

Заключение

Предложенный подход к построению персонализированных пользовательских интерфейсов автоматизированных систем позволяет учитывать особенности когнитивного портрета отдельных пользователей и адаптировать взаимодействие с информацией в системе к этим особенностям. Это может повысить уровень комфорта работы пользователя системы с информацией, снизить уровень информационной перегрузки, улучшить сосредоточенность и время реакции на критические события.

Поскольку величина времени реакции пользователя влияет на общий уровень надежности и эффективности функционирования системы, можно рассчитывать, что применение адаптивных персонализированных пользовательских интерфейсов позволит повысить показатели надежности и эффективности автоматизированных систем.

Литература

1. Турзин П.С., Пономаренко В.А., Рысакова-Ромашкан С.Л. Уровни понимания информации и структура коммуникативного акта // Психологический журнал. 1992. Т. 13. № 1. С. 30–39.
2. Солсо Р. Когнитивная психология. 6-е изд. СПб: Питер, 2006. 589 с.
3. Якиманская И.С. Принципы построения образовательных программ и личностное развитие учащихся // Вопросы психологии. 1999. Т. 13. № 3. С. 39–47.
4. Гальскова Н.Д. Современная методика обучения иностранным языкам: Пособие для учителя. М.: АРКТИ, 2004. 192 с.
5. Ляховицкий М.В. Методика преподавания иностранных языков: Методическое пособие. М.: Дрофа, 2006. 369 с.
6. Верлань А.Ф., Сопель М.Ф., Фуртат Ю.О. Особенности оперативного тестирования на рабочем месте операторов систем поддержки принятия решений (СППР) // Математичне та комп'ютерне моделювання. Сер. Технічні науки. 2010. В. 3. С. 37–45.
7. Немов Р.С. Психология: Учебник для студентов высш. пед. учеб. заведений: В 3-х кн. 4-е изд. М.: Гуманитарный издательский центр «ВЛАДОС», 2003. Кн. 1. 688 с.

Фуртат Юрий Олегович

– младший научный сотрудник, Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, Киев, Украина, saodhar@ukr.net

Yurii Furtat

– junior scientific researcher, Ukrainian National Academy of Sciences, Pukhov Institute for Modeling in Energy Engineering, Kiev, Ukraine, saodhar@ukr.net

УДК 004.274

МЕТОД ОТОБРАЖЕНИЯ ЗАДАЧ НА КРУПНОГРАНУЛЯРНЫЕ РЕКОНФИГУРИРУЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

А.С. Румянцев^а

^а Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Россия, darkstreamray@gmail.com

Произведен анализ существующих подходов к отображению задач на реконфигурируемые вычислительные системы, особое внимание уделялось методам отображения на крупногранулярные реконфигурируемые вычислительные системы. На основе произведенного анализа сформированы цель и задачи создания нового эвристического метода отображения задач на крупногранулярные реконфигурируемые вычислительные системы, который базируется на методе разделения графа с выталкиванием вершин, алгоритме покрытия графов, эвристическом подходе к оптимизации и упаковке графа для конкретного варианта крупногранулярной реконфигурируемой вычислительной системы и разработанном методе отображения графа потока данных задачи на ресурсы крупногранулярной реконфигурируемой вычислительной системы. В ходе работы было осуществлено имитационное моделирование разработанного метода и существующих подходов к отображению задач на крупногранулярные реконфигурируемые вычислительные системы на модели системы с крупногранулярной реконфигурируемым аппаратным ускорителем MATRIX. Приведены экспериментальные результаты, доказывающие эффективность предлагаемого подхода по сравнению с широко используемыми методами отображения задач на крупногранулярные реконфигурируемые вычислительные системы и возможность использования динамических параметров функционирования крупногранулярной реконфигурируемой вычислительной системы для дальнейшего улучшения получаемого отображения задачи.

Ключевые слова: крупногранулярные реконфигурируемые вычислительные системы, отображение задач на вычислительные системы, покрытие графов.

TASKS MAPPING METHOD FOR COARSE GRAIN RECONFIGURABLE SYSTEMS

A. Romyantsev^b

^b Saint Petersburg State University of Information Technologies, Mechanics and Optics, Saint Petersburg, Russia, darkstreamray@gmail.com

This paper deals with analysis of existing approaches to tasks mapping on reconfigurable computing systems with special attention paid to mapping methods for coarse grained reconfigurable computing systems. The purpose and objectives of a new heuristic method for tasks mapping on coarse grained reconfigurable computing systems are produced on the base of the carried out analysis. This novel method for tasks mapping on coarse grained reconfigurable computing systems is based on the method of graph partitioning with pushing vertices, graph covering algorithm, a heuristic approach to optimizing and packaging for a particular graph on coarse grained reconfigurable computing system and displaying methods of data flow graph on resources of coarse grained reconfigurable computing system. The simulation was conducted on system model with coarse grained reconfigurable hardware accelerator MATRIX. Experimental results are given. They prove the effectiveness of the proposed approach as compared with the widely used methods for tasks mapping on coarse grained reconfigurable computing systems and the ability to use dynamic functional parameters of coarse grained reconfigurable computing system to further improvement of the mapping results.

Keywords: coarse grain reconfigurable computing systems, tasks mapping on computing systems, graph covering.

Введение

Для того чтобы крупногранулярные реконфигурируемые вычислительные системы (КРВС) [1] заняли свою нишу между цифровыми сигнальными процессорами и программируемыми логическими интегральными схемами (ПЛИС), они должны быть просты в использовании. Целевое приложение должно быть легко компилируемым и загружаемым во встроенную систему, в которой используется КРВС. Отображение задачи на КРВС подразумевает выделение в задачи вычислительно-интенсивных последовательностей операций, которые могут быть отображены на КРВС, назначение временных интервалов исполнения операций задачи, отображение данных операций на конкретную архитектуру вычислительных элементов КРВС и маршрутизацию входных и выходных потоков данных и управления вычислительных элементов КРВС [1]. В процессе отображения задачи на КРВС необходимо учитывать следующие проектные ограничения из-за фиксированного набора ресурсов КРВС: организацию вычислительных элементов, способ доступа к памяти, способ организации межсоединения, порты ввода-вывода, синхронизацию и механизм реконфигурации [2]. Стоит отметить, что в своей общей трактовке задача отображения приложения на КРВС входит в класс комбинаторных задач с неполиномиальной оценкой числа итераций (является NP-сложной), так же как и проблема планирования задач на многопроцессорных системах [3].

В настоящей работе предлагается новый эвристический метод отображения задачи на встроенную систему с КРВС, который поддерживает различные типы КРВС и их конфигурации, обеспечивая наименьшее время поиска решения из всех рассмотренных эвристических методов отображения. В первом разделе представлен обзор существующих методов отображения задачи на КРВС. Во втором разделе приведено описание разработанного эвристического метода отображения. В третьем разделе приведено сравнение предлагаемого метода отображения с существующими решениями.

Обзор существующих решений

Подход в [4] применяет отображение на основе модульного планирования для максимального использования параллелизма в циклах. Модульное планирование позволяет конвейеризировать выполнение циклов [5]. Данный подход создает расписание для одной итерации заданного цикла таким образом, чтобы это же расписание можно было выполнять через определенные промежутки времени с учетом всех внутренних и внешних зависимостей и ограничений по ресурсам. В работе [6] представлен метод отображения задачи на основе модульного планирования с приоритизацией по выходным потокам данных. Метод призван ускорить процесс отображения задачи по сравнению с подходом на основе модульного планирования. Таким образом, вместо того чтобы до маршрутизации по потокам данных произвести отображение операций на вычислительные элементы КРВС, данный метод откладывает задачу отображения операций на вычислительные элементы до тех пор, пока не будет собрана необходимая информация о маршрутизации входных и выходных поток данных.

Подход в работе [7] использует высокоуровневый синтез для отображения задач на КРВС FloRA. Подход использует параллелизм в циклах для их конвейеризации. Для решения задачи отображения сначала применяется алгоритм планирования на основе списка для поиска начального решения. В дальнейшем для улучшения решения применяется эволюционный алгоритм QEA (Quantum-inspired Evolutionary Algorithm) [8].

В разработанном методе отображения используется подход, базирующийся на подходах к эффективному покрытию графа с помощью заданных или генерируемых шаблонов [9, 10]. В работе [11] данная методика используется на стадии покрытия графа потока данных задачи шаблонами поддерживаемых

операций. В дальнейшем происходит оптимальное разделение графа задачи на отдельные кластеры для непосредственного отображения на КРВС путем решения задач целочисленного линейного программирования. С целью избавления от необходимости решения многочисленных ресурсоемких задач линейного программирования в [12] был представлен эвристический метод решения задачи кластеризации, SPKM (Split-Push Kernel Mapping). Данный метод базируется на основе подхода к построению графа произвольной сложности с использованием операций разделения дуг графа и образования новых вершин [13].

Во всех рассмотренных подходах к отображению задач на КРВС отсутствует потенциальная возможность применения метода к другому типу КРВС одновременно с независимостью метода от структурной организации КРВС. Кроме того особенность организации быстрых эвристических методов отображения задач на КРВС делает невозможным использование динамических параметров функционирования системы для дальнейшей оптимизации отображения.

Эвристический метод отображения программы на КРВС

Основная цель эвристического метода заключается в том, чтобы отобразить программу цифровой обработки сигналов или мультимедийных данных, написанных на языке высокого уровня, например, языке Си, на встроенную систему с процессором общего назначения и спроектированным КРВС, что значительно ускорит выполнение данной программы и уменьшит энергопотребление системы [14]. Разработанный эвристический метод отображения с использованием алгоритма «разделения и выталкивания» вершин графа (OPB) во многом основан на подходах из области составления оптимального расписания для многопроцессорных систем [15, 16] и эвристическом методе решения задачи кластеризации SPKM [12].

Предлагаемый метод состоит из четырех этапов.

1. Преобразование исходного кода в граф потока данных с тегированием управляющими командами с выделением вычислительно-интенсивных ядер и выполнении оптимизаций над полученным графом. На данной стадии используется платформа для исследования методов компиляции для высокопроизводительных машин SUIF [17]. После использования инструментария SUIF получается в общем случае неациклический граф потока данных и управления. В дальнейшем используются подходы к оптимизации на основе «гиперблоков», представленных в работе [18], а также выделение подходящих «гиперблоков» вычислительно интенсивных ядер с использованием набора заранее заданных критериев для последующего отображения их на КРВС. Участки исходного кода, которые оказались нецелесообразно исполнять на КРВС или которые не удалось отобразить в результате последующих этапов метода, будут исполняться на процессоре общего назначения.
2. Кластеризация полученного графа на полносвязанный граф неограниченного количества вычислительных элементов КРВС с использованием алгоритма покрытия графов [9]. Задача данного этапа состоит в том, чтобы отобразить полученные на предыдущем этапе графы на граф, состоящий из неограниченного количества вычислительных элементов КРВС, при этом все соседние вычислительные элементы в этом графе связаны друг с другом. Во время выполнения операции кластеризации осуществляется попытка уменьшить количество используемых вычислительных элементов КРВС, их различных конфигураций и критический путь исходного графа. В результате выполнения второй стадии получается граф отображения задач на вычислительные элементы КРВС.
3. Планирование отображения с сжатием полученного графа потока данных позволяет учесть количество вычислительных элементов КРВС и связи между ними. Кроме того, на данном этапе могут быть использованы динамические параметры функционирования системы для получения оптимального сжатия графа конкретной задачи. Предлагаемый метод планирования состоит из двух стадий:
 1. использование метода разделения графа с выталкиванием вершин [13] и нахождение минимального сечения графа на основе применения метода оптимального планирования задач на многопроцессорных системах CASS-II [19];
 2. применение разработанных эвристических подходов к оптимизации и упаковке графа для конкретной конфигурации КРВС.

Предлагаемые эвристические подходы к оптимизации и упаковке графа:

- осуществляется попытка изменения конфигурации транзитных вершин с использованием маршрутизации на основе дерева «Штейнера» [20];
- осуществляется избавление от транзитных вершин, которые могут быть убраны в определенных случаях;
- задействуются возможности контроллера вычислительного элемента второго уровня для последовательного выполнения в одной вершине нескольких операций, что позволяет уменьшить критический путь в отображенном графе задаче и уменьшить количество транзитных вершин;
- производится процедура «сжатия» графа по строкам и столбцам за счет коммутации входного и выходного банка памяти и использования контроллера вычислительного элемента второго уровня.

4. Финальное отображение на ресурсы КРВС исходных алгоритмов с генерацией конфигураций для каждого вычислительного элемента КРВС. Задачи этого этапа.

1. Уместить в имеющиеся ресурсы КРВС и в ресурсы ее отдельных вычислительных элементов подготовленные графы задач. Для достижения заданных целей используется разработанный алгоритм отображения распланированного графа на ресурсы КРВС, псевдокод которого приведен ниже:

/* На вход подается распланированный и сжатый на предыдущем шаге граф G

На выходе формируется конфигурация для каждого вычислительного элемента КРВС для каждого такта */

```

0 function AllocateCGRAResources(G) {
1   foreach row in G {
2     Allocate PEs of the current clock cycle;
3   do {
4     try to move each output to memory or internal
5     PE's register or stack memory
6   } if failed GOTO label_recover
7   do {
8     try to move each input to proper PE's outputs
9     at the clock cycle, which is N steps before, where
10    N in (1,2,3,4,...,starting_step)
11  } if failed GOTO label_recover
12  GOTO label_next_row
13  label_recover: insert one or more clock cycles
14  before the current one
15  label_next_row:
16  }
17  }

```

2. Формирование конфигурационных карт для каждого цикла исполнения отдельных вычислительных элементов КРВС. Множество таких конфигурационных карт представляет собой конфигурационную программу исполнения исходной задачи на КРВС.

Экспериментальные результаты

Для проведения экспериментальных измерений были выбраны различные методы эвристического отображения задач на КРВС: DRESC [4], AHN [11], SPKM [12], QEA [8], а также разработанный метод OPB. Все эксперименты проводились на упрощенной модели встроенной системы, реализованной на языке проектирования системного уровня SystemC, с вариантом КРВС MATRIX [21], которая поддерживает конфигурируемое распределение вычислительных ресурсов. КРВС MATRIX состоит из массива одинаковых 8-битных базовых вычислительных блоков 4×4, объединенных каналами передачи данных и конфигурационной сетью. Каждый базовый вычислительный блок содержит блок памяти на 256 Б, 8-битовое арифметико-логическое устройство и блок снижения логики управления. На рис. 1 показана доля успешных отображений для выборки из случайных задач в зависимости от количества вершин в отображаемом графе потока данных для различных методов отображения. Из графика видно, что разработанный метод OPB заметно опережает в эффективности методы AHN, DRESC и SPKM, находясь наравне с методом QEA.

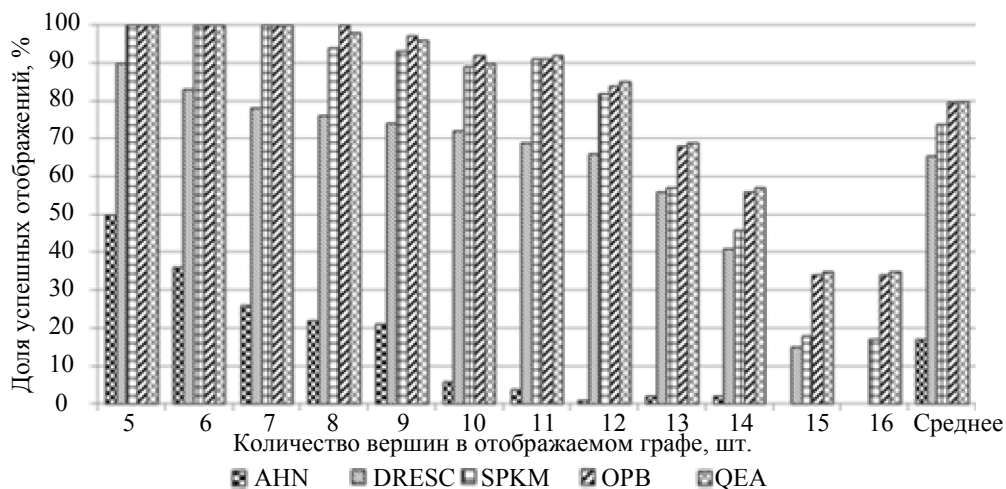


Рис. 1. Зависимость количества успешных отображений от количества вершин в графе

На рис. 2 показана зависимость времени выполнения метода от количества вершин в отображаемом графе потока данных. Разработанный метод ОРВ в среднем оказывается в 1,5 раза быстрее остальных эвристических методов. По времени выполнению отображенной задачи на КРВС из рис. 3 видно, что разработанный метод ОРВ показывает себя не хуже, а в 6 из 10 проведенных тестов даже лучше остальных методов.

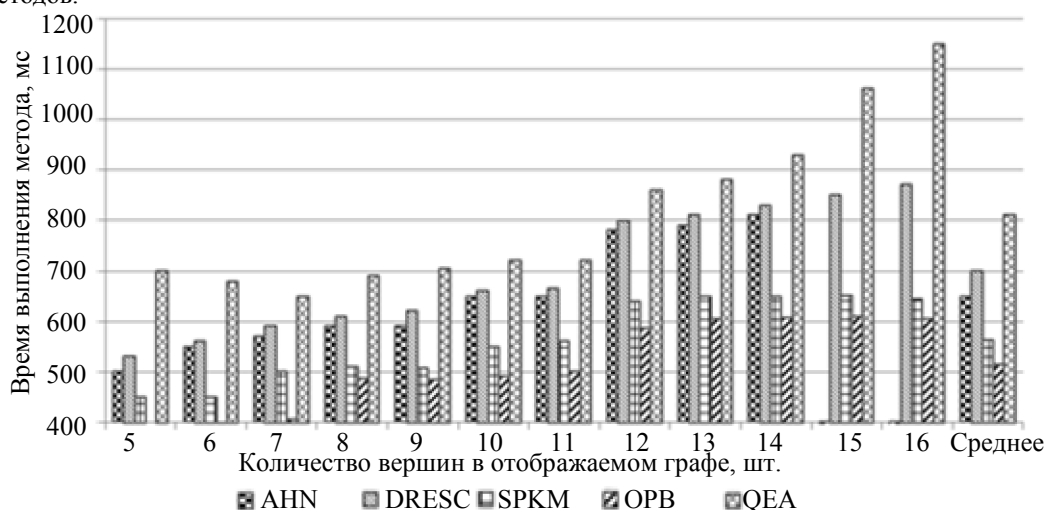


Рис. 2. Зависимость времени выполнения метода от количества вершин в графе

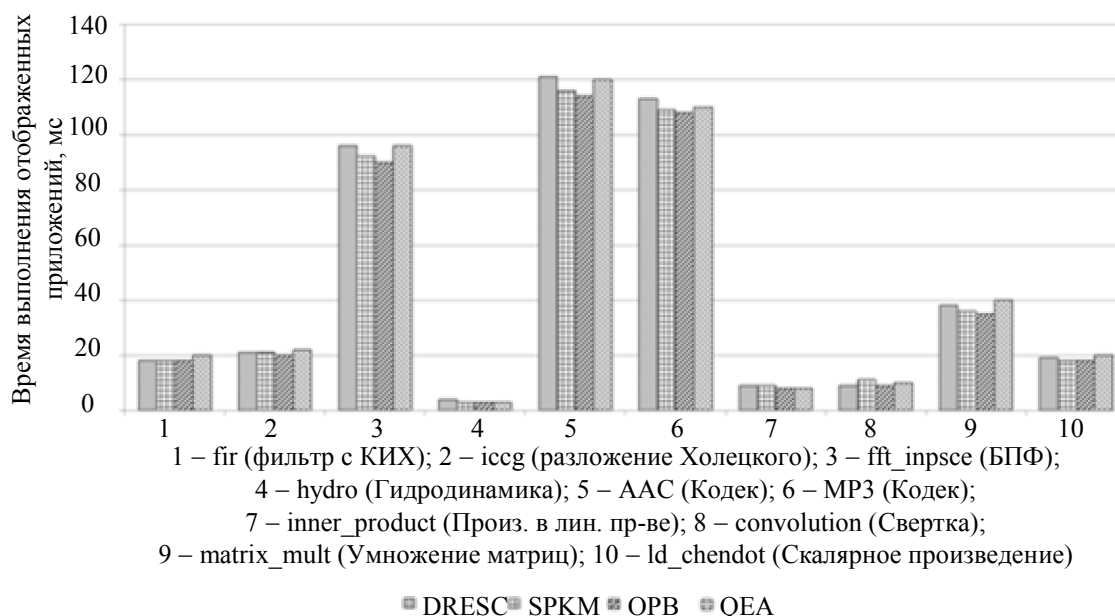


Рис. 3. Относительное время выполнения задач [22], отображенных на систему различными методами

Заключение

В рамках работы предложен новый эвристический метод отображения задач на крупногранулярные вычислительные системы ОРВ, который выполняется на 20–40% быстрее существующих эвристических методов отображения и обеспечивает оптимальное отображение задач более чем в 15% случаев. Кроме того, по эффективности отображенной задачи представленный метод оказывается не хуже существующих методов и позволяет использовать динамические параметры функционирования системы для дальнейшего улучшения получаемого отображения задачи.

Литература

1. Румянцев А.С. Организация и инструментальные средства реконфигурируемых вычислительных систем // Научно-технический вестник информационных технологий, механики и оптики. 2012. № 4 (80). С. 80–86.
2. Hannig F., Dutta H., Teich J. Mapping of Regular Nested Loop Programs to Coarse-Grained Reconfigurable Arrays - Constraints and Methodology // Proc. 18th International Parallel and Distributed Processing Symposium (IPDPS'04). IEEE Computer Society, 2004. Santa Fe, USA (2004). Workshop 3. 8 p.

3. El-Rewini H., Lewis T., Ali H. Task Scheduling in Parallel and Distributed Systems. Englewood Cliffs, NJ: Prentice Hall, 1994. 290 p.
4. Mei B., Vernalde S., Verkest D., De Man H., Lauwereins R. DRESC: a retargetable compiler for coarse-grained reconfigurable architectures // Proc. of the IEEE International Conference on Field-Programmable Technology. Hong Kong, 2002. P. 166–173.
5. Rau B. R. Iterative Modulo Scheduling // Proc. of the 27th Annual International Symposium on Microarchitecture. San Jose, California, 1994. P. 63–74.
6. Park H., Fan K., Mahlke S.A., Oh T., Kim H. Edge-Centric Modulo Scheduling for Coarse-Grained Reconfigurable Architectures // Proc. of the 17th International Conference on Parallel Architecture and Compilation Techniques (PACT-2008). Toronto, Ontario, Canada, 2008. P. 166–176.
7. Lee G., Lee S., Choi K., Dutt N. Routing-Aware Application Mapping Considering Steiner Point for Coarse-Grained Reconfigurable Architectures // Proc. of the 6th International Conference on Reconfigurable Computing: Architectures, Tools and Applications (ARC-2010). Berlin-Heidelberg: Springer-Verlag, 2010. P. 231–243.
8. Han K.-H., Kim J.-H. Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization // IEEE Trans. Evolutionary Computation. 2002. V. 6. N 6. P. 580–593.
9. Guo Y., Smit G.J.M., Broersma H., Heysters P.M. A Graph Covering Algorithm for a Coarse Grain Reconfigurable System // ACM Sigplan Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES'03). California, USA, 2003. P. 199–208.
10. Guo Y., Hoede C., Smit G.J.M. A Pattern Selection Algorithm for Multi-pattern Scheduling // Proc. of the 20th International Conference on Parallel and distributed processing (IPDPS' 06). IEEE Computer Society, 2006. P. 198–205.
11. Ahn M., Yoon J.W., Paek Y., Kim Y., Kiemb M., Choi K. A Spatial Mapping Algorithm for Heterogeneous Coarse-grained Reconfigurable Architectures // Proc. of the Conference on Design, Automation and Test in Europe (DATE'06). Leuven, Belgium, 2006. P. 363–368.
12. Yoon J.W., Shrivastava A., Park S., Ahn M., Paek Y. A Graph Drawing Based Spatial Mapping Algorithm for Coarse-Grained Reconfigurable Architectures // IEEE Transactions on Very Large Scale Integration Systems. 2009. V. 17. N 11. P. 1565–1578.
13. Battista G.D., Patrignani M., Vargiu F. A Split-and-Push Approach to 3D Orthogonal Drawing // 6th International Symposium on Graph Drawing. Lecture Notes in Computer Science. 1998. V.1547. Springer-Verlag, 1998. P. 87–101.
14. Петров С.В., Юрков К.В., Овсянников Е.П. Сравнительный анализ сложности реализации быстрых цифровых преобразований на RISC-процессорах // Изв. вузов. Приборостроение. 2011. Т. 54. № 9.С. 34–38.
15. Sarkar V. Partitioning and Scheduling Parallel Programs for Execution on Multiprocessors. Cambridge, MA: MIT Press, 1989. 213 p.
16. Liou J.-C., Palis M.A., Wei D.S.L. Performance analysis of task clustering heuristics for scheduling static DAGs on multiprocessor system // Parallel Algorithms and Applications. 1997. V. 12. N 1–3. P. 185–203.
17. Callahan T.J., Hauser J.R., Wawrzynek J. The Garp Architecture and C compiler // IEEE Computer. 2000. V. 33. N 4. P. 62–69.
18. Liou J.-C., Palis M.A. A New Heuristic for Scheduling Parallel Programs on Multiprocessor // Proc. of the International Conference on Parallel Architectures and Compilation Techniques (PACT'98). Paris, 1998. P. 358–365.
19. Patrignani M., Pizzonia M. The Complexity of the Matching-Cut Problem // Proc. of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'01). Lecture Notes in Computer Science. 2001. V. 2204. P. 284–295.
20. Cohoon J.P., Richards D.S., Salowe J.S. A Linear-Time Steiner Tree Routing Algorithm for Terminals on the Boundary of a Rectangle // IEEE International Conference on Computer-Aided Design (ICCAD-88). Digest of Technical Papers. 1988. P. 402–405.
21. Mirsky E., DeHon A. MATRIX: A Reconfigurable Computing Architecture with Configurable Instruction Distribution and Deployable Resources // IEEE Symposium on FPGAs for Custom Computing Machines (FCCM'96). Napa Valley, California, 1996. P. 157–166.
22. DSP Compiler and Processor Evaluation – DSPstone [Электронный ресурс]. Режим доступа: <http://www.ice.rwth-aachen.de/research/tools-projects/entry/detail/dspstone/>, свободный. Яз. англ. (дата обращения 10.05.2013).

Румянцев Александр Сергеевич – аспирант, Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург, Россия, darkstreamray@gmail.com

Alexander Rumyantsev – postgraduate, Saint Petersburg State University of Information Technologies, Mechanics and Optics, Saint Petersburg, Russia, darkstreamray@gmail.com