

ОПТИМИЗАЦИЯ ВРЕМЕНИ РАБОТЫ АЛГОРИТМА ДЛЯ МНОГОЯДЕРНЫХ ПРОЦЕССОРОВ В ПРОГРАММНОМ КОМПЛЕКСЕ ТЕПЛОВОГО ПРОЕКТИРОВАНИЯ ЭЛЕКТРОННЫХ МОДУЛЕЙ

Е. К. Гребенникова, П. И. Кандалов

*Федеральный научный центр Научно-исследовательский институт
системных исследований Российской академии наук
ekaterina.grebennikova@inbox.ru, petrki87@gmail.com*

Статья посвящена оптимизации времени работы алгоритма программного комплекса STF-ElectronMod, предназначенного для теплового проектирования и моделирования температурных полей электронных модулей как в детерминированных, так и в условиях неопределенности входных данных. Неопределенность входных данных моделируется интервально стохастической моделью, а оптимизация достигается методом распараллеливания матричных вычислений на стационарных компьютерах с многоядерными процессорами. Приведены обоснование выбора способа оптимизации, результаты тестовых вычислений и их анализ. Приведено сравнение скорости работы библиотеки, написанной на языке программирования C++, и скорости работы функции программного комплекса STF-ElectronMod, написанной на языке Pascal. Показано, что выигрыш во времени исполнения библиотечной функции, написанной на C++ при одном задействованном потоке, составляет не менее 30 % по отношению к времени счета, осуществляемого по изначальному неоптимизированному алгоритму программного комплекса STF-ElectronMod. Данное ускорение связано в первую очередь с возможностью компилятора, поддерживающего язык программирования C++, разворачивать циклы для выполнения задач в конвейере команд, что практически не используется в компиляторе Pascal. Результаты тестов показали, что производительность реализованного алгоритма на 4-ядерном процессоре приблизительно в 6 раз выше. Алгоритм оптимизации реализован в виде отдельной библиотеки и включен в программный комплекс.

Ключевые слова: распараллеливание, оптимизация, многоядерные процессоры, компьютерное моделирование.

OPTIMIZATION OF ALGORITHM EXECUTION TIME FOR MULTICORE PROCESSORS IN THERMAL DESIGNING SOFTWARE OF ELECTRONIC MODULES

E. K. Grebennikova, P. I. Kandalov

*System Research Institute, Russian Academy of Sciences
ekaterina.grebennikova@inbox.ru, petrki87@gmail.com*

The article is devoted to the optimization of the algorithm execution time in the STF-ElectronMod software. The software is intended to design and model the temperature fields of electronic modules in both deterministic conditions and under uncertainty. The uncertainty of the input data is modeled by the stochastic interval model, and optimization is achieved by parallelizing matrix calculations on PC with multi-core processors. Optimization method is provided and results are presented. A comparison between the speed of the library written in C++ and the speed of the function of the STF-ElectronMod software package written in Pascal is made. It is shown that the function performance of C++ library compared to Pascal function is not less than 30 % using the original non-optimized algorithm of the STF-ElectronMod software package. This performance is primarily due to the ability of the C++ compiler to use loops unrolling, which is practically not used in the Pascal compiler. The test results showed that performance of the new algorithm may be 6 times higher on

the quad-core processor. The algorithm has been developed as a library and included in STF-ElectronMod software package.

Keywords: parallelization, optimization, multi-core processors, computer simulation.

Создание современных конкурентоспособных электронных систем (ЭС) невозможно без теплового проектирования и обеспечения тепловых режимов ЭС и элементов [1–2]. Одной из структурных единиц ЭС является электронный модуль (ЭМ), состоящий из многослойной печатной платы (МПП) с разнородными слоями и установленными на ней элементами – микросхемами (МС) и электро-радиоэлементами (ЭРЭ) – дискретными диодами, транзисторами, резисторами, конденсаторами, катушками индуктивности, электрическими разъемами и пр. Математическое и компьютерное моделирование температурных полей ЭМ является одной из главных проблем при тепловом проектировании. Такое положение во многом обусловлено сложным характером конструкции ЭМ, обладающей такими конструктивными характеристиками, как большое число (до нескольких десятков) разнородных в теплофизическом отношении слоев в МПП, ни одним из которых практически нельзя пренебречь; трехмерный характер и сложная пространственная конфигурация конструкции ЭМ, МС и ЭРЭ; неоднородные включения и полости; сложная динамика рассеиваемой мощности при работе ЭМ; взаимодействие элементов между собой в системе. Необходимо отметить, что на рынке программных комплексов (ПК) имеется довольно много как зарубежных, так и отечественных ПК, предназначенных для моделирования тепловых режимов различных ЭС и ЭМ, обладающих вместе с тем серьезными недостатками, которые не позволяют использовать их в профессиональной практике проектирования электронных систем [1–2]. Наиболее адекватным по точности моделирования является ПК STF-ElectronMod [3–4], созданный для теплового проектирования ЭМ, расчета трехмерных температурных полей в сложных конструкциях ЭМ, не имеющего ограничений на количество слоев в МПП, сложность конструкции, реальных условий монтажа и охлаждения. Кроме того, ПК STF-ElectronMod позволяет осуществлять анализ и моделирование температурных распределений в ЭМ в условиях неопределенности входных данных (мощностей МС, тепловых сопротивлений корпусов МС, условий монтажа, контактных тепловых сопротивлений, температуры окружающей среды и т. д.), не имеющих аналогов в зарубежных и российских ПК.

В основу системы математического и компьютерного моделирования температурных полей ЭМ в ПК STF-ElectronMod был положен новый матрично-топологический метод, разработанный в [3]. В процессе работы алгоритма метода производится многократное формирование матриц большой размерности многократное обращение к различным матричным операциям. Ввиду большой размерности матриц – до сотен элементов, в зависимости от требуемой точности моделирования, количества МС и сложности конструкции – операции формирования матриц и действий над ними занимают значительную часть времени работы ПК. В этой связи возникает необходимость в оптимизации времени вычислений при работе алгоритма ПК STF-ElectronMod. Учитывая, что современные персональные компьютеры являются многоядерными, наиболее эффективным методом оптимизации (минимизации) времени счета является распараллеливание алгоритма.

В статье рассматривается метод оптимизации времени работы алгоритма ПК STF-ElectronMod путем распараллеливания вычислений на персональных компьютерах с многоядерными процессорами.

Основная задача, решаемая при оптимизации времени работы алгоритма, состояла в оптимизации того участка вычислительного ядра ПК, отвечающего за вычисления матриц теплового влияния – R [3–5], на которые приходится основное время, занимаемое вычислениями.

Каждый элемент матрицы R является независимым от других элементов, поэтому необходимости в распараллеливании на уровне передачи сообщений или по данным нет, достаточно использовать модель параллелизма с общей памятью [6]. Для решения данной задачи

была использована технология OpenMP [7], так как данная технология является кроссплатформенной, позволяет использовать каждое ядро процессора персонального компьютера как отдельный вычислительный элемент, а также поддерживает технологию multi-threading [8]. Для нахождения матрицы R в ПК STF-ElectronMod используется подпрограмма, которая использует в своей реализации набор функций ядра. Для использования технологии OpenMP данная функция была переписана в качестве библиотеки на язык C++, позволяющий поддерживать данную технологию и, кроме того, является довольно распространенным и наиболее часто используемым высокоуровневым кроссплатформенным языком программирования. Поэтому в тестовых данных будет сравниваться скорость работы алгоритма на языке Pascal со скоростью работы алгоритма при разном числе потоков на языке C++.



Рис. 1. Структура алгоритма компьютерного моделирования

Структура библиотечной функции на языке C++ приведена в виде блок-схемы на рис. 1. В ПК STF-ElectronMod полное описание параметров ЭМ задается на этапе ввода исходных данных. В библиотечную функцию передается часть необходимых параметров, таких как размер МПП, толщина и коэффициенты теплопроводности слоев, для каждого элемента описываются величины потребляемой мощности, геометрические размеры, координаты центра и положение относительно МПП.

На первом этапе на основе переданных параметров для каждой комбинации количества

членов m и n двойного ряда, получаемого косинус-преобразованием Фурье, производится расчет двумерной матрицы G [3–4]. На втором этапе на основе полученной матрицы G для каждого сочетания m и n рассчитываются постоянные коэффициенты, зависящие от положения элементов и параметров слоев МПП, но не зависящие от мощности потребления элементов [3–4]. На следующем этапе рассчитывается образ температуры, для определения которого используются найденные постоянные коэффициенты матрицы G и мощности на элементах, вычисленные с помощью подпрограммы [3–4]. Последний этап отвечает за вычисления элементов матрицы R , осуществляемых на основе вычисленного ранее температурного образа.

Время вычислений 1-го и 2-го этапов является незначительным и потому не нуждается в оптимизации. Основные вычислительные затраты приходятся на 3-й и 4-й этапы, которые и следует распределить между потоками. Отметим, что матрица R является симметричной, поэтому достаточно вычислить верхнюю или нижнюю треугольную матрицу, и затем отразить ее элементы относительно диагонали. В рассматриваемом случае вычисляется верхняя треугольная матрица.

Разработанная библиотека задействует максимальное число потоков, поддерживаемых системой, на которой в данный момент выполняется работа приложения с учетом возможности использования технологии multi-threading посредством функции OpenMP [8]. Нахождение верхней треугольной матрицы R распределяется по потокам построчно, начиная от верхней строки матрицы (с наибольшим числом элементов) и заканчивая нижней строкой, состоящей из одного элемента. Посредством такого распределения строк матрицы R по потокам обеспечивается равномерная балансировка загрузки многоядерного процессора. С помощью динамического планирования (dynamic scheduling) [8] каждый поток, завершивший подсчет строки матрицы R , приступает к вычислению не рассчитанной строки матрицы, не дожидаясь завершения подсчета других строк. Данная функция позволяет эффективно осуществлять распределение вычислений, так как разные потоки могут выполнять примерно равноценные вычислительные задачи за разное время, что связано в первую очередь с неравномерной загруженностью ядер процессора другими приложениями и их различной тактовой частотой. Последним этапом алгоритма является отражение треугольной матрицы R и передача ее из библиотеки в вызывающее приложение.

Приведем основные результаты тестового моделирования. Количество вычислительных операций не зависит от реального положения элементов на МПП, поэтому для сбора данных о скорости вычисления матрицы R достаточно задать количество элементов сверху и снизу МПП и количество слоев в ней. Остальные параметры задаются в тестовом варианте автоматически, а именно: элементы располагаются на МПП последовательно, а слои МПП имитируют слои в реальных системах с помощью чередования диэлектрических слоев из полиимида и тонких медных энергетических слоев. Таким образом, размеры МПП, элементов и центров элементов по оси координат X являются постоянными, а по оси Y вычисляются согласно количеству элементов ЭМ. Данные исследований показали также, что двойные ряды, получаемые в результате преобразования Фурье [3], являются быстросходящимися, поэтому для моделирования температуры с приемлемой для практики теплового проектирования точностью достаточно учитывать первые сто членов ряда.

Тестирование показало, что время, затрачиваемое на вычисления матрицы R , в малой степени зависит от числа слоев МПП, но довольно сильно зависит от количества элементов в математической модели ЭМ, поэтому основное тестирование проводилось при различном количестве элементов в системе. В качестве тестовой машины использовался компьютер под управлением ОС Windows 8.1 с 4-ядерным процессором Intel Pentium CPUN3540 с частотой 2,16 ГГц без поддержки технологии multi-threading. Для проверки правильности определения времени для каждого набора числа потоков и элементов при фиксированном количестве слоев МПП, равном 30, как наиболее часто используемом на сегодняшний день, было проведено

несколько замеров времени работы алгоритма и получено среднее значение. Результаты тестирования времени (с) работы библиотечной функции, вычисляющей матрицу R, приведены в табл. 1 и на рис. 2.

Таблица 1

Время работы библиотечной функции для вычисления коэффициентов матрицы R

Число потоков	Количество элементов матрицы R			
	50	100	150	200
1	5,2	19,7	45,1	75,8
2	2,6	9,8	21,8	38,2
3	1,8	6,6	14,7	26,1
4	1,4	5,3	11,6	21,0

Данные о времени работы функции, вычисляющей матрицу R на языке Pascal, представлены в табл. 2. Так как данная версия функции не учитывала симметричность матрицы R, то для чистоты сравнения было использовано среднее значение между испытаниями, деленное на два, поскольку вычисление полной матрицы производится практически в два раза медленнее, чем треугольной, включая ее отражение относительно диагонали матрицы.

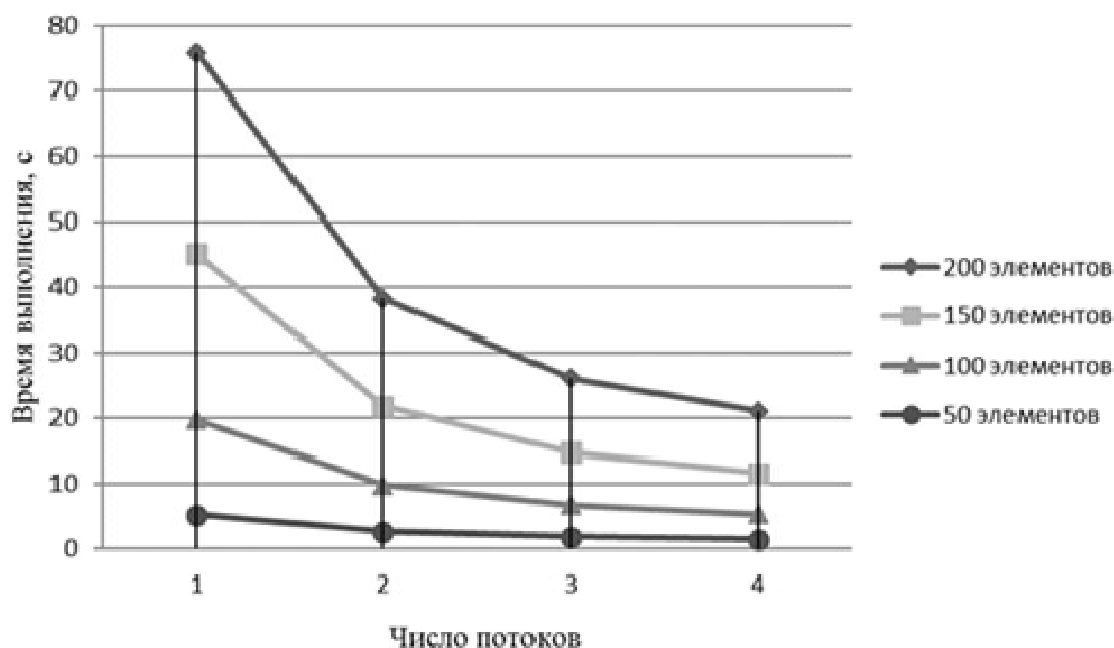


Рис. 2. Время вычисления матрицы R при разном числе потоков

Таблица 2

**Время работы функции для вычисления
коэффициентов матрицы R на языке Pascal**

Количество элементов матрицы R	50	100	150	200
Среднее время вычисления диагональной матрицы, с	8,3	31,6	70,6	128,9

Матрица R квадратично зависит от количества элементов, причем данная зависимость наблюдается и в скорости ее вычисления с помощью библиотеки, приведенной в табл. 1 и на рис. 2. С увеличением числа элементов время выполнения увеличивается более чем в 2 раза,

что связано с увеличением не только самой матрицы, но и с увеличением количества вычислений при нахождении ее элементов. Анализируя график, можно заметить, что при малом количестве элементов время вычисления в меньшей степени зависит от числа задействованных потоков. При 100–150 элементах в ЭМ наибольший прирост времени вычислений наблюдается при 2–3 потоках. Использование четырех и более потоков наиболее выгодно при количестве элементов, большем 150. Отметим также, что выигрыш в ускорении времени вычислений при использовании большего числа потоков постепенно сокращается, что обусловливается дополнительными задержками при получении и записи данных для каждого нового потока данных.

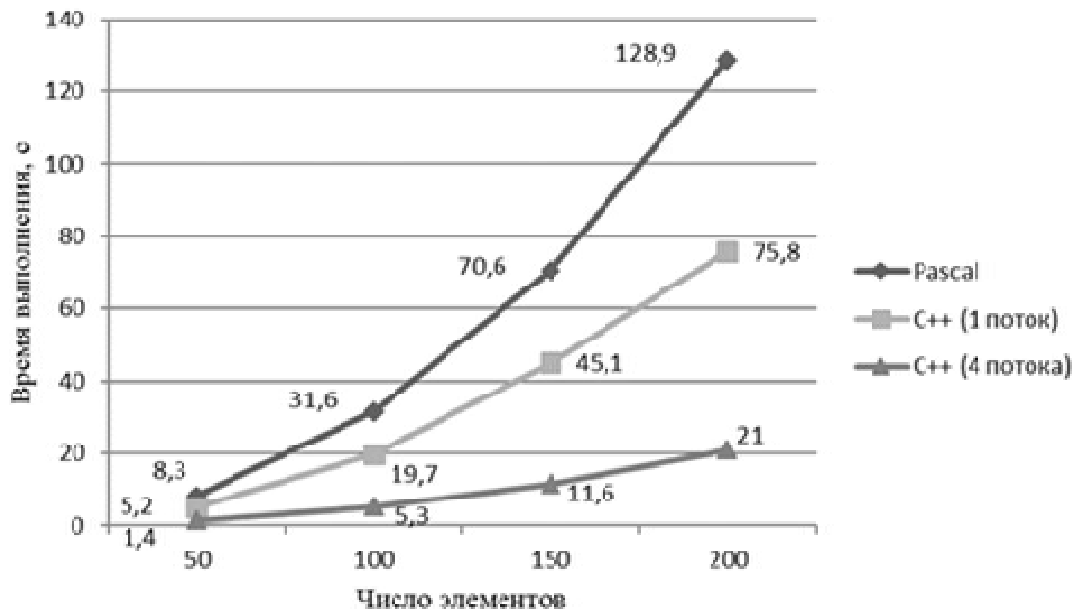


Рис. 3. Сравнение производительности алгоритма на языках C++ и Pascal

На рис. 3 отображена скорость работы библиотеки, написанной на языке C++, и скорость работы функции ПК STF-ElectronMod, написанной на языке Pascal. Согласно данным табл. 1 и 2 и рис. 3 выигрыш во времени исполнения библиотечной функции, написанной на C++, при одном задействованном потоке составляет не менее 30 % по отношению ко времени счета, осуществляемого по первоначально заложенной в ПК функции. Данное ускорение связано в первую очередь с возможностью компилятора, поддерживающего язык программирования C++, разворачивать циклы для выполнения задач в конвейере команд, что практически не используется в компиляторе Pascal. А при использовании всех четырех потоков скорость будет приблизительно в 6 раз выше для C++, чем для Pascal.

Литература

1. Мадера А. Г. Концепция математического и компьютерного моделирования тепловых процессов в электронных системах // Программные продукты и системы. 2015. № 4 (112). С. 79–86.
2. Мадера А. Г., Решетников В. Н. Многофункциональный программный комплекс теплового проектирования электронных систем: требования к архитектуре и функциональным возможностям моделирования // Программные продукты и системы. 2017. № 3 (30). С. 367–372.
3. Мадера А. Г., Кандалов П. И. Матрично-топологический метод математического и компьютерного моделирования температурных полей в электронных модулях: программный комплекс STF-ElectronMod // Программные продукты и системы. 2012. № 4. С. 160–164.

4. Мадера А. Г., Кандалов П. И. Моделирование трехмерных температурных полей в электронных модулях // Программные продукты и системы. 2010. № 2. С. 29–33.
5. Свидетельство о гос. регистрации программы для ЭВМ № 2015619835. Заявка № 2015613488. Зарегистрировано в Реестре программ для ЭВМ 15.09.2015.
6. Уильямс Э. Параллельное программирование на C++ в действии: практика разработки многопоточных программ. М. : ДМК Пресс, 2016. 672 с.
7. Левин М. П. Параллельное программирование с использованием OpenMP : учеб. пособие / М. : ИНТУИТ ; БИНОМ. Лаборатория знаний, 2008. 188 с.
8. Хьюз К, Хьюз. Т. Параллельное и распределенное программирование с использованием C++. М. : Вильямс, 2004. 672 с.