

УДК 004.421

Жаныс А.Б., доктор философии PhD, и.о. профессора Кокшетауского университета им. А.Мырзахметова
Надырова Ф.К., преподаватель Кокшетауского университета им. А.Мырзахметова

ОБОСНОВАНИЕ НЕОБХОДИМОСТИ РАЗРАБОТКИ НОВОГО АЛГОРИТМА ФОРМИРОВАНИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ ИС

Аннотация: Предметом исследования в данной работе являются существующие математические и алгоритмические методы формирования и решения математических моделей ИС. Проведённый анализ позволил выявить преимущества и недостатки существующих методов, и сформулировать требования к алгоритму компиляционного формирования математических моделей. По результатам анализа были предложены новый алгоритм формирования математической модели ИС, позволяющий ускорить процесс формирования математической модели в виде системы линейных алгебраических уравнений (СЛАУ), а также алгоритм получения аналитических выражений для вычисления корня СЛАУ, записанных в явном виде, позволившие ускорить процесс решения математической модели. Актуальность. С развитием технологий в областях проектирования и производства ИС, происходит постоянное усложнение как самих проектируемых устройств и их отдельных компонентов, так и программных средств, позволяющих проводить их моделирование. В частности, повышение точности производства отдельных компонентов ИС привело к необходимости более полного и более точного описания физических характеристик схмотехнических моделей элементов и протекающих в них процессов. В задачах моделирования ИС подобные требования привели к усложнению описания программных моделей элементов, что, в свою очередь, послужило причиной увеличения времени машинного расчёта характеристик ИС.

Ключевые слова: машинное формирование математической модели ИС, численные значения матрицы, узловых проводимостей, вектора, теория языков, программирование, алгоритма компиляционного формирования математических моделей ИС, функционирования моделей полупроводниковых элементов, моделирование схем,

Abstract: The subject of the research in this paper are the existing mathematical and algorithmic methods of forming and solving mathematical models of IP. The analysis revealed the advantages and disadvantages of the existing methods, and to formulate the requirements for the compilation algorithm of formation of mathematical models. According to the analysis it has been proposed a new algorithm for the formation of a mathematical model of the IP, which allows to speed up the process of forming a mathematical model of a system of linear algebraic equations (Slough), as well as the ability to obtain analytical expressions for calculating the root SLAE recorded explicitly, will speed up the process of solving a mathematical model. Relevance. With the development of technology in the design and manufacture of ICs, there is the increasing complexity of both the designed devices and their individual components, and software tools that enable them to carry out simulations. In particular, increased accuracy of individual components of IC production has led to the need for more complete and more accurate description of the physical characteristics of circuit elements and patterns occurring in them. The problems of modeling of IP such requirements have led to the complication of the description of software models of elements that, in turn, caused the increase of the calculation time machine performance ICs.

Keywords: machine forming a mathematical model of the IP, the numerical values of the matrix, nodal conduction, the vector theory of languages, programming, algorithm forming the compilation of mathematical models of IP, functioning models of semiconductor devices, circuit simulation.

В ходе выполнения машинного формирования математической модели ИС происходит заполнение численными значениями матрицы узловых проводимостей Y и вектора источников тока J выражения (1)

$$Y \cdot \vec{\varphi} = \vec{J} \quad (1)$$

Процесс заполнения значениями матрицы проводимостей Y и вектора тока J в простейшем варианте реализации происходит как показано в листинге 1. Производиться последовательный опрос программных моделей всех компонентов схемы, которые должны вносить вклад в матрицу Y и вектора I , во время которого для этих элементов определяются значение их проводимостей и токов в зависимости от текущих значений потенциалов в узлах схем. Программное представление матрицы проводимостей как элемента математической модели в виде непосредственно матрицы в памяти с точки зрения теории языков программирования на практике никогда не встречается.

```
for (i = 0; i < schemeelements, count = i + 1){
    scheme elements [i]. get_y (Y_matrix);
    scheme elements [i]. get_i (I_vector);
}
```

Листинг 1. Пример кода, иллюстрирующего процесса машинного формирования математической модели.

В первую очередь это связано с сильной разреженностью матрицы, то есть было показано, что для эффективного хранения разреженных матриц разработан ряд алгоритмов, которые позволяют компактно размещать в памяти разреженные матрицы больших размерностей. Тем не менее, рассмотренные форматы хранения данных обладают некоторой избыточностью связанной с необходимостью, помимо непосредственных ненулевых значений хранить в памяти еще и информацию об их положении в реальной матрице.

Приблизительно оценить степень избыточности для рассмотренного формата хранения A_{ij} можно следующим образом.

Определим число ненулевых элементов матрицы как $N_{NonZero}$ размер одного элемента матрицы в байтах как D_{Size} , количество байт, необходимое для хранения информации о положении одного элемента матрицы в столбце или строке, как P_{Size} . Тогда эффективность применения формата для хранения разреженной матрицы можно определить следующим выражением (2):

$$Eff = \frac{N_{NonZero} \cdot D_{size} + N_{NoneZero} \cdot P_{size} + N_{NonZero} \cdot M_{Dim}}{N_{NonZero} \cdot D_{size}}, \quad (2)$$

Рисунок 1. Предлагаемый алгоритм компиляционного формирования математических моделей ИС.

Для иллюстрации разработанного алгоритма компиляционного формирования математических моделей ИС рассмотрим процесс аналитического формирования математической модели на примере схемы, изображенной на рисунке 1.1.

В схеме присутствуют два линейных резистора, один линейный конденсатор, источник импульсного напряжения и идеальный диод.

Проведём аналитическое составление компонентов математической модели при условии использования МУП. В качестве метода интегрирования будем использовать неявный метод Эйлера. Для простоты также будем использовать в качестве формулы для диода формула для прямого графика.

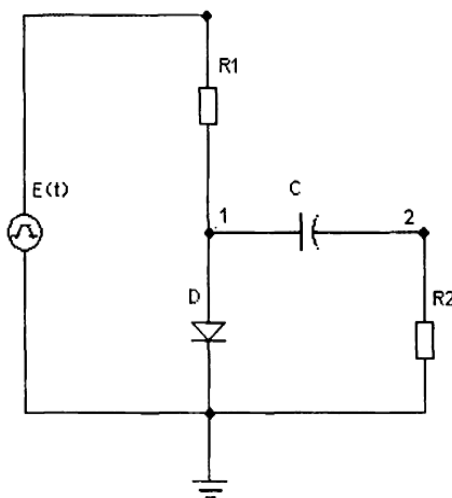


Рисунок 1.1. Тестовая схема.

Это допущение не является принципиальным, в то же время позволяет упростить запись и понимание идеи алгоритма компиляционного формирования.

Компоненты вектора токов будут записаны в виде (3), а компоненты матрицы проводимостей в виде (4).

$$\begin{cases} J_1 = \frac{E(t) - \varphi_1}{R_1} - I_0 \left(e^{\frac{\varphi_1}{m\varphi_t}} - 1 \right) - \frac{C}{h} (\Delta\varphi_1 - \Delta\varphi_2), \\ J_2 = \frac{C}{h} (\Delta\varphi_1 - \Delta\varphi_2) - \frac{\varphi_2}{R_2}, \end{cases} \quad (3)$$

$$\left\{ \begin{array}{l} Y_{11} = -\frac{1}{R_1} - \frac{I_0}{m\varphi_t} e^{\frac{\varphi_1}{m\varphi_t}} - \frac{C}{h}, \\ Y_{12} = \frac{C}{h}, \\ Y_{21} = \frac{C}{h}, \\ Y_{22} = -\frac{C}{h} - \frac{1}{R_2}. \end{array} \right. \quad (4)$$

В этих выражениях J_k - k-ный компонент вектора токов, Y_{ij} - компонентом i-той строки и j-того столбца матрицы узловых проводимостей. φ_1 и φ_2 - потенциалы в узлах 1 и 2 схемы, h – величины шага метода интегрирования.

В случае программой реализации выражение (3) и (4), говоря о затратах оперативной памяти на хранение компонентов математической модели очевидно, что запись выражений в таком виде и будет наиболее эффективной с точки экономии памяти поскольку матрицы в явном виде нет, необходимости разграничивать нулевые и ненулевые элементы также не наблюдается. Храниться будут только те компоненты, для которых сформированы выражения (3) и (4).

Идея алгоритма компиляционного формирования математической модели ИС состоит в том, чтобы для моделируемой схемы сформировать программный код на высокоуровневом языке программирования, который бы описывал математическую модель схемы в виде выражения (3) и (4).

В соответствии с приведённой формулой (1), эффективность хранения компонентов математической Eff для данного алгоритма будет составлять 1. Поскольку в аналитических выражениях (3) и (4) участвуют только ненулевые элементы компонентов математической модели, затраты на хранение информации о столбце/строке каждого из ненулевых элементов будут отсутствовать. Таким образом, величины P_{Size} , M_{Dim} и D_{Size} выражения (1) будут равны нулю.

Помимо сокращения используемой оперативной памяти, предлагаемый алгоритм формирования математической модели позволит существенно ускорить процесс моделирования.

В процессе компиляции выражений, описывающих компоненты математической модели, некоторые выражения могут быть вычислены автоматически. Например, в выражении (4) в пределах одного шага интегрирования вычисляемые величины Y_{12} , Y_{21} и Y_{22} будут постоянными, поэтому вычислить их можно будет всего 1 раз. В случае использования метода интегрирования с постоянным шагом эти величины будут постоянными на протяжении всего сеанса моделирования.

В общем случае можно сделать вывод, что экономия ресурсов будет наблюдаться для линейных сопротивлений в течение всего цикла моделирования, для линейных конденсаторов и индуктивностей в пределах одного шага моделирования.

Значительное сокращение и времени моделирования, и используемой оперативной памяти может быть достигнуто за счёт учёта одинаковых по номиналу элементов. Конечно, разработаны и применяются алгоритмы учёта одинаковых элементов, позволяющие в определённых ситуациях обнаруживать вхождение одинаковых по физическим параметрам элементов и рассчитывать их характеристики только один раз за требуемый интервал времени.

Разработанный алгоритм позволит учесть одинаковые элементы ещё на этапе генерации программного кода, тем самым сократит время работы симулятора в процессе моделирования.

Несмотря на преимущества, получаемые в результате использования компиляционного алгоритма формирования математической модели ИС, механизм использования данного алгоритма не является очевидным. Для проектировщика не представляется возможным проводить аналитическое формирование системы уравнений, описывающей математическую модель каждой моделируемой схемы, как было показано на примере систем (3) и (4), и выполнять написание программы на языке программирования высокого уровня, реализующей моделирование каждой конкретно взятой схемы.

Для использования преимуществ, обеспечиваемых формированием выражений математической модели схемы в явном виде, следует перестроить алгоритмы работы симулятора.

Предлагаемый алгоритм работы программы моделирования ИС.

Основной задачей при разработке программ-симуляторов, поддерживающих возможность компиляционного формирования математической модели ИС, является задача разработки алгоритма, позволяющего получить описание математической модели схемы в явном виде на языке программирования высокого уровня. Описание математической модели схемы формируется из описаний математических моделей отдельно взятых компонентов.

Полученный программный код дополняется участками кода различного назначения. Примерами таких участков кода могут служить: код, описывающий процесс решения математической модели, сформированной с использованием предлагаемого алгоритма, код, отвечающий за сохранение результатов моделирования в требуемом формате, код для инициализации очистки памяти. Поскольку для разных схем формируются различные математические модели, получаемый код программы на языке программирования высокого уровня соответствует только одной, моделируемой в текущий момент схеме. После формирования исходного кода программы должны выполняться компиляция программы и запуск скомпилированной программы на исполнение. Схема работы симулятора представлена на рисунке 2.

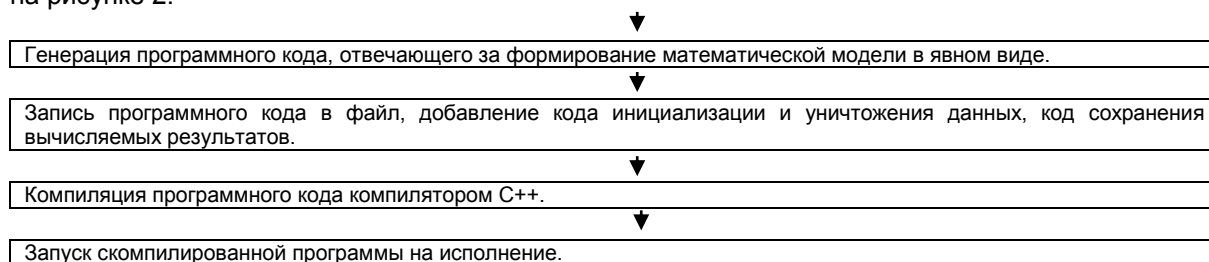


Рисунок 2. Предлагаемая последовательность этапов работы программы моделирования.

Анализ преимуществ, обусловленных применением алгоритма компиляционного формирования математических моделей ИС. Получаемые благодаря применению алгоритма формирования в явном виде выражений для записи математической модели преимущества позволяют добиться появления трёх важных преимуществ:

1. сокращение времени на формирование математической модели ИС, которое достигается за счет отказа от необходимости последовательного обхода программных моделей элементов для получения требуемых характеристик и за счет возможности учёта одинаковых по своим характеристикам элементов;

2. сокращение времени на вычисление значений физических характеристик (то есть вкладов в вектор токов и матрицу проводимостей) большинства линейных элементов как минимум в пределах одного шага интегрирования;

3. сокращение затрат оперативной памяти при хранении математической модели.

Получение данных преимуществ является целью модификаций алгоритмов формирования математической модели ИС, реализуемых в большинстве существующих на сегодняшний день программах симуляторов.

В таблице 1 приводится сводная характеристика рассмотренных методов ускорения формирования математической модели.

Таблица 1.

Краткое сравнение рассмотренных методов.

Метод	Достоинства	Недостатки
Учёт проводимостей линейных элементов	Повышение скорости формирования математической модели за счёт уменьшения количества обращений к программным моделям линейных элементов.	Даёт преимущество только для схем, содержащих большое число линейных компонентов. Дополнительные затраты памяти, связанные с необходимостью хранить в памяти две матрицы проводимостей.
Технологии хранения и обработки разреженных матриц	Уменьшение затрат используемой оперативной памяти, которое достигается благодаря отсутствию необходимости хранить нулевые элементы матрицы проводимостей.	Очевидное усложнение процесса формирования и решения математической модели.

Секция с названием «Генерация дополнительно программного кода» включает в себя генерацию программного кода для решения нижеперечисленных задач.

1. Выделение и освобождение памяти, инициализация и очистка данных.
2. Реализация требуемого вида анализа, методов интегрирования и линеаризации.
3. Реализация решения сформированной математической модели.
4. Сохранение данных в требуемом формате.
5. Обеспечение функционирования моделей полупроводниковых элементов.
6. Вывод информационных сообщений в коде моделирования.

Таблица 2.

Сравнение предлагаемого алгоритма и алгоритма учёта проводимостей линейных элементов.

Анализируемые характеристики	Алгоритм учёта проводимостей нелинейных элементов	Предлагаемый алгоритм
Преимущество: Повышение скорости формирования математической модели.	Наблюдается благодаря: сокращению времени вычисления проводимостей линейных элементов.	Наблюдается, благодаря тому, что: 1. для линейных элементов значения вычисляются ещё на этапе компиляции, в процессе расчёта время не затрачивается. 2. выражения математической модели "записаны в явном виде, поэтому не требуется тратить время на опрос моделей элементов.
Недостаток: Дополнительные затраты оперативной памяти.	Наблюдаются, поскольку алгоритм предполагает хранение двух матриц: общей матрицы проводимостей схемы и матрицы проводимостей линейных элементов.	Не наблюдаются, поскольку сам алгоритм не требует наличия дополнительной матрицы.

В таблицах 2 и 3 приводится сравнение данных методов с предлагаемым в диссертационной работе методом компиляционного формирования математической модели ИС.

Очевидно, что при сохранении скорости формирования математической модели случае применения предлагаемого алгоритма по сравнению с алгоритмом учёта проводимостей линейных элементов значительный выигрыш достигается благодаря снижению затрат используемой оперативной памяти.

Таблица 3.

Сравнение предлагаемого алгоритма и технологий хранения и обработки разреженных матриц

Анализируемые характеристики	Технологии хранения и обработки разреженных матриц	Предлагаемый алгоритм
Преимущество: Значительная экономия оперативной памяти на хранение разреженной матрицы.	Наблюдается, благодаря специальному способу записи матрицы.	Наблюдается, благодаря тому, что формируемые выражения содержат только те значения, которые формируют ненулевые элементы матрицы проводимостей.
Недостаток: Избыточность хранимой информации.	Наблюдаются, благодаря тому, что помимо непосредственно значений матрицы требуется хранить служебную информацию.	Не наблюдаются благодаря тому, что результирующий файл с программным кодом для формирования математической модели попадают только те значения матрицы проводимостей, которые не являются нулевыми.

Благодаря тому, что предлагаемый алгоритм не предполагает какой-либо специальной организации данных для хранения математической модели, он является более эффективным с точки зрения затрат оперативной памяти.

Области применения предлагаемого алгоритма.

Выше было показано, что применение предлагаемого алгоритма для схем любых классов и любых размеров даст преимущество по скорости формирования математической модели и сопровождающим этот процесс затратам оперативной памяти.

Существует ряд задач, в которых применение разработанного алгоритма даст наибольший

положительный эффект, однако отдельно следует упомянуть два из них: моделирование схем, полученных после экстракции из топологии, содержащих большое число RC – цепочек, и моделирование схем в ходе выполнения задачи характеристики библиотечных элементов. Эти две задачи в настоящее время являются одними из самых ресурсоемких по объёмам потребляемой памяти и по скорости выполнения расчетов, поэтому имеет смысл рассмотреть их подробнее.

Применение в задаче характеристики библиотечных элементов.

Задача характеристики библиотечных элементов – одна из самых низкоэффективных задач с точки зрения отношения полезного времени работы программы моделирования к общему времени работы [49,50]. Ее суть состоит в многократном моделировании одних и тех же схем, как правило, небольшого размера, с различными последовательностями входных воздействий.

Полное время работы симулятора, выполняющего моделирование ИС, включает в себя три показателя: время чтения лицензии и подготовки к запуску T_{lic} , время чтения текстового описания схемы, формирования и заполнения внутренних структур данных T_{read} и непосредственно время расчетов и сохранения результатов T_{sim} .

Выражение (5) показывает взаимосвязь этих показателей:

$$T_{\Sigma} = T_{lic} + T_{read} + T_{sim}, \quad (5)$$

Здесь T_{Σ} показывает полное время моделирования.

Для оценки эффективности работы программ моделирования введём следующее понятие - время эффективной работы. Оно оценивается выражением (6):

$$T_{eff} = \frac{T_{\Sigma}}{T_{sim}}. \quad (6)$$

Это выражение показывает, насколько общее время моделирования больше времени непосредственно процесса расчёта, включающего этапы формирования и решения математической модели, а также сохранения полученных результатов.

Очевидно, что с увеличением размера схем эта величина будет стремиться к 1. Чем она ближе к 1, тем эффективнее будет расходоваться время работы симулятора.

При моделировании схем в процессе выполнения задачи характеристики библиотечных элементов выполняется многократный запуск программ моделирования. При этом также многократно затрачиваются времена T_{lic} и T_{read} , что является не совсем эффективным распределением времени.

Оптимальным механизмом проведения характеристики был бы такой алгоритм, который выстроил бы моделирование схемы в последовательность:

1. запуск программы, проверка лицензии на корректность использования программы моделирования;
2. чтение описания схемы, формирование внутренних данных и их инициализация;
3. анализ качества и количества комбинаций входных воздействий;
4. загрузка последовательностей входных воздействий во внутренние структуры данных;
5. начало моделирования.

Именно такой механизм может быть воспроизведён с применением предлагаемого в диссертации алгоритма компиляционного моделирования. Проводя предварительный анализ схемы, в итоговый программный код помещается не только описание процесса формирования математической модели и её решения, но и задаётся последовательность подачи входных воздействий.

Тогда, при выполнении характеристики библиотечного элемента, требующей последовательного N-кратного запуска с различными входными параметрами, для оценки эффективности работы симулятора с применением традиционного подхода будет получено выражение (7). В то же время с применением разработанного алгоритма будет получено выражение (8).

$$T_{standart_{\Sigma}}^N = N(T_{lic} + T_{read} + T_{sim}),$$

$$T_{standart_{eff}}^N = \frac{T_{standart_{\Sigma}}^N}{N \cdot T_{sim}} \quad (7)$$

$$T_{compile_{\Sigma}}^N = T_{lic} + T_{read} + T_{sim},$$

$$T_{compile_{eff}}^N = \frac{T_{compile_{\Sigma}}^N}{N \cdot T_{sim}} \quad (8)$$

Для оценки положительного эффекта в виде сокращения времени моделирования в результате применения алгоритма компиляционного моделирования следует обратиться к рисунку 3.

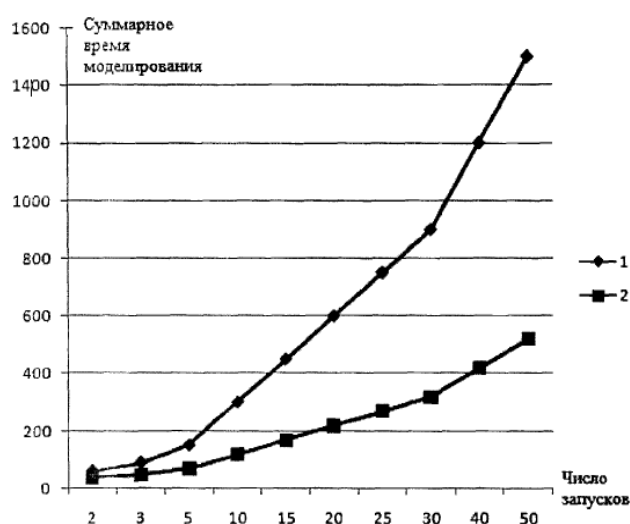


Рисунок 3. Оценка эффективности разработанного алгоритма для задачи характеристики.

На представленном рисунке кривая 1 иллюстрирует время работы в случае использования одного из традиционных симуляторов, кривая 2 показывает время работы программы с применением компиляционного алгоритма формирования математической модели. В качестве значений по оси абсцисс указано число запусков программы с различными входными воздействиями, по оси ординат — время работы в условных единицах.

При построении графика считалось, что от запуска к запуску программы время получения лицензии и время чтения текстового описания схемы оставалось неизменным.

Применение предложенного алгоритма в задаче моделирования схем с паразитными элементами.

Моделирование схем с паразитными элементами является самой сложной задачей с точки зрения требований к объёму оперативной памяти [5]. Связано это с тем, что при моделировании таких схем сильно возрастает размерность формируемой системы уравнений, и, следовательно, значительно увеличивается время её решения.

Этап экстракции паразитных элементов, представляемых в виде последовательностей RC-цепочек, является единственным механизмом на этапе схемотехнического проектирования для моделирования не только физических свойств отдельных компонентов ИС, но и физических аспектов распространения сигналов в межсоединениях.

В результате процедуры экстракции, каждое межсоединение в проектируемой схеме сопровождается наличием одной или нескольких пар элементов R и C, которые имитируют задержку в - разрядов конденсатора и падение уровня сигнала при его передаче на расстояния распространения сигнала через задержку зарядов через падение напряжения на резисторе.

В предыдущей главе было показано, что применение форматов хранения разреженных

матриц может существенно снизить объём занимаемой на хранение математической модели оперативной памяти, но, в то же время, он является сильно избыточным. Формат хранения данных, организуемый благодаря применению разработанного алгоритма, будет наименее требовательным к памяти.

Ещё одним фактором, который может повлиять на потребление аппаратных ресурсов, является применение методов интегрирования с постоянным шагом, которые могут существенно снизить время вычислений. Это может произойти благодаря тому, что при использовании методов интегрирования с постоянным шагом имеется возможность при формировании математической модели вычислить проводимость уже на этапе компиляции не только для сопротивлений, но и для конденсаторов. Поскольку в схемах, полученных после экстракции паразитных элементов резисторов и конденсаторов большее число, чем любых других элементов, переход на постоянный шаг метода интегрирования позволит уменьшить время формирования математической модели.

Выводы:

1. Предлагается алгоритм компиляционного формирования математических моделей ИС на этапе схемотехнического проектирования, позволяющий ускорить процесс формирования математических моделей без дополнительных затрат оперативной памяти.

2. Предлагается алгоритм работы программ симуляторов, позволяющий в процессе работы использовать разработанный алгоритм формирования математических моделей ИС.

3. Показаны преимущества предлагаемого алгоритма компиляционного формирования математических моделей ИС по сравнению с существующими методами ускорения этапа формирования математических моделей ИС.

4. Предложенный алгоритм рассматривается в применении к формированию математических моделей ИС при моделировании во временной области, однако его использование, благодаря проиллюстрированным преимуществам, позволит сократить время формирования математических моделей ИС при выполнении других видов анализа.

Список использованной литературы

1. Казённов Г. Г. Основы проектирования интегральных схем и систем. М. :Бином. Лабораториязнаний . 2005.
2. Булах Д.А. Использование языка Verilog-A в современных схемотехнических САПР. // Известия высших учебных заведений. Электроника. , №1. 2007.
3. Булах Д.А. Язык Verilog-A как средство описания моделей полупроводниковых элементов. // Микроэлектроника и информатика-2006, тезисы доклада. Москва. 2006.
4. Вержбицкий В.М. Основы численных методов.- М. :Высшаяшкола, 2002.
5. Лобанов А.И. , Петров И.Б. Лекции по вычислительной математике. М. :Бином. Лабораториязнаний. 2006.
6. Кузовкин В. А. Теоретическая электротехника: учебник для ВУЗов. М.:Логос. 2006.
7. Валях Е. Последовательно-параллельные вычисления. - М.: Мир, 1985.
8. Плохотников К.Э. Математическое моделирование и вычислительный эксперимент.
9. Методология и практика. Едиториал УРСС, 2003.
10. Устинов СМ., Замницкий В. А. Вычислительная математика. БХВ-Петербург, 2009.
11. Theoretical evaluating the effectiveness of the algorithm. International Scientific and Practical Conference "WORLD SCIENCE", Proceedings of the IInd International Scientific and Practical Conference "Modern Scientific Achievements and Their Practical Application (October 20-21, 2015, Dubai, UAE)". № 3(3), Vol.1, November 2015. P. 16-20