хранения относительного адреса a следующего ненулевого элемента W_{ij} строки;

NC – для хранения относительного адреса a следующего ненулевого элемента W_{ij} столбца;

ER – для хранения относительного адреса a начального элемента очередной строки;

EC – для хранения относительного адреса a начального элемента очередного столбца.

Достоинством метода связанных списков является возможность записи элементов массива WZ в любом порядке, и как следствие, возможность введения в описание дополнительных ненулевых элементов, которые неизбежно появляются при решении уравнений. Следует также отметить, что классический метод связанных списков позволяет сканировать исходную матрицу, как по строкам, так и по столбцам, что является также достоинством метода. Вместе с тем некоторым недостатком метода связанных списков является значительное число массивов, необходимых для хранения информации о ненулевых элементах исходной матрицы.

Если учесть, что массивы WZ, WI, WJ, NR, NC имеют длину m , определяемую числом ненулевых элементов, а массивы ER, EC имеют длину n , определяемую порядком исходной матрицы, то эффективность использования памяти для данных T типа double, может быть определена для метода связанных списков выражением:

$$\beta = \frac{8n^2}{16 * n^2 * \alpha + 4 * n} \cong \frac{1}{2 * \alpha}.$$

Из полученного выражения видно, что с уменьшением степени разреженности матрицы α эффективность основного метода связанных списков неограниченно возрастает.

Сокращенные методы связанных списков позволяют уменьшить число используемых массивов путем исключения из полного описания ряда массивов. При этом возможно построение сокращенного метода, который реализует сканирование лишь по строкам, для чего необходимо ввести только массивы WZ, WJ, NR, ER, и построение сокращенного метода, который реализует сканирование только по столбцам, для чего необходимо использовать только массивы WZ, WI, NC, EC.

Эффективность любой из схем сокращенного метода связанных списков может быть определена выражением:

$$\beta = \frac{8 * n^2}{n^2 * \alpha * 12 + n * 2} \cong \frac{1}{1,5 * \alpha}.$$

Очевидно, что эффективность сокращенной схемы связанных списков выше, чем эффективность основной схемы, однако это достигается за счет ограничения возможности выбора направления сканирования, что является некоторым недостатком сокращенного метода связанных списков. Однако возможность введения в описание дополнительных ненулевых элементов, а так же, произвольный порядок их записи в массиве остается в силе. Это обстоятельство является существенным достоинством сокращенного метода связанных списков.

Вместе с тем практическая реализация методов связанных списков связана с необходимостью полной перекодировки всех массивов при ведении в описание дополнительных элементов, что связано с большими техническими трудностями в процессе построения программного обеспечения и снижением производительности его дальнейшего функционирования. Возникающие проблемы могут быть устранены путем перехода к объектно-ориентированному подходу описания схемы связанных списков на основе существующей в современных языках программирования C# и Java методики работы с коллекциями объектов класса ArrayList предназначенный для поддержки динамических массивов, размеры которых могут изменяться в процессе работы [9–13].

При этом для решения задачи построения компактного описания необходимо создание объектов rows и columns класса ArrayList [14]. Каждый элемент объекта rows соответствует определенной строке матрицы и ссылается либо на ненулевой элемент в данной строке, либо указывает в null, если в строке элементов нет, а каждый элемент объекта columns соответствует определенному столбцу матрицы и ссылается либо на ненулевой элемент в данном столбце, либо указывает в null, если в столбце отсутствуют элементы.

Описание элементов в списке задается в классе Element, члены которого val, i, j описывают значение элемента w_{ij} исходной матрицы и значения, определяющие его координаты i, j , а ссылки nextI, nextJ указывают на следующий элемент в строке и на следующий элемент в столбце соответственно.

Реализация методов включения в список новых элементов и методов для доступа к элементам списка базируется на использовании стандартных методов, определенных в интерфейсах класса ArrayList.

Общая структура, отображающая связь списков с объектами rows и columns, для полной схемы связанных списков приведена на рис. 1.

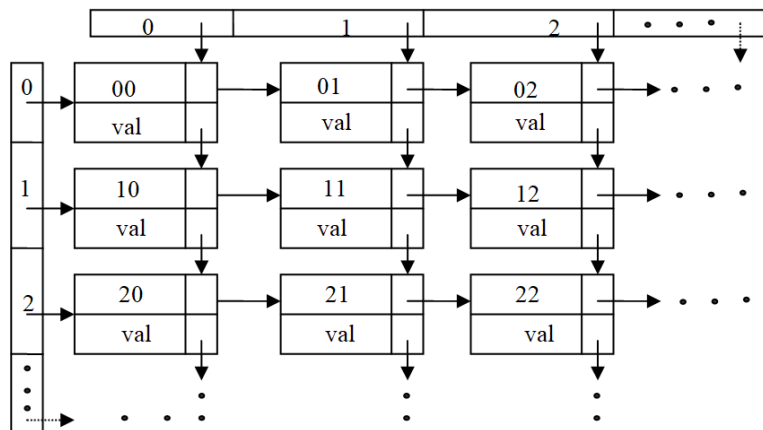


Рис. 1. Связь списков с объектами rows и columns

Эффективность списковой схемы при использовании класса коллекций по сравнению с классической схемой связанных списков несколько снижается, так как для каждого объекта класса `Element` помимо значений `val`, `i`, `j` необходимо также хранить значения ссылок `nextI` и `nextJ`. При этом эффективность полной схемы определяется выражением

$$\beta = \frac{8n^2}{24n^2\alpha + 8n} \cong \frac{1}{3\alpha}.$$

Аналогичным образом могут быть реализованы и сокращенные схемы связанных списков.

Методы фиксированного формата. К методам фиксированного формата относятся метод строчного фиксированного формата и метод сжатия данных на основе структурно-симметричных матриц [1].

При сжатии данных исходной разреженной матрицы W , содержащей m ненулевых элементов, на основе метода строчного фиксированного формата, требуются следующие массивы:

WZ – массив для хранения значения ненулевых элементов w_{ij} исходной матрицы W .

WJ – массив для хранения индексов столбцов ненулевых элементов исходной матрицы W .

ER – массив, содержащий относительные адреса точек входа в очередную строку исходной матрицы W .

Массивы WZ , WJ содержат m элементов, а длина массива ER составит $n + 1$ элементов, при этом в $(n + 1)$ – й элемент массива ER заносится значение $m + 1$.

Если учесть, что длина массивов WZ и WJ составляет m элементов, а длина массива ER составляет $n + 1$ элементов, то коэффициент эффективности метода структурно-симметричного фиксированного формата может быть определена выражением:

$$\beta = \frac{8 * n^2}{n^2 * \alpha * 10 + (n + 1) * 2} \cong \frac{1}{1,25 * \alpha}.$$

Таким образом, эффективность метода повышается при уменьшении коэффициента разреженности.

Отличительной особенностью метода строчного фиксированного формата является невозможность произвольного выбора порядка записи ненулевых элементов в массиве WZ , и как следствие, невозможность включения в описание дополнительных ненулевых элементов. Это объясняется тем, что формат всех массивов жестко зафиксирован, и не может меняться произвольным образом в процессе расчета. Таким образом, классический метод строчного фиксированного формата не позволяет непосредственно применять его для обработки компактных матриц при решении систем уравнений вследствие неизбежного появления новых ненулевых элементов в процессе этого решения.

В методе структурно-симметричных матриц также используется фиксированный формат, однако, в отличие от предыдущего метода, предполагается, что исходная матрица является структурно симметричной, так что для каждого ненулевого элемента w_{ij} можно поставить в соответствие элемент w_{ji} . В случае если такой элемент в исходной матрице отсутствует, то его необходимо создать искусственно путем включения в компактное описание элемента $w_{ji} = 0$.

Метод структурно-симметричных матриц основан на приведении исходной матрицы к форме, содержащей диагональную, наддиагональную и поддиагональную части. Для сжатия данных исходной разреженной матрицы W , содержащей m ненулевых элементов, на основе метода структурно-симметричного фиксированного формата требуются следующие массивы:

WD – для хранения диагональных элементов,

WL – для хранения ненулевых элементов, расположенных ниже диагонали (поддиагональных элементов),

WU – для хранения ненулевых элементов, расположенных выше диагонали (наддиагональных элементов).

При формировании массива WU наддиагональные элементы записываются по строкам, а при формировании массива WL поддиагональные элементы записываются по столбцам. Согласно принятому порядку формирования массивов WU, WL, относительный адрес a некоторого элемента w_{ij} , расположенного в массиве WU, совпадает с относительным адресом элемента w_{ji} , расположенном в массиве WL, что существенно упрощает процесс программирования для организации сканирования элементов.

Хранение индексов строк и столбцов ненулевых элементов в методе структурно-симметричного фиксированного формата осуществляется в массиве WJI, при этом этот массив содержит номера столбцов ненулевых элементов, расположенных выше диагонали, которые совпадают с номерами строк транспонированных ненулевых элементов, расположенных ниже диагонали.

Для определения точки входа в строку выше диагонали (точка входа в столбце ниже диагонали) используется массив ERC. В последний n -ый элемент этого массива заносится значение $\frac{m-n}{2} + 1$.

Поскольку длина массива WD составляет n элементов, длина массивов WU, WL, WJI составляет $\frac{m-n}{2}$ элементов, а длина массива ERC составляет n элементов, то коэффициент эффективности метода структурно-симметричного фиксированного формата может быть определен выражением:

$$\beta = \frac{8 * n^2}{n^2 * \alpha * 9 + n} \cong \frac{8}{9} = \frac{1}{1,1 * \alpha}.$$

Следовательно, эффективность метода структурно-симметричного фиксированного формата выше, чем метода строчного фиксированного формата.

Так же, как и для метода строчного фиксированного формата, метод структурно-симметричного фиксированного формата не позволяет перечислять ненулевые элементы в произвольном порядке и поэтому отсутствует возможность включения дополнительных элементов в компактное описание. Следовательно, в классической форме этот метод неприменим для обработки компактных матриц при решении систем уравнений вследствие неизбежного появления новых ненулевых элементов в процессе этого решения. Поэтому для практического использования методов фиксированного формата для моделирования систем необходимо реализовать расширение формата с целью резервирования в нем дополнительных позиций для размещения новых ненулевых элементов, появляющихся в процессе решения уравнений.

Отсюда следует, что рассмотренные схемы сжатия данных в фиксированном формате могут быть практически использованы только для хранения в компактной форме разреженных матриц. Что же касается полного решения задачи компактной обработки информации в процессе моделирования систем, когда требуется выполнять решение сформированных уравнений, то рассмотренные схемы сжатия данных в фиксированном формате не позволяют решить такую задачу.

Метод индексно-адресных матриц. Метод индексно-адресных (метод адресации) основан на введении некоторой целочисленной матрицы A , которая в точности повторяет структуру исходной разреженной матрицы, входящей в уравнение системы моделирования. В качестве элементов индексно-адресная матрица содержит порядковый номер a ненулевых элементов, которые перечисляются в некотором массиве WZ [15, 16]. При этом если порядковый номер ненулевого элемента исходной матрицы равен a , то в индексно-адресную матрицу вводится значение $A(i,j)=a$.

При реализации метода индексно-адресных матриц достаточно провести сканирование индексно-адресной матрицы и выбрать из нее очередной ненулевой элемент. В соответствии с ее численным значением, затем необходимо выбрать соответствующий порядковый элемент из массива WZ . Эффективность метода индексно-адресных матриц в случае, когда для хранения индексно-адресной матрицы A используется тип данных длиной два байта, а для хранения каждого значащего элемента исходной матрицы, используются данные длиной 8 байтов, может быть определена выражением:

$$\beta = \frac{8n^2}{n^2(\alpha * 8 + 2)} = \frac{1}{\alpha + 0,25}.$$

При работе с комплексными матрицами, когда для хранения каждого значащего элемента исходной матрицы используются данные длиной 16 байтов, эффективность метода индексно-адресных матриц определяется соотношением

$$\beta = \frac{16n^2}{n^2(\alpha * 16 + 2)} = \frac{1}{\alpha + 0,125}.$$

Из приведенного соотношения видно, что коэффициент экономии памяти для данного метода увеличивается с уменьшением коэффициента разреженности α .

При практической реализации задачи построения программного обеспечения для работы с индексно-адресными матрицами целесообразно применить для ее решения существующую в современных языках программирования C# и Java технологию работы с коллекциями, и использовать класс `ArrayList`, предназначенный для поддержки динамических массивов, размеры которых могут изменяться в процессе работы. Для этого класса необходимо создать два объекта – для параметров схемы (например, WZ) и для задающих источников (например, SZ), ссылки на которые удобно объявить в специально созданном классе глобальных переменных (например, GV). В классе глобальных переменных следует также объявить ссылку типа `short` на индексно-адресный массив A . Создание самих объектов выполняется в произвольной функции после ввода значений всех переменных для размерности схемы.

При построении программного обеспечения для работы с индексно-адресной матрицей необходимо решать две основные задачи. Первая задача связана с формированием компактного описания системы, вторая – с решением уравнений на основе созданного компактного описания.

Целью формирования компактного описания системы является занесение в массивы WZ и SZ некоторых значений, определяемых параметрами компонентов и структурой схемных связей. Так, если в компактный массив WZ необходимо зане-

сти некоторое значение z , которое в полном описании разреженной матрицы должно было бы заноситься в i -ю строку и j -й столбец этой матрицы, то при занесении в компактный массив WZ такая задача решается путем постоения функции $set(int\ i, int\ j, Tz)$, где $T=complex$ для частотной области и $T=double$ для временной области и моделирования стационарного режима.

Описание этой функции имеет вид

```
void set(int i, int j, T z)
{ int a = GV.A[i, j];
  if (a != 0) GV.WZ[a] = (T)GV.WZ[a] + z;
  else
    { GV.WZ.Add(z);
      GV.A[i, j] = (short)(GV.WZ.Count - 1); }
}
```

Аналогичным образом формируется и компактный массив SZ .

При решении уравнений любым из известных методов (метод Гаусса, Гаусса-Жордано, LU-факторизация) при использовании полного описания разреженной матрицы w необходимо выполнять k -й шаг приведения к треугольной или диагональной матрице

$$w[i, j] = w[i, j] - w[i, k] * w[k, j] / w[k, k];$$

При использовании компактного описания такая задача решается путем вызова функции $step(int\ i, int\ j)$, которая имеет описание

```
void step(int i, int j)
{ if (GV.A[i, j] == 0)
  { GV.WZ.Add(new T(0, 0));
    GV.A[i, j] = (short)(GV.WZ.Count - 1); }
  GV.WZ[GV.A[i, j]] = (T)GV.WZ[GV.A[i, j]] - (T)GV.WZ[GV.A[i, k]] *
  (T)GV.WZ[GV.A[k, j]] / (T)GV.WZ[GV.A[k, k]];
}
```

Здесь $T=complex$ для частотной области и $T=double$ для временной области и для моделирования стационарного режима.

В результате выполнения этой функции в массиве WZ будет выполнен k -й шаг виртуального преобразования к диагональной единичной матрице. Аналогичным образом решается задача преобразования массива SZ .

При решении уравнений важным вопросом является стратегия оптимального упорядочивания строк для выбора на каждом шаге главного (ведущего) элемента. Такое упорядочивание преследует две цели – сохранение числовой устойчивости и минимизацию числа операций при решении систем уравнений. Для матриц общего вида устойчивость вычислительного процесса обеспечивается при использовании традиционного подхода, основанного на выборе на каждом шаге наибольшего по модулю элемента матрицы. Однако такая стратегия не обеспечивает минимизацию числа новых ненулевых элементов, которые неизбежно появляются при решении уравнений любым известным способом. Поэтому при решении уравнений на основе компактной обработки разреженных матриц процедура оптимального упорядочивания должна быть основана на стратегии минимизации числа новых ненулевых элементов с проверкой условия, что модуль главного элемента не превышает некоторого значения ε (обычно считаются допустимыми значениями $\varepsilon = 10^{-3} - 10^{-5}$).

При этом возможно реализовать три подхода к процедуре оптимального упорядочивания

1. Исходные уравнения (исходная матрица располагается в соответствии с числом ненулевых элементов в порядке их возрастания)

2. На каждом шаге в качестве ведущей выбирается строка, для которой существует минимальное число потенциально-возможных новых ненулевых элементов (алгоритм минимальной степени)

3. На каждом шаге в качестве ведущей выбирается строка, для которой существует минимальное число реальных ненулевых элементов (алгоритм минимизации локального заполнения).

Первая методика может обеспечить эффективный результат только на первом шаге, на последующих шагах, в связи с появлением новых ненулевых элементов расположение строк по числу ненулевых элементов оказывается не упорядоченным, и минимизация появления новых ненулевых элементов не происходит

Вторая методика реализуется исходя из выбора минимального значения произведения $R_k C_k$, где C_k – число ненулевых элементов в k - строке на k -ом шаге, а C_k – число ненулевых элементов в k -ом столбце. Применительно к структурно-симметричной матрицы, $R_k = C_k$ алгоритм минимальной степени приводит к требованию минимизации числа элементов в k -строке.

Третья методика является усложненным вариантом второй методики и по своим результатам незначительно отличается от алгоритма минимальной степени.

Поэтому для решения уравнений на основе компактной обработки разреженных матриц процедура оптимального упорядочивания должна быть основана на стратегии минимизации числа новых ненулевых элементов на основе алгоритма минимальной степени.

Двухэтапная технология сжатия данных. Поскольку фиксированный формат устанавливается для исходного описания, то для введения новых ненулевых элементов в таком формате не зарезервировано требуемое место и для возможности использования методов фиксированного формата необходима модификация этих методов. Для возможности учета новых ненулевых элементов в компактном описании моделируемой системы в фиксированном формате в качестве такой модификации следует использовать двухэтапную процедуру формирования компактного описания [17, 18].

Двухэтапная процедура сжатия данных для обоих методов фиксированного формата выполняется по однотипной структурной схеме. При этом на первом этапе должна решаться задача формирования топологического образа моделируемой схемы, целью которого является определение размеров всех используемых массивов, однако не ставится задача численного формирования этих массивов. Так как на первом этапе не ставится задача формирования численных массивов, то выполнение этого этапа может быть осуществлено путем заполнения всех массивов произвольными численными константами, которые должны отобразить наличие или отсутствие соответствующих элементов матрицы некоторой информацией. Следовательно, на первом этапе работа над численной матрицей может быть заменена обработкой некоторой индексной матрицы S , элементы которой имеют только два произвольных значения, например, (0 и 1). При этом, если в исходной матрице некоторый элемент $w_{ij} \neq 0$, то для введенной индексной матрицы S , соответствующий элемент $S_{ij} = 1$, а все остальные элементы равны 0. Таким образом, вместо рассмотрения исходной схемы, будет рассматриваться топологический образ этой схемы, в точности отображающий ее структуру, но не содержащий информации о численных значениях параметров.

Численное формирование описания реализуется на втором этапе, при выполнении которого в известном формате формируются численные массивы. Такой формат должен содержать зарезервированные места для всех ненулевых элемен-

тов. Следует отметить, что процесс моделирования электронных схем, как правило, связан с выполнением многовариантных или итерационных расчетов одной и той же схемы. Поэтому топологический этап в этом процессе выполняется единственный раз, в то время как численный этап выполняется многократно и временные затраты на выполнение топологического этапа в общих временных затратах на процесс моделирования оказываются весьма незначительными.

Структурная схема двухэтапной процедуры сжатия данных для метода структурно-симметричных матриц приведена на рис. 2. Для метода строчного фиксированного формата структурная схема имеет аналогичный вид, отличие заключается только в обозначениях обрабатываемых массивов (вместо массивов WJI, ERC обрабатываются массивы WJ, ER, а вместо массивов WD, WU, WL – массив WZ).

Первый этап начинается с формирования индексного массива C , который открывается исходя из максимально-возможного в задаче числа переменных n моделируемой схемы. В индексном массиве отображается информация о структуре расположения ненулевых элементов в исходной матрице W . В нулевом столбце матрицы C целесообразно расположить описание структуры ненулевых элементов в задающем векторе S . В нулевой строке матрицы C следует отобразить содержание вектора перестановок P . В $n+1$ столбце содержится информация о числе ненулевых элементов в соответствующей строке в заштрихованной части матрицы. Эта информация будет использоваться для принятия решения о выборе очередного шага оптимальных перестановок строк и столбцов.

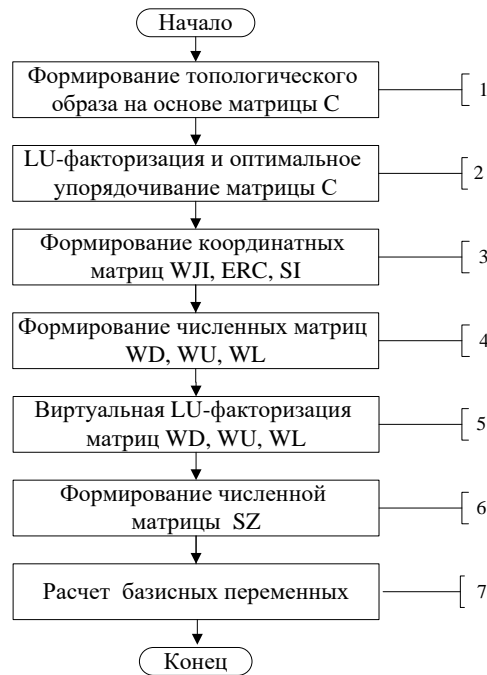


Рис. 2. Структурная схема двухэтапной процедуры сжатия данных для метода структурно-симметричных матриц

Если в полной разреженной матрице элемент $w[i, j]$ не равен нулю, то соответствующему элементу индексного массива $C[i, j]$ должно присваиваться единичное значение. Такая задача практически реализуется путем создания совокупности

функций $sform_comp(C)$ для всех типов компонентов $comp$ моделируемой схемы. Такие функции могут быть легко построены на основании частных матриц компонентов заменой входящих в них параметров единичными элементами.

На втором шаге топологического этапа проводится LU-факторизация индексной матрицы C , с приведением оптимального упорядочивания строк и столбцов с целью минимизации числа появления новых ненулевых элементов. На этом же шаге осуществляется построение вспомогательного вектора P , отображающего все выполненные перестановки строк (столбцов) матрицы C . Оптимальная перестановка строк и столбцов реализуется согласно алгоритму минимальной степени, исходя из выбора минимального значения произведения величины $R_k C_k$, где C_k – число ненулевых элементов в k -й строке на k -ом шаге, а R_k – число ненулевых элементов в k -ом столбце. Применительно к структурно-симметричной матрице имеет место соотношение $R_k = C_k$, т.е. алгоритм минимальной степени приводит к требованию минимизации числа элементов в k -й строке.

При рассмотрении алгоритма оптимального упорядочивания необходимо иметь в виду следующие обстоятельства. Для сохранения структурной симметрии при оптимальном упорядочивании необходимо осуществлять перестановку как строк, так столбцов. Поскольку перестановка столбцов приводит к перестановке вектора переменных, то информацию о выполненных перестановках необходимо обязательно сохранить до конца расчета. Помимо решения задачи оптимального упорядочивания строк и столбцов, исходя из критерия минимизации появления новых ненулевых элементов при выборе ведущего элемента, следует так же обеспечивать неравенство нулю этого элемента. Очевидно, что только такое условия позволяет выполнить пересчет элементов при LU -факторизации. Вместе с тем, при описании систем в модифицированном расширенном базисе узловых потенциалов [19, 20], неравные нулю значения диагональных коэффициентов матрицы обеспечиваются только верхним диагональным блоком матрицы W . Что же касается нижнего блока, то в общем случае, его диагональные элементы могут иметь нулевые значения, и, следовательно, выбор их в качестве ведущих недопустим.

Чтобы устранить указанное препятствие, следует соответствующим образом организовать работу оптимального упорядочивания строк и столбцов в процессе LU -факторизации. С этой целью сначала следует провести перестановки строк и столбцов индексы, которые не превышают значение nv , а затем осуществить процедуру упорядочивания строк и столбцов, изменяя индексы в пределах от $nv+1$ до n . Если следовать указанному алгоритму, то на первом этапе оптимального упорядочивания в процессе пересчета элементов нижней блочной строки диагонального блока, все ее диагональные элементы приобретут ненулевые значения и дальнейшие преобразования второго шага окажутся допустимыми. Вся информация о выполняемых перестановках строк и столбцов в процессе их оптимального упорядочивания записывается временно в нулевую строку массива C .

На третьем шаге топологического этапа на основе топологического портрета схемы, отображенного в матрице C , осуществляется построение координатных матриц WJI , ERC , SI . Компактная матрица WJI содержит индексы столбцов исходной полной матрицы в ее наддиагональной части. Вследствие структурной симметрии исходной матрицы, компактная матрица WJI одновременно содержит так же и индексы строк ненулевых элементов поддиагональной части исходной матрицы.

Относительный адрес aw индекса j столбца в WJI , может быть найден путем последовательного сканирования строк индексной матрицы C с подсчетом текущего суммарного значения числа aw элементов $C_{ij} = 1$, расположенных выше

диагонали. В массиве ERC хранятся относительные адреса входа в строку. Таким образом, формирование координатных матриц ERC , WJI осуществляется в одном цикле путем нахождения относительного адреса aw .

Необходимо также формировать массив SI , содержание которого отображает структуру нулевого столбца исходной матрицы. Относительный адрес элемента этого массива as подсчитывается так же путем вычисления числа ненулевых элементов в этом столбце. После выполнения всех перечисленных шагов информация о структуре индексной матрицы C с учетом появления новых ненулевых элементов будут содержаться в координатных матрицах и, следовательно, матрицу C можно удалить из памяти.

На численном этапе (шаги 3-7) в соответствии с уже известным форматом компактного описания, можно открыть и сформировать массив для численных матриц WD , WU , WL и осуществить численную LU-факторизацию. При работе с компактным описанием исходная разреженная матрица W в описании не существует и LU-факторизация должна выполняться виртуально путем обработки компактных массивов WD , WU , WL .

Поэтому для этой цели необходимо построить виртуальный алгоритм LU-факторизации, который должен установить взаимосвязь элементов полной матрицы w_{kk} , w_{kj} , w_{jk} , w_{ij} с относительными адресами a , определяющими местоположение этих элементов в компактных массивах WD , WU , WL .

При этом следует иметь в виду, что элемент w_{kk} расположен на диагонали полной матрицы и, следовательно, может быть выбран непосредственно из массива WD по относительному адресу $a=k$, т.е. $w_{kk}=WD[k]$. Для выбора элементов w_{kj} и w_{jk} следует воспользоваться координатными массивами ERC и WJI и выполнить сканирование k -ой строки и соответственно k -го столбца.

После проведения численной LU-факторизации формируется компактный массив задающих источников SZ и проводится расчет переменных путем выполнения прямого и обратного хода. Поскольку исходная матрица в описании отсутствует, то прямой и обратный ход необходимо выполнять, используя компактное описание на основе алгоритмов виртуального типа, которые основаны на использовании компактных массивов для обработки информации о задающих источниках s_i на топологическом этапе (массив SI) и на численном этапе (массив SZ).

Заключение. Наибольшую практическую значимость при создании высокопроизводительных систем автоматизированного проектирования представляют метод индексно-адресных матриц, который позволяет наиболее простым путем реализовать построение программного обеспечения системы моделирования, и метод структурно-симметричных матриц, который выгодно отличается от других методов своей высокой эффективностью. Для возможности коррекции формата в методе структурно-симметричных матриц следует использовать двухэтапную процедуру, основанную на предварительном построении топологического портрета моделируемой схемы с последующим численным расчетом схемы в реальном формате.

Практическая реализация метода адресации и метода структурно-симметричных матриц на основе двухэтапного процесса сжатия данных позволяет существенно повысить эффективность работы систем автоматизированного проектирования.

Исследование выполнено при поддержке программы фундаментальных исследований президиума РАН «Исследование инновационных методов автоматизации проектирования СБИС и систем на кристалле» в рамках проекта 0071-2018-0006.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Писсанецки С.* Технология разреженных матриц: пер. с англ. – М.: Мир, 1988. – 406 с.
2. *Тьюарсон Ф.Р.* Разреженные матрицы: пер. с англ. – М.: Мир, 1977. – 189 с.
3. *Эстербю О., Златев З.* Прямые методы для разреженных матриц: пер. с англ. – М.: Мир, 1987. – 118 с.
4. *Норенков И.П.* Введение в автоматизированное проектирование технических устройств и систем. – М.: Высшая школа, 1986. – 304 с.
5. *Баталов Б.В., Егоров Ю.Б., Русаков С.Г.* Основы математического моделирования больших интегральных схем на ЭВМ. – М.: Радио и связь, 1982. – 168 с.
6. *Норенков И.П., Маничев В.Б.* Основы теории и проектирования САПР. – М.: Высшая школа, 1990. – 335 с.
7. *Глориозов Е.А., Ссорин В.Г., Сытчук П.П.* Введение в автоматизацию схемотехнического проектирования. – М.: Советское радио, 1976. – 224 с.
8. *Кнут Д.* Искусство программирования для ЭВМ: пер. с англ. Т. 1. – М.: Мир, 1976. – 734 с.
9. *Ноутон П., Шилдт Г.* Java 2: пер. с англ. – СПб.: Изд-во «БХВ-Петербург» 2001. – 1072 с.
10. *Дейтел Х.М., Дейтел П.Дж.* Как программировать на Java: пер. с англ. – М.: Бинном Пресс, 2003. – 848 с.
11. *Троелсон Э.* Язык программирования C# 2005 и платформа .NET 2.0: пер. с англ. – М.: Изд-во «Вильямс», 2007. – 1167 с.
12. *Шилдт Г.* Полный справочник по C#: пер. с англ. – М.: Изд-во «Вильямс», 2006. – 752 с.
13. *Мэтью Мак-Дональд, Шпушта М.* Microsoft ASP.NET 2.0 с примерами на C# 2005 для профессионалов: пер. с англ. – М.: Изд-во «Вильямс», 2006. – 1408 с.
14. *Анисимов В.И., Амахер Ю.М.* Компактные методы обработки разреженных матриц задач мониторинга на основе списочных структур // Труды Пятой Международной конференции «Приборостроение в экологии и безопасности человека». – СПб.: ГУАП, 2007. – 316 с.
15. *Анисимов В.И., Дмитриевич Г.Д., Скобельцын К.Б. и др.* Диалоговые системы схемотехнического проектирования. – М.: Радио и связь, 1988. – 288 с.
16. *Гридин В.Н., Рыжов Н.Г., Анисимов В.И., Абухазим М.М.* Методы повышения эффективности процессов моделирования динамических режимов нелинейных систем // Информационные технологии. – 2017. – № 11. – С. 796-802.
17. *Гридин В.Н., Анисимов В.И., Абухазим М.М.* Методы моделирования систем на основе методов декомпозиции и компактной обработки разреженных матриц // Информационные технологии в проектировании и производстве. – 2016. – № 1. – С. 3-8.
18. *Гридин В.Н., Анисимов В.И., Абухазим М.М.* Сжатие данных в системах автоматизации схемотехнического проектирования на основе методов фиксированного формата // Системы высокой доступности. – 2016. – № 4. – С. 34-40.
19. *Сигорский В.П., Петренко А.И.* Алгоритмы анализа электронных схем. – М.: Советское радио, 1976. – 608 с.
20. *Влах И., Сингхал К.* Машинные методы анализа и проектирования электронных схем: пер. с англ. – М.: Радио и связь, 1988. – 560 с.

REFERENCES

1. *Pissanetski C.* Tekhnologiya razrezhennykh matrity [Sparse matrix technology]: transl. from engl. Moscow: Mir, 1988, 406 p.
2. *T'uarson F.R.* Razrezhennye matrity [Sparse matrices]: transl. from engl. Moscow: Mir, 1977, 189 p.
3. *Esterbyu O., Zlatev Z.* Pryamye metody dlya razrezhennykh matrity [Direct methods for sparse matrices]: transl. from engl. Moscow: Mir, 1987, 118 p.
4. *Norenkov I.P.* Vvedenie v avtomatizirovannoe proektirovanie tekhnicheskikh ustroystv i sistem [Introduction to computer-aided design of technical devices and systems]. Moscow: Vysshaya shkola, 1986, 304 p.
5. *Batalov B.V., Egorov YU.B., Rusakov S.G.* Osnovy matematicheskogo modelirovaniya bol'shikh integral'nykh skhem na EVM [Fundamentals of mathematical modeling of large integrated circuits on a computer]. Moscow: Radio i svyaz', 1982, 168 p.

6. *Norenkov I.P., Manichev V.B.* Osnovy teorii i proektirovaniya SAPR [Fundamentals of CAD theory and design]. Moscow: Vysshaya shkola, 1990, 335 p.
7. *Gloriozov E.A., Ssorin V.G., Sypchuk P.P.* Vvedenie v avtomatizatsiyu skhemotekhnicheskogo proektirovaniya [Introduction to circuit design automation]. Moscow: Sovetskoe radio. 1976, 224 p.
8. *Knut D.* Iskusstvo programmirovaniya dlya EVM [The art of computer programming]: transl. from engl. Vol. 1. Moscow: Mir, 1976, 734 p.
9. *Nouton P., SHildt G.* Java 2: transl. from engl. Saint Petersburg: Izd-vo «BKHV-Peterburg» 2001, 1072 p.
10. *Deytel Kh.M., Deytel P.Dzh.* Kak programmirovat' na Java [How to program in Java]: transl. from engl. Moscow: Binom Press, 2003, 848 p.
11. *Troelson E.* YAzyk programmirovaniya C# 2005 i platforma .NET 2.0 [C# 2005 programming language and .NET 2.0 platform]: transl. from engl. Moscow: Izd-vo «Vil'yams», 2007, 1167 p.
12. *SHildt G.* Polnyy spravochnik po C# [A complete reference for C#]: transl. from engl. Moscow: Izd-vo «Vil'yams», 2006, 752 p.
13. *Met'yu Mak-Donal'd, SHpushta M.* Microsoft ASP.NET 2.0 s primerami na C# 2005 dlya professionalov [Microsoft ASP.NET 2.0 with examples on C# 2005 for professionals]: transl. from engl. Moscow: Izd-vo «Vil'yams», 2006, 1408 p.
14. *Anisimov V.I., Amakhvyr YU.M.* Kompaktnye metody obrabotki razrezhennykh matrits zadach monitoringa na osnove spisochnykh struktur [Compact methods of processing of sparse matrices of monitoring tasks on the basis of list structures], Trudy Pyatoy Mezhdunarodnoy konferentsii «Priborostroenie v ekologii i bezopasnosti cheloveka» [Proceedings of the Fifth International conference "Instrument-making in ecology and human security"]. Saint Petersburg: GUAP, 2007, 316 p.
15. *Anisimov V.I., Dmitrevich G.D., Skobel'tsyn K.B. i dr.* Dialogovye sistemy skhemotekhnicheskogo proektirovaniya [The system circuit design]. Moscow: Radio i svyaz', 1988, 288 p.
16. *Gridin V.N., Ryzhov N.G., Anisimov V.I., Abukhazim M.M.* Metody povysheniya effektivnosti protsessov modelirovaniya dinamicheskikh rezhimov nelineynykh sistem [Methods of improving the efficiency of simulation processes of dynamic modes of nonlinear systems], *Informatsionnye tekhnologii* [Information technology], 2017, No. 11, pp. 796-802.
17. *Gridin V.N., Anisimov V.I., Abukhazim M.M.* Metody modelirovaniya sistem na osnove metodov dekompozitsii i kompaktной obrabotki razrezhennykh matrits [Methods of modeling systems based on decomposition methods and compact processing of sparse matrices], *Informatsionnye tekhnologii v proektirovanii i proizvodstve* [Information technologies in design and production], 2016, No. 1, pp. 3-8.
18. *Gridin V.N., Anisimov V.I., Abukhazim M.M.* Szhatie dannykh v sistemakh avtomatizatsii skhemotekhnicheskogo proektirovaniya na osnove metodov fiksirovannogo formata [Data compression in circuit design automation systems based on fixed format methods], *Sistemy vysokoy dostupnosti* [High availability systems], 2016, No. 4, pp. 34-40.
19. *Sigorskiy V.P., Petrenko A.I.* Algoritmy analiza elektronnykh skhem [Algorithms for analysis of electronic circuits]. Moscow: Sovetskoe radio, 1976, 608 p.
20. *Vlakh I., Singkhal K.* Mashinnye metody analiza i proektirovaniya elektronnykh skhem [Machine methods of analysis and design of electronic circuits]: transl. from engl. Moscow: Radio i svyaz', 1988, 560 p.

Статью рекомендовал к опубликованию д.ф.-м.н., профессор В.А. Черепенин.

Гридин Владимир Николаевич – Центр информационных технологий в проектировании РАН; e-mail: info2@ditc.ras.ru; г. Одинцово, ул. Маршала Бирюзова, 7а; тел.: 84955960219; научный руководитель; д.т.н.; профессор.

Анисимов Владимир Иванович – e-mail: vianisimov@inbox.ru; тел.: 89117448228; г.н.с.; д.т.н.; профессор.

Абухазим Монзер Мохаммед – Санкт-Петербургский Государственный Электротехнический университет; e-mail: Abuhazim_monther@Yahoo.com; г. Санкт-Петербург, ул. С. Корзуна 20, кв. 5; тел.: 88122343675; аспирант.

Gridin Vladimir Nikolaevich – Design information technologies Center Russian Academy of Sciences; e-mail: info2@ditc.ras.ru; Odintsovo, Marshala Biryuzova, 7a; phone: +74955960219; scientific director; dr. of eng. sc.; professor.

Anisimov Vladimir Ivanovich – e-mail: vianisimov@inbox.ru; phone: +79117448228; chief scientific officer; dr. of eng. sc.; professor.

Abukhazim Monzer Mohammed – St. Petersburg State Electrotechnical University; e-mail Abukhazim_monther@Yahoo.com; St. Petersburg, st. S. Korzun, 20 sq. M. 5; phone: +78122343675; post-graduate student.

УДК 004.94

DOI 10.23683/2311-3103-2018-4-38-47

В.И. Анисимов, С.А. Васильев, В.Н. Гридин**ВЫСОКОСКОРОСТНОЙ ПОЛНОДУПЛЕКСНЫЙ МЕТОД ОБМЕНА ДАННЫМИ ДЛЯ РАСПРЕДЕЛЕННЫХ САПР***

Указываются принципы построения ИТ-инфраструктуры систем автоматизированного проектирования (САПР). Описываются преимущества использования сервис-ориентированной архитектуры (SOA) для создания информационно-программных САПР. Обозначается возможная платформа Windows Communication Foundation (WCF) для реализации ИТ-инфраструктуры САПР на основе SOA и полнодуплексного протокола обмена данными WebSocket. Приводятся исторические аспекты становления технологий WCF, SOA, WebSocket. Определяются преимущества использования платформы WCF для создания распределенной САПР в сравнении со стандартной концепцией построения службы. Показаны методы создания коммуникационных модулей клиентской и серверной составляющих для сервисов, использующих платформу WCF. В статье предоставляется код поведения контракта серверной составляющей коммуникационного модуля. Указываются различные параметры настроек класса привязок для использования асинхронных методов обмена информацией и обеспечения связующих служебных функций. В статье приводятся методы работы с WSDL документом и рекомендации транспортной функции протоколов ws:// и http://. Описываются возможные специфичные параметры транспорта для обмена сообщениями. Приводится краткий обзор особенностей для коммуникации в гетерогенных средах. Обозначаются критические случаи взаимодействия кроссплатформенного программного обеспечения. Демонстрируются результаты использования коммуникационных модулей распределенной САПР на примере консольных отчетов. Суммируются преимущества использования всех трех вышеописанных технологий в едином программном обеспечении возможно для использования при построении САПР. Оказывается внимание стандартизированным методам взаимодействия с описанными коммуникационными модулями.

САПР; информационная система; распределенные системы; асинхронный обмен; коммуникационные модули; SOA; WebSocket; WCF.

V.I. Anisimov, S.A. Vasilev, V.N. Gridin**HIGH-SPEED FULL-DUPLEX METHOD OF DATA EXCHANGE FOR DISTRIBUTED CAD**

The principles of IT infrastructure creation of automated design engineering systems (CAD) are specified. Advantages of using the service oriented architecture (SOA) for creation of an information and program CAD are described. The possible Windows platform of Communication Foundation (WCF) for implementation of CAD IT infrastructure on the basis of SOA and the full-duplex exchange protocol is designated by data of WebSocket. Historical aspects of formation of WCF, SOA and WebSocket technologies are given. Advantages of using the WCF platform for creation of the

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-07-00082а.