

19. Promyslov V.G., Semenov K.V., Mengazetdinov N.E. Issledovanie metodov autentifikatsii operatorov v promyshlennykh sistemakh upravleniya [Research of methods of authentication of operators in industrial control systems], *Problemy upravleniya* [Management Problems], 2022, No. 3, pp. 40-54.
20. Artamonov V.A., Artamonova E.V. Iskusstvennyy intellekt i bezopasnost': problemy, zabluzhdeniya, real'nost' i budushchee [Artificial intelligence and security: problems, delusions, reality and future], *Rossiya: tendentsii i perspektivy razvitiya* [Russia: trends and development prospects], 2022, No. 17-1, pp. 585-594.

Статью рекомендовал к опубликованию д.т.н. Л.А. Видовский.

Частикова Вера Аркадьевна – Кубанский государственный технологический университет; e-mail: chastikova_va@mail.ru; г. Краснодар, Россия; тел.: +79184635536; кафедра компьютерных технологий и информационной безопасности; к.т.н.; доцент.

Жерлицын Сергей Анатольевич – e-mail: kpytooooo@gmail.com; тел.: +79181965775; кафедра компьютерных технологий и информационной безопасности; аспирант.

Chastikova Vera Arkadyevna – Kuban State Technological University; e-mail: chastikova_va@mail.ru; Krasnodar, Russia; phone: +79184635536; the department of computer technologies and information security; cand. of eng. sc.; associate professor.

Zherlitsyn Sergey Anatolyevich – e-mail: kpytooooo@gmail.com; phone: +79181965775; the department of computer technologies and information security; graduate student.

УДК 004.021

DOI 10.18522/2311-3103-2022-4-200-212

К.Н. Алексеев, Д.А. Сорокин, А.Л. Леонтьев

МЕТОДИКА СОЗДАНИЯ ТОПОЛОГИЧЕСКИХ ОГРАНИЧЕНИЙ ПРИ ВЫСОКОЙ УТИЛИЗАЦИИ РЕСУРСОВ ПЛИС

Рассмотрена проблема достижения высокой реальной производительности реконфигурируемых вычислительных систем при решении вычислительно трудоёмких задач различных предметных областей. Величину реальной производительности реконфигурируемых систем определяют параметры выполняемых на них программ, основной компонентой которых являются вычислительные структуры обработки данных, реализованные в виде конфигурационных файлов ПЛИС. При этом одним из ключевых параметров любой вычислительной структуры является тактовая частота ее работы, которая непосредственно влияет на её производительность. Однако достижение высоких тактовых частот сопряжено с рядом проблем, которые современные средства САПР не решают. Причина кроется в неоптимальном топологическом размещении функциональных узлов вычислительной структуры на поле примитивов ПЛИС, особенно при высокой утилизации ресурсов. Это приводит к повышенной нагрузке на коммутационную матрицу ПЛИС и, как следствие, связи между примитивами ПЛИС, имеющими функциональную зависимость, оказываются значительно длиннее, чем это допустимо. Кроме того, излишняя длина связей наблюдается при трассировке соединений между примитивами, которые расположены на разных кремниевых кристаллах ПЛИС или же физически разделены встроенными периферийными устройствами. В настоящей статье описывается методика, которая позволяет рационализировать размещение элементов вычислительной структуры на поле примитивов ПЛИС, минимизировать длину трасс между примитивами, а также минимизировать число трасс между физически разделёнными топологическими областями ПЛИС. Работоспособность предложенной методики показана на примере решения тестовой задачи «КИХ-фильтр» на реконфигурируемом компьютере «Терциус». Проиллюстрированы основные проблемы при достижении целевой тактовой частоты и описан способ их преодоления. Применение методики позволило увеличить тактовую частоту и тем самым поднять производительность «Терциус» на 25% без переработки функциональной схемы вычислительной структуры задачи. Текущие исследования

эффективности предложенной методики позволяют утверждать, что автоматизированные средства создания топологических ограничений на её основе позволят существенно сократить время разработки программ с требуемыми характеристиками для реконфигурируемых вычислительных систем.

Реконфигурируемые вычислительные системы; ПЛИС; САПР; Physical Constraints; Placement Constraints; Timing Closure.

K.N. Alekseyev, D.A. Sorokin, A.L. Leontyev

METHODOLOGY OF TOPOLOGICAL RESTRICTIONS FOR INTENSIVELY USED FPGA RESOURCE

In the paper we consider the problem of achieving high real performance of reconfigurable computer systems in implementing computationally expensive tasks from various problem areas. The parameters of the programs executed on reconfigurable systems determine their real performance. The main component of these programs is the computing data processing structures implemented as FPGA configuration files. At the same time, one of the key parameters of any computing structure is the clock frequency of its operation, which directly affects its performance. However, there are several problems concerning the achievement of high clock rates, and they cannot be solved with the help of modern CAD tools. The reason is the non-optimal topological placement of functional blocks of the computing structure within the field of FPGA primitives, especially with high resource utilization. Due to this, the load on the FPGA switching matrix is increasing, and, as a result, the connections among functionally dependent FPGA primitives turn out to be much longer than is acceptable. In addition, excessive connection length is observed when tracing connections among primitives that are placed on different FPGA chips or are physically separated by on-chip peripherals. In the paper we describe a methodology which provides optimization of the placement of computing structure elements on FPGA primitives, and minimizes the length of traces among primitives, and also minimizes the number of traces among physically separated FPGA topological sections. To prove the proposed methodology, we implemented the test task "FIR-filter" on a reconfigurable computer "Tertius." We have demonstrated the main problems concerning reaching the target clock rate and have described a method for their solution. Owing to our methodology, it is possible to increase the clock rate; hence, the performance of Tertius will increase by 25% without revising the functional circuit of the task's computing structure. According to our current research of the suggested methodology and its efficiency, we claim that CAD tools, used for creating topological restrictions and based on our methodology, will significantly reduce the time for developing programs with the required characteristics for reconfigurable computer systems.

Reconfigurable computer systems; FPGA; CAD; Physical Constraints; Placement Constraints; Timing Closure.

Введение. Основной аппаратной компонентой высокопроизводительных реконфигурируемых вычислительных систем (РВС) являются ПЛИС, объединённые мощной пространственной коммутацией [1]. Именно реконфигурационные возможности современных ПЛИС позволяют соответствующим образом адаптировать архитектуру РВС и решать вычислительно-трудоемкие задачи математической физики, цифровой обработки сигналов, искусственного интеллекта и других областей с производительностью порядка 60–90% от пиковой, тогда как на большинстве задач серверы традиционной архитектуры и кластерные вычислители зачастую работают с производительностью, не превышающей 10% [1, 2].

Однако в отличие от классических вычислительных систем, программирование РВС является более сложным процессом, включающим разработку конфигурационных файлов ПЛИС (англ. *configware*), реализующих вычислительную структуру задачи, разработку программ управления контроллерами внешней памяти и программ управления процессом решения. Проектирование и отладка такой программной триады требуют высокой квалификации специалистов и значительных временных затрат [1, 2]. Конечно, процесс программирования РВС пытаются

ускорить, активно развивая высокоуровневые средства автоматизированного синтеза, однако чаще всего требуемая производительность системы не обеспечивается без низкоуровневой оптимизации вычислительной структуры решаемой задачи [3].

В основном это связано с неспособностью САПР [4, 5], применяемых для получения конфигурационных файлов, обеспечить целевые тактовые частоты при высокой утилизации ресурсов ПЛИС. Только при утилизации менее 40% размещение функциональных узлов вычислительной структуры, выполненное САПР в автоматическом режиме, позволяет проложить трассы оптимальной длины. Это подтверждается экспериментальными исследованиями зависимости тактовых частот от утилизации ресурсов ПЛИС, проведёнными при разработке конфигурационных файлов для таких вычислительно-трудоемких задач, как прогнозирование и исключение кратных волн [6, 7], обратная кинематическая задача сейсморазведки [8], LU-разложение [9], модифицированный попеременно-треугольный метод решения сеточных уравнений [10], молекулярное моделирование [11].

Повышение утилизации естественным образом сокращает число доступных вариантов трассировки соединений, удовлетворяющих требованиям к тактовой частоте. Результаты исследований свидетельствуют о настолько неоптимальном автоматическом размещении функциональных узлов вычислительной структуры с утилизацией логических ресурсов ПЛИС выше 40% и встроенных арифметических блоков и блоков памяти выше 60%, что полученная тактовая частота оказывается до двух раз меньше целевой. В связи с этим, разработчики САПР [4, 5] предлагают оптимизировать процессы топологического размещения вычислительной структуры на поле примитивов ПЛИС и трассировки соединений между ее функциональными узлами путём перебора различных стратегий (алгоритмов размещения и трассировки). Некоторые САПР также предлагают возможность автоматического определения наилучшего сочетания стратегий на основе данных машинного обучения [4], другие ориентированы на наиболее оптимальное размещение элементов функциональных узлов и трассировку связей между ними [12–15].

Однако для получения высокой реальной производительности РВС, помимо прочего, должны учитываться требования по размещению функциональных узлов вычислительной структуры относительно друг друга, а также геометрические и архитектурные особенности ПЛИС. Важно отметить, что на размещение функциональных узлов влияют не только обусловленные математическим аппаратом решаемой задачи функциональные зависимости, но и зависимости от узлов управления ходом вычислений и потоками данных. Из-за отсутствия комплексного анализа обозначенных выше требований, современные САПР не позволяют при высокой утилизации ПЛИС в автоматическом режиме получать конфигурационные файлы с необходимыми характеристиками [3].

В настоящее время удовлетворить такой совокупности требований можно лишь путем «ручного» управления процессом топологического размещения элементов вычислительной структуры на поле примитивов ПЛИС. В качестве инструментария современные САПР предлагают использовать так называемые топологические ограничения [16]. Анализ работ [14, 15, 17, 18], в которых топологические ограничения используются для достижения целевых тактовых частот, позволяет утверждать, что на сегодняшний момент не существует какой-либо методики, нацеленной на достижение реальной производительности РВС, близкой к пику.

В рамках данной статьи описана методика создания топологических ограничений ПЛИС, основанная на подходе, разработанном и используемом специалистами ООО «НИЦ СЭ и НК» [3]. Применение методики обеспечивает достижение целевых тактовых частот при высокой утилизации доступных ресурсов РВС за счет более оптимального топологического размещения вычислительной структуры

на поле примитивов ПЛИС по сравнению со всеми известными САПР. Работоспособность методики проиллюстрирована на примере решения прикладной тестовой задачи.

Методика создания топологических ограничений ПЛИС. Реальная производительность РВС напрямую зависит от степени масштабирования вычислительной структуры и тактовой частоты ее работы.

В свою очередь степень масштабирования и тактовая частота зависят от утилизации аппаратного ресурса ПЛИС. Аппаратный ресурс определяется числом примитивов (таблиц истинности LUT, триггеров FF, встроенных арифметических блоков DSP и блоков памяти BRAM), числом аппаратно-реализованных периферийных устройств (каналы ввода/вывода данных и встроенные стандартные интерфейсы). Так же к аппаратному ресурсу относится коммутационная матрица – внутренний интерконнект, расположенный по горизонтали и вертикали между примитивами ПЛИС и имеющий коммутационные узлы в местах пересечений трасс.

При высокой утилизации ресурсов ПЛИС именно длины трасс между примитивами в большей степени определяют предельную тактовую частоту вычислительной структуры. В свою очередь на длины трасс влияют как мощность коммутационной матрицы, так и технологии изготовления ПЛИС, предполагающие сопряжение нескольких кремниевых кристаллов в одном корпусе. Фирма AMD – Xilinx использует технологию Stacked Silicon Interconnect Technology для объединения больших областей реконфигурируемой логики (Super Logic Regions, SLRs) в единое поле примитивов (к примеру, семейства UltraScale+ и Versal) [19, 20]. Фирма Intel использует технологию 3D packing для сопряжения реконфигурируемой логики с НВМ памятью и аппаратно-реализованными периферийными устройствами (например, семейство Agilex) [21]. Результаты исследований подтверждают, что целевая тактовая частота не может быть достигнута в случае, если вычислительная структура использует множество трасс между примитивами, расположенными на больших расстояниях на разных кремниевых кристаллах, из-за чего реальная производительность РВС может быть ниже предельной на 20% и более. Аналогичная ситуация наблюдается при пересечении трассами областей ПЛИС с аппаратно-реализованными периферийными устройствами.

Помимо прочего, примитивы LUT и FF функциональных узлов концентрируются средствами САПР в непосредственной близости от функционально зависимых примитивов DSP и BRAM. В случае неоптимального размещения функциональных узлов на поле примитивов ПЛИС, их элементы мешают друг другу проложить трассы требуемой длины между собой. Обычно подобная ситуация наблюдается при высокой утилизации ресурсов ПЛИС, когда при размещении DSP и BRAM не учитываются функциональные зависимости между узлами вычислительной структуры и зависимости от узлов управления ходом вычислений и потоками данных.

Для повышения реальной производительности РВС при решении вычислительно-трудоемких задач предлагается нижеизложенная методика создания топологических ограничений ПЛИС. Методика направлена как на минимизацию длин трасс между примитивами за счет рационального размещения элементов функциональных узлов в заданной топологической области ПЛИС, так и на минимизацию числа трасс между топологическими областями, разделенными аппаратно-реализованными периферийными устройствами или представляющими собой отдельные кремниевые кристаллы.

Методика предполагает выполнение следующей совокупности действий:

Во-первых, необходимо разработать траекторию размещения узлов вычислительной структуры, учитывающую как функциональные зависимости между ними, так и зависимости от узлов управления ходом вычислений и потоками данных. При этом траектория должна обеспечить:

- ◆ размещение в непосредственной близости функциональных узлов вычислительной структуры, имеющих функциональную зависимость по приему и передаче данных;
- ◆ последовательное размещение функциональных узлов вычислительной структуры, относительно их зависимости от узлов управления ходом вычислений и потоками данных;
- ◆ размещение узлов, отвечающих за организацию связи реконфигурируемой логики ПЛИС с внешними устройствами, в непосредственной близости к встроенным периферийным устройствам;
- ◆ минимальное число пересечений трассами топологических областей ПЛИС, содержащих аппаратно-реализованные периферийные устройства;
- ◆ минимальное число пересечений трассами топологических областей сопряжения разных кремниевых кристаллов ПЛИС;
- ◆ исключение пересечений уже занятых топологических областей ПЛИС.

Во-вторых, необходимо разместить элементы функциональных узлов на поле примитивов ПЛИС в соответствии с разработанной траекторией:

- ◆ создать топологические ограничения для элементов вычислительной структуры, использующих примитивы аппаратно-реализованных арифметических блоков и блоков памяти ПЛИС;
- ◆ создать топологические ограничения для элементов вычислительной структуры, использующих примитивы аппаратно-реализованных периферийных устройств;
- ◆ создать топологические ограничения для элементов вычислительной структуры, размещенных в конкретных CLB.

Проведенные исследования показали, что размещение арифметических блоков и блоков памяти в соответствии с разработанной траекторией позволяет обеспечить более равномерную нагрузку на коммутационную матрицу ПЛИС за счет оптимизации длины трасс между логическими ячейками и в целом между функциональными узлами. Стоит отметить, что иногда этого достаточно для достижения целевой тактовой частоты.

Инструменты топологических ограничений имеются у каждого производителя ПЛИС. В качестве примера приведен инструментарий Physical Constraints фирмы AMD – Xilinx [16]:

- ◆ LUTNM и HLUTNM: размещение двух LUT в одной ячейке с использованием выходов O5 и O6;
- ◆ PROHIBIT: запрет на размещение элементов функциональных узлов в указанной ячейке или области ПЛИС (clock region; SLR);
- ◆ PBLOCK (Physical Block): физическое ограничение отдельной области ПЛИС для реализации в ней некоторых функциональных узлов;
- ◆ LOC: размещение элемента узла в указанной ячейке;
- ◆ BEL: размещение элемента узла в указанной части CLB;
- ◆ U_SET и HU_SET: объединение элементов в одну логическую группу;
- ◆ RLOC: расположение элементов относительно друг друга в рамках одной логической группы;
- ◆ FIXED_ROUTE: фиксация трассы между двумя конкретными ячейками.

В-третьих, необходимо уменьшить длины трасс, использующих несколько последовательно подключенных примитивов LUT (трассы с высокой логической нагрузкой, англ. logic levels):

- ◆ применить инструменты ограничения топологической области ПЛИС для размещения элементов функциональных узлов;

- ◆ создать топологические ограничения, обеспечивающие размещение элементов вычислительной структуры относительно их взаимного расположения;
- ◆ создать топологические ограничения, обеспечивающие размещение элементов вычислительной структуры в конкретных примитивах LUT и FF;
- ◆ создать топологические ограничения для фиксации конкретных траекторий трасс между используемыми примитивами ПЛИС.

Подобный вид размещения в некоторых случаях приводит также к минимизации занимаемого функциональным узлом топологического пространства ПЛИС.

В-четвертых, необходимо минимизировать число трасс между примитивами, расположенными на разных кремниевых кристаллах или разделенными аппаратно-реализованными периферийными устройствами:

- ◆ создать топологические ограничения для крупных областей ПЛИС, отталкиваясь от архитектурных особенностей конкретной микросхемы;
- ◆ определить, к какой из определенных топологических областей относится каждый функциональный узел вычислительной структуры;
- ◆ разместить элементы функциональных узлов в конкретной топологической области ПЛИС.

Если после размещения всех функциональных узлов утилизация ресурсов различных топологических областей существенно отличается, возникает неравномерная нагрузка на коммутационную матрицу ПЛИС: в топологической области с меньшей утилизацией ресурсов остается больше вариантов трассировки соединений между элементами, тогда как в физической области с большей утилизацией ресурсов – меньше вариантов. Такое неравномерное использование ресурсов в разных топологических областях может значительно влиять на результирующую тактовую частоту работы вычислительной структуры.

Для того чтобы избежать излишней нагрузки на коммутационную матрицу ПЛИС, необходимо:

- ◆ разделить крупные пограничные функциональные узлы по операциям;
- ◆ разместить пограничные функциональные узлы в обеих смежных топологических областях ПЛИС: часть операций каждого узла разместить в одной области, другую часть – в другой области.

Если доступный аппаратный ресурс топологических областей отличается, иногда рационально изменять саму структуру функционального узла, подстраивая ее под архитектуру конкретной области.

В-пятых, если это возможно, необходимо минимизировать длину трасс между примитивами, расположенными на разных кремниевых кристаллах или разделенными аппаратно-реализованными периферийными устройствами:

- ◆ изменить вычислительную структуру на границах пересечения данных топологических областей путем добавления блоков с нулевой логической нагрузкой ($\text{logic levels} = 0$): связи типа «триггер-триггер»;
- ◆ создать топологические ограничения для элементов функциональных узлов, трассы между которыми пересекают области ПЛИС с аппаратно-реализованными периферийными устройствами или области сопряжения разных кремниевых кристаллов, в непосредственной близости от границ данных топологических областей.

При этом рационально использовать архитектурные решения производителей ПЛИС: фирма AMD – Xilinx реализовала Input multiplexer (IMUX) registers – дополнительные регистры, расположенные на границе Interconnect/CLB (серия Versal) [20]; фирма Intel использует технологию HyperFlex, согласно которой реализовано множество дополнительных регистров в узлах коммутационной матрицы (серии Stratix, Agilex) [21].



Рис. 2. Размещение функциональных узлов на ПЛИС XCKU095, выполненное САПР Vivado 2020.2 при реализации тестовой задачи «КИХ-фильтр»

В соответствии с описанной выше методикой, при создании конфигурационных файлов была разработана траектория размещения по типу «змейка», которая обеспечивает: соседство узлов, имеющих функциональную зависимость или зависимость от процессов плавного запуска решения и плавного сброса; минимальное число пересечений с областями ввода/вывода данных; отсутствие пересечений занятых топологических областей ПЛИС.

Далее было выполнено размещение блоков DSP в соответствии с разработанной траекторией с помощью инструмента AMD – Xilinx Physical Constraints – LOC [16]. Оценка качества созданного набора топологических ограничений показала, что на данном этапе максимальная тактовая частота вычислительной структуры составляет 445 МГц. Визуализация размещения функциональных узлов в ПЛИС (рис. 3) косвенно свидетельствует об оптимизации нагрузки на коммутационную матрицу ПЛИС. В этом случае использовать такие инструменты как BEL, U_SET, NU_SET, RLOC и FIXED_ROUTE [16] не требуется.

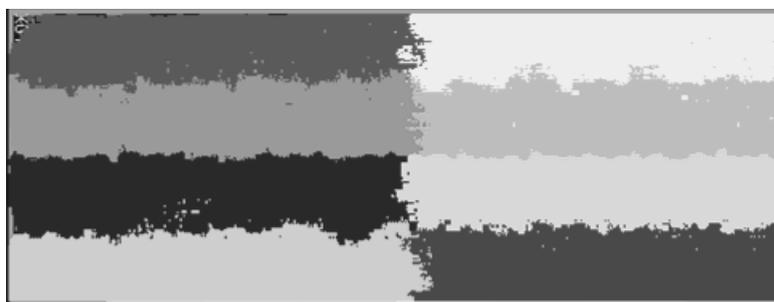


Рис. 3. Результат размещения функциональных узлов на ПЛИС XCKU095 после привязки элементов к примитивам DSP при реализации тестовой задачи «КИХ-фильтр»

Однако при попытке реализации вычислительной структуры на целевой тактовой частоте подавляющее большинство трасс, длина которых не удовлетворяет заданным требованиям, пересекает области ввода/вывода данных (рис. 4).

Для сокращения числа таких трасс поле примитивов ПЛИС с помощью инструмента PBLOCK [16] было разделено на три топологических области, ограниченных областями ввода/вывода данных. Поскольку в ПЛИС XCKU095 два центральных столбца DSP непосредственно граничат с областями ввода/вывода данных, часть операций функциональных узлов, использующих данные примитивы, была размещена в одной из смежных топологических областей, а оставшаяся – в другой.

Такое размещение обеспечило практически одинаковую утилизацию ресурсов в каждой топологической области и равномерную нагрузку на коммутационную матрицу ПЛИС. Для сокращения длины трасс между смежными топологическими областями в соответствующих функциональных узлах были реализованы дополнительные каскады с нулевой логической нагрузкой.

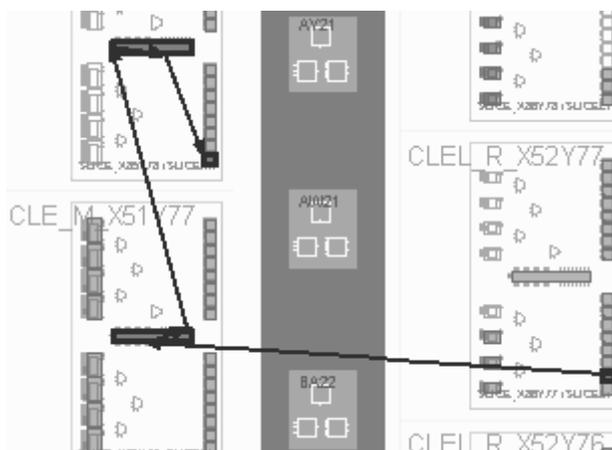


Рис. 4. Трасса, пересекающая области ввода/вывода данных при реализации тестовой задачи «КИХ-фильтр»

Характеристики конфигурационных файлов, полученных после описанных преобразований, приведены в табл. 2.

Таблица 2

Характеристики реализации тестовой задачи «КИХ-фильтр» и занимаемый аппаратный ресурс после применения разработанной методики

FF	LUT	DSP	BRAM	Frequency	Performance
780970	372755	768	16	500 МГц	702 Gflops
72.63%	69.34%	100%	0.95%		

Применение разработанной методики создания топологических ограничений ПЛИС позволило обеспечить работу вычислительной структуры тестовой задачи «КИХ-фильтр» на целевой тактовой частоте 500 МГц при высокой утилизации ресурсов ПЛИС.

Как следствие, при решении тестовой задачи, производительность РК «Терциус» увеличилась в 1.25 раза.

Помимо данной иллюстрации, представленная методика была апробирована при решении тестовой задачи «КИХ-фильтр» на других РВС. Так, без применения предложенной методики, САПР позволил достичь частоты работы 400 МГц на РВС «Сегин» [26], основным вычислительным компонентом которой являются ПЛИС Xilinx семейства Virtex UltraScale Plus. Благодаря применению методики, тактовая частота была увеличена до 525 МГц, за счет чего производительность РВС выросла в 1.31 раза.

Вместе с этим применение методики позволило увеличить производительность РВС при решении ряда других прикладных вычислительно-трудоемких задач разных предметных областей. К примеру, при решении СЛАУ методом LU разложения на РК «Терциус» [9] тактовая частота увеличена с 350 МГц до 435 МГц, за счет чего производительность РВС выросла в 1.24 раза.

Заключение. В рамках данной работы рассмотрено решение задачи многокритериальной оптимизации топологического размещения функциональных узлов вычислительной структуры на поле примитивов ПЛИС. Основными критериями, определяющими сложность данной задачи, являются высокая утилизация ресурсов ПЛИС и высокая частота работы вычислительной структуры.

Показано, что современные САПР зачастую не позволяют получать конфигурационные файлы, удовлетворяющие высоким требованиям к производительности системы. Достижение заданных характеристик системы можно обеспечить путем топологического размещения функциональных узлов вычислительной структуры на поле примитивов ПЛИС.

В результате формализации подхода, разработанного и применяемого специалистами «НИЦ СЭ и НК», была описана методика создания топологических ограничений ПЛИС, нацеленная на достижение высокой реальной производительности РВС при решении вычислительно-трудоемких задач. В отличие от существующих технологий, предлагаемых производителями САПР, разработанная методика позволяет:

- ◆ учитывать функциональные зависимости между узлами вычислительной структуры;
- ◆ учитывать зависимости от узлов управления ходом вычислений и потоками данных;
- ◆ учитывать геометрические и архитектурные особенности ПЛИС путем балансировки степени утилизации во всех топологических областях за счет адаптации функциональных узлов под ресурс области их размещения.

Автоматизация предложенной методики создания топологических ограничений ПЛИС позволит сократить время разработки параллельно-конвейерных программ для высокопроизводительных РВС при решении вычислительно-трудоемких задач. Перспективными являются разработка и применение в автоматическом режиме эвристических алгоритмов оптимального размещения функциональных узлов вычислительной структуры в заданной топологической области ПЛИС, что потенциально может привести к более значительному сокращению времени разработки прикладного программного обеспечения с требуемыми характеристиками по производительности для РВС.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Янус-К, 2003. – 380 с.
2. *Каляев И.А., Левин И.И., Семерников Е.А., Шмойлов В.И.* Реконфигурируемые мультимедийные вычислительные структуры. – 2-е изд., перераб. и доп. / под общ. ред. И.А. Каляева. – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2009. – 344 с. – ISBN 978-5-902982-61-6.
3. *Алексеев К.Н., Сорокин Д.А., Леонтьев А.Л.* Метод управления размещением элементов вычислительной структуры при максимальной утилизации ресурсов ПЛИС // XIV Всероссийская мультимедийная конференция по проблемам управления (МКПУ-2021): Матер. XIV мультимедийной конференции (Дивноморское, Геленджик, 27 сентября – 2 октября 2021 г.): в 4 т. Т. 2. Редкол.: И.А. Каляев, В.Г. Пешехонов и др. – Ростов-на-Дону; Таганрог: Изд-во ЮФУ, 2021. – ISBN 978-5-9275-3846-1. – С. 238-240.
4. AMD Xilinx Vivado Overview. – Режим доступа: <https://www.xilinx.com/products/design-tools/vivado.html> (дата обращения: 18.05.2022).
5. Intel Quartus Prime Software Suite Overview. – Режим доступа: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html> (дата обращения: 18.05.2022).
6. *Alekseev K., Levin I., Sorokin D.* Implementation of surface-related multiple prediction task on reconfigurable computer systems // Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software. – 2020. – No. 13 (1). – P. 81-94.

7. *Алексеев К.Н., Сорокин Д.А., Матросов А.Ю., Семерникова Е.Е.* Структурно-процедурная реализация алгоритма прогнозирования кратных волн на ПЛИС // Известия ЮФУ. Технические науки. – 2016. – № 12. – С. 16-28.
8. *Алексеев К.Н., Левин И.И.* Реализация обратной кинематической задачи сейморазведки для микросейсмического мониторинга на реконфигурируемых вычислительных системах в реальном масштабе времени // Известия ЮФУ. Технические науки. – 2018. – № 8 (202). – С. 221-231.
9. *Левин И.И., Пелипец А.В.* Эффективная реализация распараллеливания на реконфигурируемых системах // Вестник компьютерных и информационных технологий. – 2018. – № 8. – С. 11-16.
10. *Левин И.И., Доронченко Ю.И., Сорокин Д.А., Чистяков А.Е.* Моделирование распространения акустических волн в массивной породе с применением реконфигурируемой вычислительной системы // Нефтяное хозяйство. – 2016. – № 3. – С. 50-53.
11. *Сорокин, Д.А., Дордопуло А.И.* Методика сокращения аппаратных затрат в сложных системах при решении задач с существенно-переменной интенсивностью потоков данных // Известия ЮФУ. Технические науки. – 2012. – № 4. – С. 213-219.
12. RippleFPGA. – Режим доступа: <https://github.com/cuhk-eda/ripple-fpga> (дата обращения: 18.05.2022).
13. AMF-Placer. – Режим доступа: <https://github.com/zslwyuan/AMF-Placer> (дата обращения: 18.05.2022).
14. *Marrouff D., Shamli A., Martin T., Grewal G., and Areibi S.* A Deep-Learning Framework for Predicting Congestion during FPGA Placement // in 30th Int'l Conference on Field Programmable Logic and Applications, Sweden, September 2020. – P. 138-144.
15. *Chak-Wa Pui, Gengjie Chen, Yuzhe Ma, Evangeline F.Y. Young, and Bei Yu.* Clock-aware UltraScale FPGA placement with machine learning routability prediction // In IEEE/ACM International Conference on Computer-Aided Design (ICCAD). – 2017. – P. 915-922.
16. Vivado Design Suite User Guide: Using Constraints. – Режим доступа: https://www.xilinx.com/content/dam/xilinx/support/documentation/sw_manuals/xilinx2021_1/ug903-vivado-using-constraints.pdf (дата обращения: 18.05.2022).
17. Vivado Design Suite User Guide: Design Analysis and Closure Techniques. – Режим доступа: https://www.xilinx.com/content/dam/xilinx/support/documents/sw_manuals/xilinx2021_2/ug906-vivado-design-analysis.pdf#nameddest=xPerformingTimingAnalysis (дата обращения: 18.05.2022).
18. *NiansongZ., XiangCh., Nachiket K.* RapidLayout: Fast Hard Block Placement of FPGA-optimized Systolic Arrays using Evolutionary Algorithms. – Режим доступа: <https://arxiv.org/abs/2002.06998> (дата обращения: 18.05.2022).
19. UltraScale Architecture Configurable Logic Block. – Режим доступа: <https://docs.xilinx.com/v/u/en-US/ug574-ultrascale-clb> (дата обращения: 18.05.2022).
20. Versal Architecture and Product Data Sheet: Overview. – Режим доступа: <https://docs.xilinx.com/v/u/en-US/ds950-versal-overview> (дата обращения: 18.05.2022).
21. Intel® Agilex™ FPGAs and SoCs Device Overview. – Режим доступа: <https://www.intel.com/content/www/us/en/docs/programmable/683458/current/fpga-and-soc-device-overview.html> (дата обращения: 18.05.2022).
22. НИЦ СЭ и НК: Терциус. – Режим доступа: <http://superevm.ru/index.php?page=tertsius> (дата обращения: 20.05.2022).
23. UltraScale Architecture and Product Data Sheet: Overview. – Режим доступа: <https://docs.xilinx.com/v/u/en-US/ds890-ultrascale-overview> (дата обращения: 20.05.2022).
24. *Лайонс Р.* Цифровая обработка сигналов: пер. с англ. А.А. Бритова. – 2-е изд. – М.: БИНОМ, 2007. – 652 с.
25. *Левин И.И., Семерников Е.А.* Устойчивость конвейерных рекурсивных фильтров // «Вестник Южного научного центра Российской академии наук». – Ростов-на-Дону: Изд-во ЮНЦ РАН, 2005. – Т. 1. – В. 2. – С. 28-40.
26. НИЦ СЭ и НК: Сегин плата вычислительного модуля. – Режим доступа: <http://superevm.ru/index.php?page=segin-plata-vychislitelnogo-modulya> (дата обращения: 20.05.2022).

REFERENCES

1. *Kalyaev A.V., Levin I.I.* Modul'no-narashchivaemye mnogoprotsessornye sistemy so strukturno-protsedurnoy organizatsiyey vychisleniy [Modular-stackable multiprocessor systems with structural and procedural organization of computing]. Moscow: Yanus-K, 2003. 380 p.
2. *Kalyaev I.A., Levin I.I., Semernikov E.A., Shmoylov V.I.* Rekonfiguriruemye mul'tikonveyernye vychislitel'nye struktury [Reconfigurable multiconveyor computing structures]. 2nd ed., rev. and suppl., under the general ed. of I.A. Kalyaeva. Rostov-on-Don: Izd-vo YuNTS RAN, 2009, 344 p. ISBN 978-5-902982-61-6.
3. *Alekseev K.N., Sorokin D.A., Leont'ev A.L.* Metod upravleniya razmeshcheniem elementov vychislitel'noy struktury pri maksimal'noy utilizatsii resursov PLIS [The method of managing the placement of elements of the computing structure with maximum utilization of FPGA resources], *XIV Vserossiyskaya mul'tikonferentsiya po problemam upravleniya (MKPU-2021): Mater. XIV mul'tikonferentsii (Divnomorskoe, Gelendzhik, 27 sentyabrya – 2 oktyabrya 2021 g.)* [XIV All-Russian Multi-conference on Management Problems (MCPU-2021): Materials of the XIV multi-conference (Divnomorskoe, Gelendzhik, September 27- October 2, 2021)]: In 4 vol. Vol. 2. Ed. board: I.A. Kalyaev, V.G. Peshekhonov, etc. Rostov-on-Don; Taganrog: Izd-vo YuFU, 2021. ISBN 978-5-9275-3846-1, pp. 238-240.
4. AMD Xilinx Vivado Overview. Available at: <https://www.xilinx.com/products/design-tools/vivado.html> (accessed 18 May 2022).
5. Intel Quartus Prime Software Suite Overview. Available at: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html> (accessed 18 May 2022).
6. *Alekseev K., Levin I., Sorokin D.* Implementation of surface-related multiple prediction task on reconfigurable computer systems, *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2020, No. 13 (1), pp. 81-94.
7. *Alekseev K.N., Sorokin D.A., Matrosov A.Yu., Semernikova E.E.* Strukturno-protsedurnaya realizatsiya algoritma prognozirovaniya kratnykh voln na PLIS [Structural and procedural implementation of the algorithm for predicting multiple waves on FPGA], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2016, No. 12, pp. 16-28.
8. *Alekseev K.N., Levin I.I.* Realizatsiya obratnoy kinematicheskoy zadachi seysmorazvedki dlya mikrozeismicheskogo monitoringa na rekonfiguriruemyykh vychislitel'nykh sistemakh v real'nom masshtabe vremeni [Implementation of the inverse kinematic problem of seismic exploration for microseismic monitoring on reconfigurable computing systems in real time], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2018, No. 8 (202), pp. 221-231.
9. *Levin I.I., Pelipets A.V.* Effektivnaya realizatsiya rasparallelivaniya na rekonfiguriruemyykh sistemakh [Effective implementation of parallelization on reconfigurable systems], *Vestnik komp'yuternyykh i informatsionnykh tekhnologiy* [Bulletin of Computer and Information Technologies], 2018, No. 8, pp. 11-16.
10. *Levin I.I., Doronchenko Yu.I., Sorokin D.A., Chistyakov A.E.* Modelirovanie rasprostraneniya akusticheskikh voln v massivnoy porode s primeneniem rekonfiguriruemoy vychislitel'noy sistemy [Modeling of acoustic wave propagation in a massive rock using a reconfigurable computing system], *Neftyanoe khozyaystvo* [Oil industry], 2016, No. 3, pp. 50-53.
11. *Sorokin, D.A., Dordopulo A.I.* Metodika sokrashcheniya apparatnykh zatrat v slozhnykh sistemakh pri reshenii zadach s sushchestvenno-peremennoy intensivnost'yu potokov dannykh [Methodology for reducing hardware costs in complex systems when solving problems with significantly variable intensity of data flows], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2012, No. 4, pp. 213-219.
12. RippleFPGA. Available at: <https://github.com/cuhk-eda/ripple-fpga> (accessed 18 May 2022).
13. AMF-Placer. Available at: <https://github.com/zslwyuan/AMF-Placer> accessed 18 May 2022).
14. *Marrouff D., Shamli A., Martin T., Grewal G., and Areibi S.* A Deep-Learning Framework for Predicting Congestion during FPGA Placement, in *30th Int'l Conference on Field Programmable Logic and Applications, Sweden, September 2020*, pp. 138-144.
15. *Chak-Wa Pui, Gengjie Chen, Yuzhe Ma, Evangeline F.Y. Young, and Bei Yu.* Clock-aware UltraScale FPGA placement with machine learning routability prediction, *In IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017, pp. 915-922.

16. Vivado Design Suite User Guide: Using Constraints. Available at: https://www.xilinx.com/content/dam/xilinx/support/documentation/sw_manuals/xilinx2021_1/ug903-vivado-using-constraints.pdf (accessed 18 May 2022).
17. Vivado Design Suite User Guide: Design Analysis and Closure Techniques. Available at: https://www.xilinx.com/content/dam/xilinx/support/documents/sw_manuals/xilinx2021_2/ug906-vivado-design-analysis.pdf#nameddest=xPerformingTimingAnalysis (accessed 18 May 2022).
18. NiansongZ., XiangCh., Nachiket K. RapidLayout: Fast Hard Block Placement of FPGA-optimized Systolic Arrays using Evolutionary Algorithms. Available at: <https://arxiv.org/abs/2002.06998> (accessed 18 May 2022).
19. UltraScale Architecture Configurable Logic Block. Available at: <https://docs.xilinx.com/v/u/en-US/ug574-ultrascale-clb> (accessed 18 May 2022).
20. Versal Architecture and Product Data Sheet: Overview. Available at: <https://docs.xilinx.com/v/u/en-US/ds950-versal-overview> (accessed 18 May 2022).
21. Intel® Agilex™ FPGAs and SoCs Device Overview. Available at: <https://www.intel.com/content/www/us/en/docs/programmable/683458/current/fpga-and-soc-device-overview.html> (accessed 18 May 2022).
22. NITS SE i NK: Tertsius [SIC SE and NC: Tertius. – Access mode]. Available at: <http://superevm.ru/index.php?page=tertsius> (accessed 20 May 2022).
23. UltraScale Architecture and Product Data Sheet: Overview. Available at: <https://docs.xilinx.com/v/u/en-US/ds890-ultrascale-overview> (accessed 20 May 2022).
24. Layons R. Tsifrovaya obrabotka signalov [Digital signal processing]: transl. from the engl. by A.A. Britova. 2nd ed. Moscow: BINOM, 2007, 652 p.
25. Levin I.I., Semernikov E.A. Ustoychivost' konveyernykh rekursivnykh fil'trov [Stability of conveyor recursive filters], *Vestnik Yuzhnogo nauchnogo tsentra Rossiyskoy akademii nauk* [Bulletin of the Southern Scientific Center of the Russian Academy of Sciences]. Rostov-on-Don: Izd-vo YuNTS RAN, 2005, Vol. 1, V. 2, pp. 28-40.
26. NITS SE i NK: Segin plata vychislitel'nogo modulya [SIC SE and NC: Seguin computing module board]. Available at: <http://superevm.ru/index.php?page=segin-plata-vychislitel'nogo-modulya> (accessed 20 May 2022).

Статью рекомендовал к опубликованию к.т.н. А.Н. Самойлов.

Алексеев Кирилл Николаевич – Общество с ограниченной ответственностью "НИЦ супер-ЭВМ и нейрокомпьютеров"; e-mail: alexseev91@mail.ru; г. Таганрог, Россия; тел.: +79283536268; научный сотрудник; к.т.н.

Сорокин Дмитрий Анатольевич – e-mail: jotun@inbox.ru; тел.: +79508668253; начальник отдела; к.т.н.

Леонтьев Антон Леонидович – e-mail: leontyev_anton@mail.ru; тел.: +79185102920; программист 2 категории.

Alekseev Kirill Nikolayevich – “Supercomputers and Neurocomputers Research Center” Co Ltd; e-mail: alexseev91@mail.ru; Taganrog, Russia; phone: +79283536268; research assistant; cand. of eng. sc.

Sorokin Dmitriy Anatol'evich – e-mail: jotun@inbox.ru; phone: +79508668253; head of department; cand. of eng. sc.

Leont'ev Anton Leonidovich – e-mail: leontyev_anton@mail.ru; phone: +79185102920; 2nd category programmer.