

DOI: 10.15514/ISPRAS-2026-38(1)-2



Учет временных ограничений на этапе размещения элементов на ПЛИС

Д.Б. Шокарев, ORCID: 0009-0009-2806-1570 <Shokarev_DB@nrcki.ru>

Р.Ж. Чочаев, ORCID: 0009-0007-0304-0496 <Chochaev_RZh@nrcki.ru>

С.В. Гаврилов, ORCID: 0000-0003-0566-4482 <s00v@yandex.ru>

*НИЦ «Курчатовский институт»,
123182 Россия, Москва, пл. Академика Курчатова, д. 1.*

Аннотация. Данная статья посвящена разработке метода учета временных ограничений при оптимизации быстродействия схем на этапе размещения элементов на программируемых логических интегральных схемах (ПЛИС). Поскольку ресурсы ПЛИС ограничены, ключевой задачей разработки систем автоматизированного проектирования (САПР) для ПЛИС является оптимизация эффективности использования этих ресурсов. Решение этой задачи подразумевает разработку новых алгоритмов, позволяющих учитывать архитектурные особенности проектируемых схем. В данной работе рассматриваются основные проблемы, связанные с задачей размещения на ПЛИС с оптимизацией задержек, и существующие подходы к их решению. В данной работе предложен новый метод оценки критичности соединений для оптимизации проектируемых схем на этапе размещения с учетом пользовательских временных ограничений. Представлен алгоритм размещения на основе предложенного метода, а также результаты анализа его эффективности. Эксперименты показали, что применение предложенного метода позволяет сократить задержки критических путей в схеме на 4,32% относительно классического подхода к оценке критичности соединений.

Ключевые слова: СБИС; ПЛИС; топологический синтез; размещение.

Для цитирования: Шокарев Д.Б., Чочаев Р.Ж., Гаврилов С.В. Учет временных ограничений на этапе размещения элементов на ПЛИС. Труды ИСП РАН, том 38, вып. 1, 2026 г., стр. 17–32. DOI: 10.15514/ISPRAS-2026-38(1)-2.

Благодарности: Работа выполнена в рамках государственного задания НИЦ «Курчатовский институт».

Timing Constraints during FPGA Placement

D.B. Shokarev, ORCID: 0009-0009-2806-1570 <Shokarev_DB@nrcki.ru>
R.Z. Chochaev, ORCID: 0009-0007-0304-0496 <Chochaev_RZh@nrcki.ru>
S.V. Gavrilov, ORCID: 0000-0003-0566-4482 <s00v@yandex.ru>

NRC «Kurchatov institute»,
1, Kurchatov Square, Moscow, 123182, Russia.

Abstract. In this paper, we propose a criticality estimation method that takes into account timing constraints during the delay optimization in the FPGA placement. Since FPGA resources are limited, the optimization of the utilization efficiency of these resources is a major challenge in FPGA CAD design. It requires the development of new algorithms that take into account the architectural features of the customers designs. In this paper, existing approaches to the FPGA timing-driven placement are shown. A new method for the criticality estimation considering user-defined timing constraints and a new timing-driven placement algorithm based on this method are proposed. Experimental results show that the proposed method allows to reduce critical path delays in the circuits by 4.32% compared to the classical method.

Keywords: VLSI; FPGA; layout synthesis; placement.

For citation: Shokarev D.B., Chochaev R.Z., Gavrilov S.V. Timing constraints during FPGA placement. *Trudy ISP RAN/Proc. ISP RAS*, vol. 38, issue 1, 2026, pp. 17-32 (in Russian). DOI: 10.15514/ISPRAS-2026-38(1)-2.

Acknowledgements: The work was carried out within the state assignment of NRC «Kurchatov institute».

1. Введение

В течение последних лет российская электронная промышленность сталкивается с необходимостью импортозамещения широкого спектра микросэлектронных компонентов, одним из которых являются программируемые логические интегральные схемы (ПЛИС). Устройства данного класса позволяют эффективно решать задачи прототипирования и разработки некоторых типов электронных устройств благодаря возможности программирования логики непосредственно пользователем. В настоящее время маршрут проектирования в базе ПЛИС полностью автоматизирован и требует использования специализированных систем автоматизированного проектирования (САПР).

Для обеспечения максимальной эффективности использования ограниченных ресурсов ПЛИС в современных САПР применяются специализированные алгоритмы оптимизации, адаптированные для архитектур поддерживаемых схем. Однако алгоритмы, применяемые в коммерческих САПР, недоступны, тогда как открытые алгоритмы не адаптированы для эффективной работы с архитектурными особенностями российских ПЛИС. Поэтому возникает необходимость разработки новых методов и алгоритмов для оптимизации различных характеристик проектируемых схем, например, быстродействия.

Одним из ключевых этапов в маршруте проектирования на ПЛИС является этап размещения. От результатов данного этапа напрямую зависит как время выполнения последующих этапов, так и качество итогового решения. Для оптимизации быстродействия проектируемых схем размещение элементов необходимо проводить с учетом задержек, минимизируя критические пути. Существуют две основные проблемы, связанные с задачей размещения на ПЛИС с оптимизацией задержек. Первая из них заключается в том, что на этапе размещения отсутствует информация о реальных задержках путей, которая может быть получена только после трассировки. Это вынуждает использовать различные модели задержек, которые на основе информации о размещении элементов на ПЛИС дают адекватную оценку значения задержки. Вторая проблема заключается в необходимости многократного проведения временного анализа схемы в процессе проектирования для оценки изменения качества размещения. Это требует значительных затрат времени и накладывает ограничения на инструменты временного анализа.

Помимо этого, классические общедоступные алгоритмы оптимизации задержек на этапе размещения на ПЛИС не позволяют в полной мере учитывать пользовательские временные ограничения. В основном, они направлены на оптимизацию комбинаций общих параметров, например, минимизацию длины межсоединений и задержки пути. К данному классу можно отнести алгоритмы, основанные на методе оценки критичности цепей, представленном в VPR [1]. Недостаток данного подхода заключается в том, что вместо реально необходимых пользователю критериев используются обобщенные, не позволяющие во многих случаях получить оптимальное решение. С такой ситуацией можно столкнуться при работе со схемами с несколькими тактовыми сигналами, в том числе локальными, использовании блоков управления тактовыми сигналами (англ. clock gating), использовании ПЛИС как блока в составе устройства, что налагает различные временные ограничения на разные выходы схемы. Таким образом, для оптимизации быстродействия проектируемых схем в базисе ПЛИС необходима разработка новых методов и алгоритмов, позволяющих проектировать схемы в соответствии с пользовательскими временными ограничениями.

В данной работе представлен метод учета временных ограничений на этапе размещения элементов на ПЛИС. Также описана реализация алгоритма размещения на его основе, интегрированного в маршрут проектирования в САПР X-CAD [2]. Содержание работы организовано следующим образом. Во втором разделе рассматриваются существующие подходы к решению задачи размещения на ПЛИС с оптимизацией задержек. Третий раздел посвящен разработанному методу учета временных ограничений при анализе быстродействия на этапе размещения на ПЛИС. В нем представлен алгоритм размещения на основе представленного метода и приведена оценка его эффективности относительно реализации на основе классического подхода к оценке критичности.

2. Задача размещения на ПЛИС с оптимизацией задержек

Этап размещения элементов в маршруте проектирования на ПЛИС играет определяющую роль в формировании конечных характеристик разрабатываемого устройства: занимаемой площади, задержек межсоединений, помехоустойчивости. На данном этапе на основе синтезированного описания схемы и геометрического описания блока ПЛИС определяются позиции всех элементов проектируемой схемы, исходя из которых, в дальнейшем проводится трассировка. Поэтому качество итогового размещения можно точно оценить исключительно после проведения трассировки и анализа конечных характеристик. От качества размещения зависят как сложность трассировки, так и предельные значения характеристик схемы. Таким образом, задача размещения заключается в поиске такой конфигурации элементов, которая позволит добиться наилучших характеристик схемы и упростить последующую трассировку. Для того, чтобы добиться одновременного удовлетворения обоих требований, применяемые алгоритмы размещения должны учитывать следующие ограничения:

1. Архитектурные ограничения ПЛИС. К таким ограничениям относятся: количество элементов в группах блоков ПЛИС для иерархических схем, число доступных шин данных и синхронизации, топология соединений в случае отсутствия возможности оттрассировать любое теоретически возможное соединение.
2. Налагаемые пользователем ограничения. К таким можно отнести требования к занимаемой площади, рабочей частоте или помехоустойчивости.

Так как предельная рабочая частота конечного устройства определена заранее, одной из основных целей проектирования становится достижение данной частоты. Для этого в процессе проектирования необходимо многократно проводить оценку задержек критических путей, чтобы удостовериться в удовлетворении заданных пользователем временных ограничений. Проблема в том, что на этапе размещения отсутствует информация о задержках соединений между элементами. Поэтому для оценки задержек путей приходится применять различные приближенные модели, позволяющие получить значение задержки, зная только позиции элементов.

2.1 Статический временной анализ

Все существующие методы временного анализа подразделяются на динамические и статические [3]. Динамические методы основаны на моделировании работы всей схемы, поэтому не подходят для использования на этапе размещения, где основным критерием выбора инструмента временного анализа является минимизация временных затрат на проведение анализа. Поэтому методы статического временного анализа (СВА), в которых рассматриваются изолированные переключения отдельных элементов, становятся единственными применимыми для использования в алгоритмах оптимизации быстродействия проектируемых схем.

Методология СВА основана на построении временного графа, представляющего собой направленный взвешенный граф, отражающий структуру блока комбинационной логики синхронной СБИС. Вершины графа соответствуют элементам схемы, тогда как ребра – цепям между ними. Графы такого типа просты и наглядны, но ввиду малого количества узлов требуют использования упрощенных оценок задержек отдельных элементов. Альтернативный подход предполагает, что вершинам графа соответствуют не элементы, а их входные и выходные контакты, тогда как ребра отражают связи между ними. Именно данный подход применяется при построении временного графа в используемом в данной работе инструменте СВА OpenSTA [4].

Процесс работы с временным графом включает в себя четыре основных шага. На первом шаге для каждого комбинационного блока схемы на основе информации из библиотеки стандартных ячеек в формате Liberty происходит расчет задержек всех ребер и длительности переключений всех сигналов в каждой из вершин. После этого определяется фактическое время прибытия сигнала (англ. Actual Arrival Time, AAT), то есть временной промежуток, за который сигнал проходит путь от входа до выбранной точки. На третьем шаге, исходя из установленных временных ограничений, для каждой вершины задается требуемое время прибытия сигнала (англ. Required Arrival Time, RAT), представляющее собой максимальное допустимое время, при соответствии которому гарантируется правильное функционирование схемы. В конце вычисляется разница между требуемым и фактическим временем, представляющая собой запас по времени (англ. Slack). Отрицательное значение запаса по времени свидетельствует о нарушении временных ограничений.

Существует два подхода к проведению СВА:

1. На основе анализа путей. Данный подход подразумевает применение алгоритмов обхода графа в глубину. Сначала определяется анализируемая группа путей, после чего для каждого из них вычисляется общая задержка. Недостаток данного подхода заключается в том, что с увеличением размера схемы скорость анализа резко снижается, поскольку общее число путей увеличивается экспоненциально.
2. На основе блочного анализа. В отличие от подхода на основе анализа путей, данный подход подразумевает применение алгоритмов обхода графа в ширину. Их преимущество заключается в линейной вычислительной сложности относительно размера схемы, однако они не способны учитывать корреляции между различными путями, например, между тактовыми путями и путями данных.

В используемом в данной работе инструменте СВА OpenSTA совмещаются оба подхода. Сперва проводится блочный анализ, позволяющий определить критичность путей. Однако, в отличие от классического подхода, информация о задержках на некритических путях не удаляется, а привязывается к сигналу. Так формируются группы из одного сигнала и связанных с ним путей. После этого становится возможным провести детальный анализ выбранных путей, но без существенной потери производительности, так как нет необходимости перебирать все возможные переключения.

2.2 Модели задержек

Фундаментальной проблемой оптимизации задержек на этапе размещения является то, что точные значения задержек можно определить только после завершения этапа трассировки. Вследствие этого возникает необходимость получения приблизительных величин задержек, обеспечивающих достаточную точность. Для простых гомогенных архитектур ПЛИС, не имеющих иерархии, приемлемым решением является замена реальных задержек линейной функцией от расстояния между элементами. Тем не менее, введение иерархии приводит к тому, что такая модель дает неадекватные результаты, поскольку не способна учесть архитектурные особенности ПЛИС. В таком случае задержка будет зависеть не только от физического расстояния между логическими блоками, но и от принадлежности их различным группам блоков ПЛИС, специфики используемых элементов и особенностей реализации архитектуры конкретного кристалла.

Одним из способов преодоления указанной проблемы является разделение общей задержки на составляющие компоненты, каждый из которых рассчитывается отдельно. Например, в работе [5] предлагается следующая формула расчета задержки $D(i, j)$ ребра (i, j) между двумя логическими блоками:

$$D(i, j) = I(i, j) + P(i, j) + C(i, j),$$

где $I(i, j)$ – задержка между выходным контактом источника цепи и первым коммутационным блоком ребра, $C(i, j)$ – компенсирующая задержка, позволяющая учитывать различие задержек различных функциональных блоков, $P(i, j)$ – задержка пути, вычисляемая следующим образом:

$$P(i, j) = B(i, j) + L(i, j),$$

где $B(i, j)$ – базовая задержка, $L(i, j)$ – задержка, зависящая от количества задействованных фрагментов цепей ПЛИС, вычисляемая по следующей формуле:

$$L(i, j) = h \times \text{WireDelayH} + v \times \text{WireDelayV},$$

где WireDelayH и WireDelayV – задержки горизонтальных и вертикальных фрагментов цепей ПЛИС, h и v – количество задействованных фрагментов цепей ПЛИС по горизонтали и вертикали соответственно.

Другая реализация данного подхода была предложена в работе [6]. Предлагаемое решение заключается в замене пути эквивалентной электрической схемой и определении элморовских задержек для полученных цепочек транзисторов. Такое решение должно дать наиболее близкий к реальным значениям результат, однако требует, по сути, характеристики путей в ПЛИС, что является чрезвычайно трудоемким подготовительным процессом, который должен быть повторен для каждой новой ПЛИС.

Общим недостатком подобных моделей является то, что предсказанные ими значения будут соответствовать действительности только в том случае, если инструмент трассировки также направлен на минимизацию задержек. В противном случае оптимальные пути, полученные на этапе трассировки, могут отличаться от тех, которые ожидались на этапе размещения. Тогда, даже имея реальную информацию о задержках, может не получиться добиться наилучших характеристик схемы.

Другим типом моделей, которые уже много лет находят применение в различных инструментах, являются эвристические. Это могут быть как сложные обобщенные модели с множеством параметров, так и узкоспециализированные варианты. К первым можно отнести модель, представленную в работе [7], которая учитывает не только горизонтальное и вертикальное расстояние между элементами, но и размер схемы, перегруженность и количество контактов цепи. Примером вторых может служить модель, используемая в открытой САПР VPR [1]. В ней значения задержек получены экспериментальным путем по

результатам работы трассировщика, используемого в САПР. Такой подход позволяет одновременно учесть как все архитектурные особенности, так и особенности используемого инструмента трассировки. Однако из-за привязанности модели задержек к конкретному трассировщику изменение алгоритма трассировки приведет к необходимости времязатратного перерасчета всех значений.

Ввиду простоты реализации для использования с целевой ПЛИС была выбрана эвристическая модель задержки VPR. Из-за архитектурных особенностей целевой ПЛИС задержка между выходным контактом источника цепи и первым коммутационным блоком любого ребра временного графа всегда одинакова, поэтому была включена в задержку самих элементов. Так были выделены следующие четыре типа связи между элементами, для которых необходимо получить задержки:

1. Между элементами одной группы блоков ПЛИС;
2. Между элементами различных групп блоков ПЛИС;
3. Между входным блоком ПЛИС и элементом группы блоков ПЛИС;
4. Между выходным блоком ПЛИС и элементом группы блоков ПЛИС.

Для каждого типа связи была составлена своя таблица задержек в зависимости от расстояния между элементами по горизонтали и вертикали. Эти таблицы построены на основе статистической информации, для получения которой проводилось несколько сотен запусков трассировки с заданным заранее размещением для перебора всех возможных взаиморасположений элементов. Сперва происходила генерация файлов размещения для всех запусков, а после проводилась трассировка и сбор информации о задержках путей. Для 1 и 2 типов связи использовалась схема, включающая в себя два инвертора, а для 3 и 4 – один. Это позволило исключить возможное влияние других цепей на результат трассировки.

2.3 Учет временных ограничений на этапе размещения

Одним из основных классов алгоритмов размещения являются эвристические алгоритмы. Такие алгоритмы принимают на вход предварительно сформированное начальное размещение и оптимизируют его, последовательно приближая к наиболее оптимальному варианту. В качестве критерия оптимальности служит целевая функция, на минимизацию которой и направлен алгоритм. Но, хотя алгоритмы данного класса и позволяют достигать глобального оптимума целевой функции, при увеличении размеров проектируемых схем скорость поиска решения значительно падает ввиду резко возрастающего числа потенциальных перестановок.

Одним из классических эвристических алгоритмов размещения элементов на ПЛИС является алгоритм имитации отжига [8]. На его основе разработан алгоритм размещения, используемый в открытой САПР VPR. Для работы алгоритма проектируемая схема представляется в виде ориентированного графа $G = (V, E)$, где V – множество вершин, представляющих элементы, а E – множество направленных дуг, представляющих связи между двумя контактами элементов. Каждой дуге присваиваются два значения: вес, соответствующий длине соединения, и вес, связанный с задержкой ребра. Задача алгоритма размещения VPR сводится к минимизации суммы этих двух величин.

Изменение целевой функции ΔC при перестановке элементов вычисляется следующим образом [1]:

$$\Delta C = \lambda \cdot \frac{\Delta C_T}{Prev_C_T} + (1 - \lambda) \cdot \frac{\Delta C_W}{Prev_C_W},$$

где C_T – стоимость задержек, C_W – стоимость длины цепей, λ – балансирующий коэффициент от 0 до 1, для задания степени влияния задержек, $Prev_C_T$ и $Prev_C_W$ – значения C_T и C_W на предыдущей итерации температуры.

Стоимость длины цепей вычисляется следующим образом:

$$C_W = \sum_{i=1}^{N_{nets}} q[i] \times (bb_x(i) + bb_y(i)),$$

где N_{nets} – общее количество цепей в схеме, $bb_x(i)$, $bb_y(i)$ – размеры охватывающего прямоугольника для i -ой цепи, $q[i]$ – коэффициент для компенсации снижения оценки длины цепи из-за использования модели охватывающего прямоугольника.

Для того, чтобы определить, задержки каких ребер следует оптимизировать в первую очередь, вводится понятие критичности ребра, которая помогает оценить степень влияния каждого соединения на задержку критического пути. Минимизация задержек наиболее критичных ребер является ключевым фактором сокращения длины критического пути всей схемы.

Стоимость задержек с учетом критичности ребер вычисляется по следующим формулам:

$$C_T = \sum_{\forall i,j \in circuit} C_T(i,j),$$

$$C_T(i,j) = Delay(i,j) \cdot Crit(i,j)^\beta,$$

где $C_T(i,j)$ – это стоимость задержки ребра (i,j) , $Delay(i,j)$ – задержка ребра (i,j) , $Crit(i,j)$ – критичность ребра (i,j) , β – степенной коэффициент.

Коэффициент β вводится для того, чтобы увеличить вес критических цепей. Во время работы алгоритма данный коэффициент постепенно возрастает от начального значения 1 до максимального значения, установленного пользователем, которое по умолчанию составляет 8. Дальнейшее увеличение перестает оказывать влияние на итоговый результат.

2.4 Классический подход к расчету критичности

Классический подход к расчету критичности заключается в определении соотношения между запасом по времени для ребер и максимальной задержкой пути в схеме. Это позволяет количественно оценить, насколько задержка конкретных ребер близка к тому, чтобы ограничивать быстродействие схемы. Классическая формула расчета критичности выглядит следующим образом [1]:

$$Crit(i,j) = 1 - \frac{Slack(i,j)}{D_{max}},$$

где $Slack(i,j)$ – запас по времени для ребра (i,j) , D_{max} – максимальная задержка пути в схеме.

При использовании данной формулы расчета критичности ребер временного графа для всех путей должно быть установлено единое требуемое время прибытия сигнала, равное D_{max} для всех маршрутов в проекте. Таким образом, для всех ребер, принадлежащих путям с максимальным временем распространения сигнала, величина запаса времени $Slack(i,j)$ будет равна нулю, в то время как критичность принимает максимальное значение 1. То есть, чем ближе задержка пути к максимальной задержке в схеме, тем выше критичность всех ребер на этом пути. Такой подход эффективен для уменьшения общей задержки критического пути, однако в реальных проектах путь с максимальной задержкой не всегда оказывается критическим. Фактическая критичность путей определяется временными ограничениями, накладываемыми проектировщиком. Нередко бывает так, что требования по времени прибытия сигнала варьируются в зависимости от участка схемы. Например, в случаях, когда используются локальные тактовые сигналы или несколько глобальных, или если максимально допустимые задержки на различных выходах устройства неодинаковы и другие подобные ограничения.

В качестве примера подобной ситуации рассмотрим тестовую схему, изображенную на рис. 1. Она состоит из двух триггеров (FF1 и FF2) и трех комбинационных элементов (LE1, LE2, LE3) с задержками 4 нс, 2 нс и 2 нс соответственно. Все ребра схемы изначально имеют одинаковые задержки, равные 1 нс. Временные ограничения выглядят следующим образом:

- Максимальная задержка на выходе out – 8 нс;
- Период синхросигнала clk – 6,5 нс;
- Время предустановки триггеров – 0,5 нс.

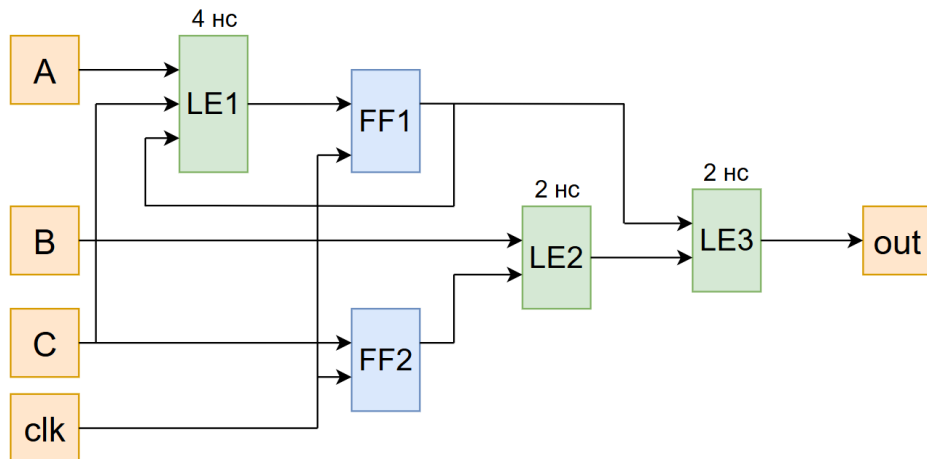


Рис. 1. Тестовая схема.
Fig. 1. Test circuit.

Для данной схемы был построен временной граф и вычислены ААТ, РАТ и запас по времени. На рис. 2 показаны два временных графа: используемый при расчете критичности ребер и получаемый при СВА с учетом заданных ограничений. На каждом из графов красным цветом выделен критический путь.

Как можно заметить, в соответствии с классической функцией критичности, веса, присваиваемые ребрам, пропорциональны задержкам соответствующих путей, однако пользовательские временные ограничения полностью игнорируются. Вследствие этого при проведении СВА с учетом вышеуказанных ограничений критические пути отличаются, причем даже если пути будут совпадать, то реальная критичность ребер может существенно отличаться от вычисляемой с помощью классической функции. Поэтому применение классической функции критичности на схемах со сложной структурой может привести к минимизации задержек не критических путей и, как следствие, к получению неоптимального результата размещения.

3. Метод учета временных ограничений на этапе размещения

Для того, чтобы решить проблему учета пользовательских временных ограничений, предлагается объединять пути с аналогичными временными ограничениями в группы для последующего анализа. Основная идея заключается в следующем: пути объединяются в группы таким образом, чтобы оценка критичности производилась исключительно в пределах каждой отдельной группы, исключая некорректные сравнения запасов по времени, относящихся к различным требуемым временам прибытия сигнала.

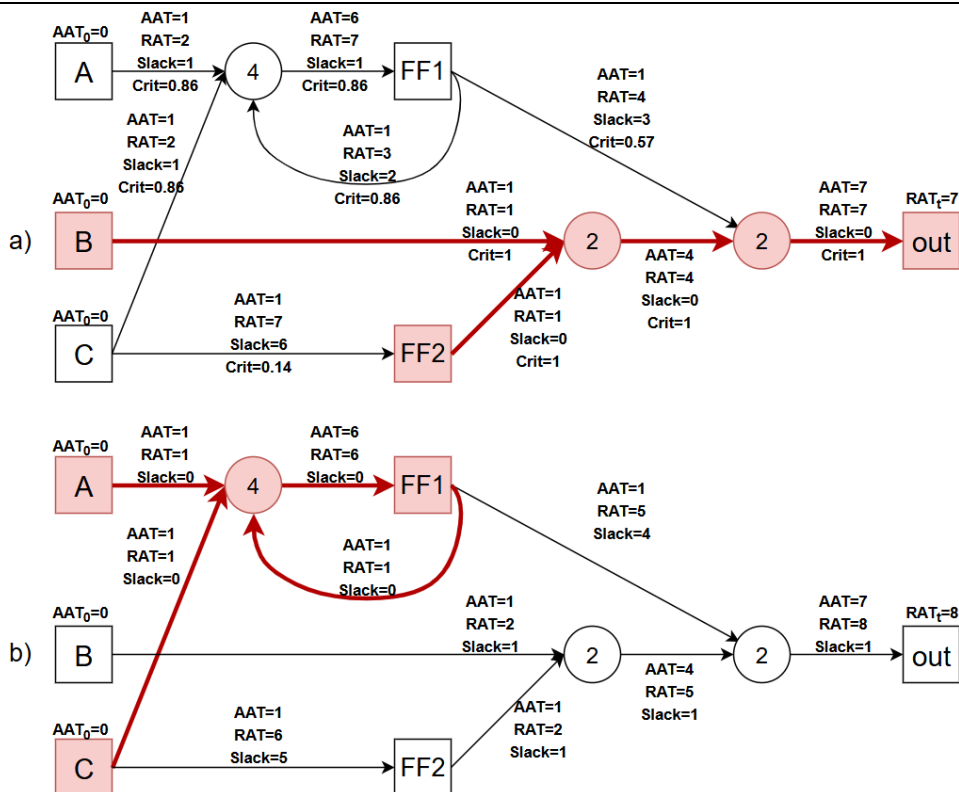


Рис. 2. (a) Пример расчета критичности по классической функции для тестовой схемы, (b) СВА с учетом заданных пользователем ограничений. RAT_t – требуемое время прибытия в терминальной точке t пути, AAT_0 – фактическое время прибытия в начальной точке пути.

Fig. 2. (a) Example of calculating criticality using the classical function for the test circuit, (b) STA taking into account user-specified constraints. RAT_t – required arrival time at the terminal point t of the path, AAT_0 – actual arrival time at the starting point of the path.

Формирование групп происходит согласно следующим двум критериям:

1. Источник ограничения – временное ограничение, определяющее требуемое время прибытия сигнала для пути. К источникам ограничений относятся период синхросигнала, а также максимально допустимая задержка на выходе схемы.
2. Источник сигнала – элемент или вход, выступающий в роли источника рассматриваемого сигнала. Им может являться входной контакт схемы или выход триггера.

Однако, если учет источника ограничения критически необходим, то деление в зависимости от источника сигнала не столь важно. Деление было введено, чтобы, используя упомянутую в разделе 2.1 особенность OpenSTA по привязке путей к сигналам, повысить быстродействие алгоритма размещения. Рассмотрим применение данного подхода на примере рассмотренной ранее тестовой схемы. Данная схема содержит семь путей, которые могут быть разделены на четыре группы, представленные в табл. 1 и на рис. 3.

Исходя из представленного деления на группы, итоговая критичность ребер формируется из двух составляющих:

- Критичности ребра в пределах собственной группы;
- Критичности группы относительно других групп.

Табл. 1. Группы путей в тестовой схеме.

Table 1. Path groups in the test circuit.

№	Путь	Источник ограничения	Источник сигнала
1	A → LE1 → FF1 B → LE1 → FF1 C → FF2	Период синхросигнала clk	Первичный вход
2	FF1 → LE1 → FF1	Период синхросигнала clk	Триггер, тактируемый сигналом clk
3	B → LE2 → LE3 → out	Первичный выход out	Первичный вход
4	FF1 → LE3 → out FF2 → LE2 → LE3 → out	Первичный выход out	Триггер, тактируемый сигналом clk

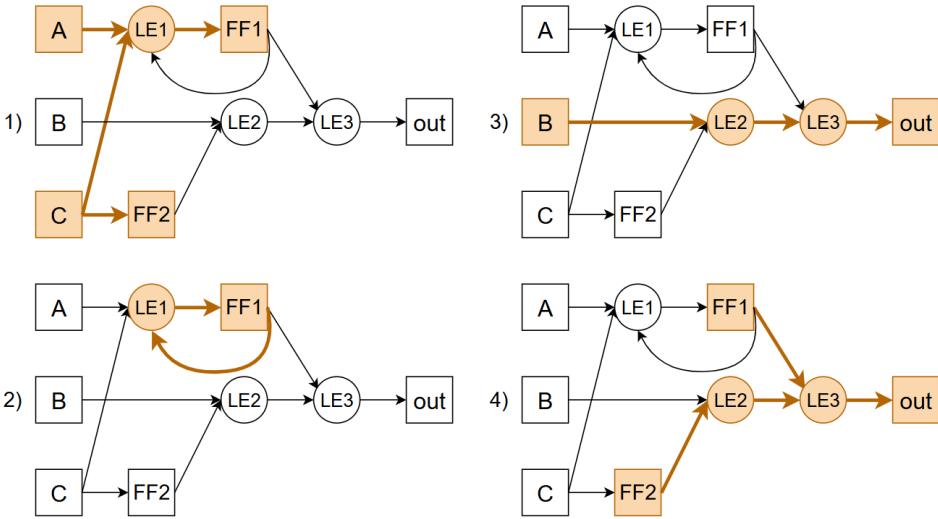


Рис. 3. Группы путей в тестовой схеме.

Fig. 3. Path groups in the test circuit.

С учетом этого, предлагается новая функция оценки критичности для ребра (i, j) :

$$Crit(i, j) = PgCrit_g \cdot \left(1 - \frac{Slack(i, j) - PgMinSlack_g}{PgMinRAT_g - PgMinSlack_g} \right),$$
$$PgCrit_g = \frac{1}{PgMinSlack_g - GlobalMinSlack + 1'}$$

где g – индекс наиболее критической группы, в которую входит ребро, $Slack(i, j)$ – запас по времени для ребра, рассчитанный для сигнала и ограничения группы g , $PgCrit_g$ – критичность группы g , $PgMinSlack_g$ – минимальный запас по времени в группе g , $PgMinRAT_g$ – минимальное требуемое время прибытия в терминальных точках путей в группе g , $GlobalMinSlack$ – минимальный запас по времени во всей схеме.

Так как через одно ребро может проходить множество путей, из-за чего оно может принадлежать одновременно нескольким группам, при расчете критичности используется наибольшее значение критичности среди всех групп, к которым оно принадлежит. Величина критичности группы путей лежит в интервале $[0, 1]$ и возрастает по мере уменьшения минимального запаса времени в группе. Общая критичность ребра получается путем

умножения критичности наиболее критичной группы, включающей это ребро, на критичность ребра в пределах данной группы. По сути, данный метод расширяет область применимости классической функции, позволяя сравнивать временные характеристики изолированных фрагментов схемы посредством введения дополнительного уровня иерархии. В качестве примера применения разработанной функции рассмотрим представленную ранее тестовую схему. Временные ограничения остаются неизменными:

- Максимальная задержка на выходе out – 8 нс;
- Период синхросигнала clk – 6,5 нс;
- Время предустановки триггеров – 0,5 нс.

На рис. 4 представлен временной граф, для которого был произведен расчет критичности с применением разработанной функции. Рассмотрим ребро, ведущее к вершине FF1, соответствующей триггеру. Согласно классической функции (рис. 2, а), критичность данного ребра составляет 0,86, поскольку она принадлежит пути с задержкой 6 наносекунд при максимальной задержке схемы 7 наносекунд. Тем не менее при выполнении СВА с учетом временных ограничений оказывается, что запас по времени для рассматриваемого ребра минимален в схеме и равен 0. Поэтому, согласно разработанной функции, его критичность принимается равной единице. Этот пример показывает преимущество предлагаемого метода перед классическим, поскольку позволяет определять степень критичности ребер с учетом пользовательских временных ограничений.

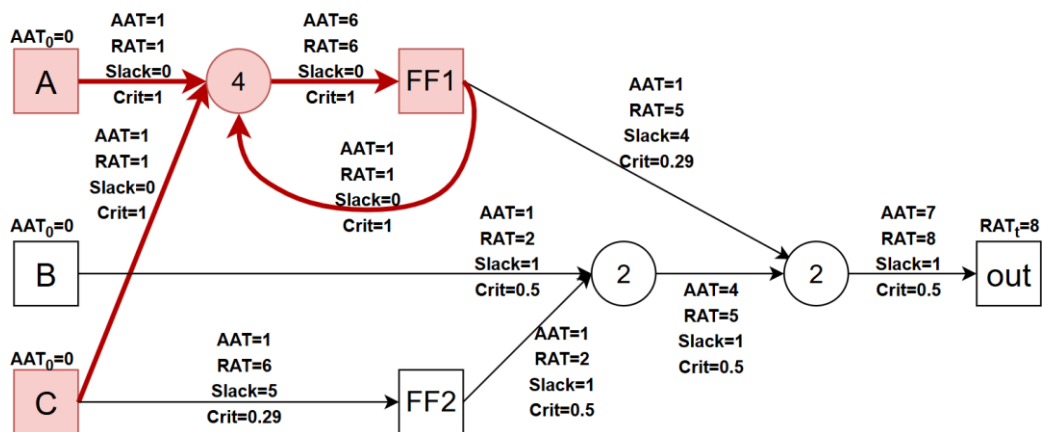


Рис. 4. Расчет критичности по разработанной функции для тестовой схемы.
Fig. 4. Calculation of criticality based on the developed function for the test circuit.

3.1 Алгоритм размещения на основе метода группировки путей

Для оценки эффективности предложенного метода учета временных ограничений разработан алгоритм размещения с оптимизацией задержек. Данный алгоритм относится к классу эвристических и базируется на рассмотренном ранее алгоритме размещения VPR. Блок-схема данного алгоритма представлена на рис. 5. Задача разработанного алгоритма состоит в минимизации следующей целевой функции:

$$C = \lambda \cdot \frac{C_T}{\text{Min}_{C_T}} + (1 - \lambda) \cdot \frac{C_W}{\text{Min}_{C_W}},$$

где C_T – стоимость задержек, C_W – стоимость длины цепей, λ – балансирующий коэффициент от 0 до 1, для задания степени влияния задержек, Min_{C_T} и Min_{C_W} – минимальные значения C_T и C_W за все время работы алгоритма.

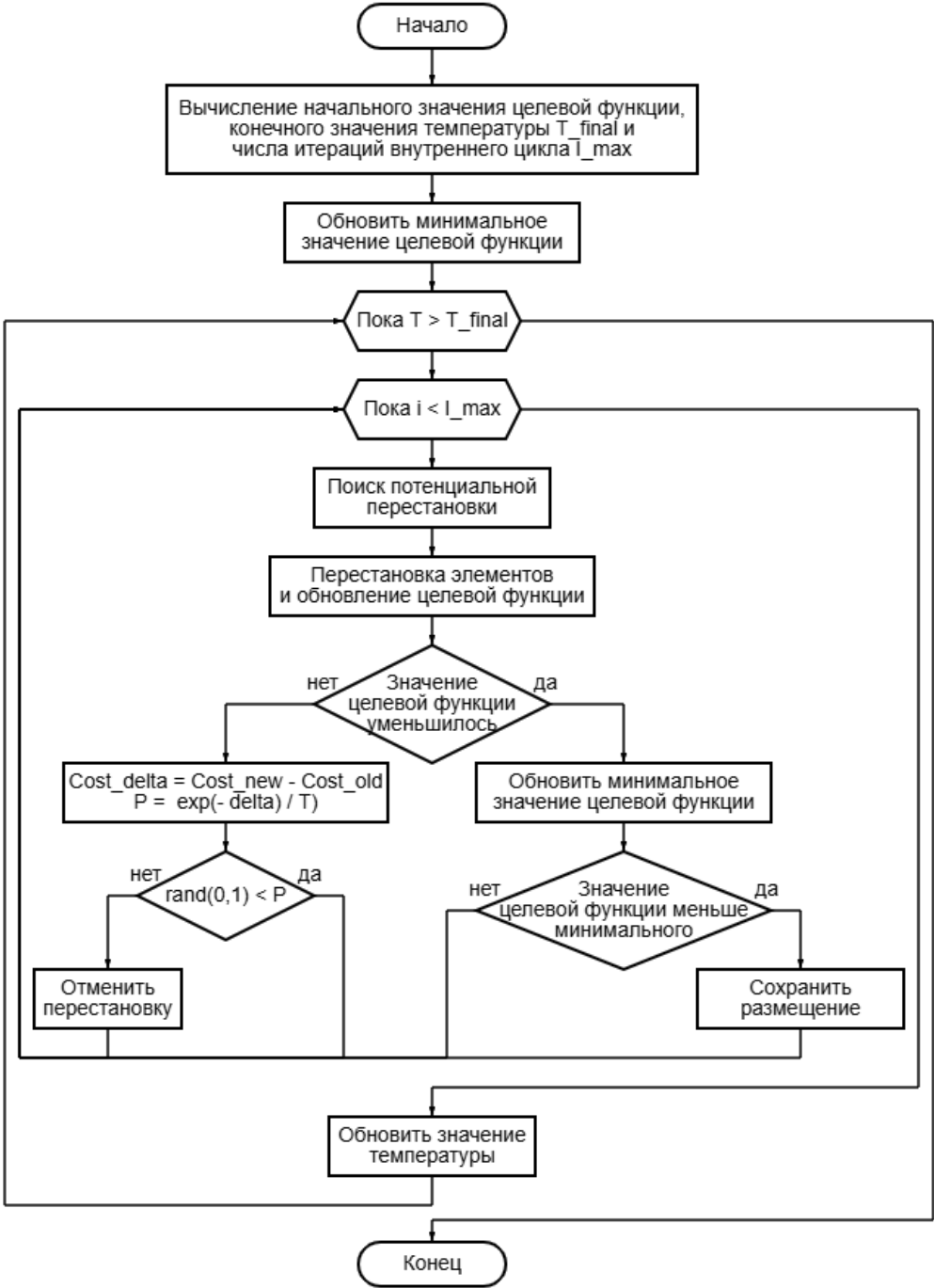


Рис. 5. Блок-схема алгоритма размещения на ПЛИС с учетом временных ограничений.

Fig. 5. Diagram of the FPGA placement algorithm taking into account timing constraints.

Функции стоимости совпадают с применяемыми в VPR, тогда как для определения задержек используются упомянутые в разделе 2.2 таблицы задержек, составленные для целевой ПЛИС.

Помимо этого, в реализации предусмотрена возможность выбора используемой функции критичности, что позволяет избежать влияния иных факторов при сравнении методов.

В качестве инструмента СВА используется OpenSTA, описанный в разделе 2.1. Данный инструмент может быть использован в качестве подключаемой библиотеки на языке C++ и встроен как часть САПР. В качестве отдельной программы OpenSTA обладает собственным Tcl интерфейсом для управления работой и загрузки временных ограничений в формате Synopsys Design Constraints (SDC) [9]. Однако при работе с ним как с библиотекой данный интерфейс отсутствует, поэтому необходима разработка собственного C++ интерфейса для работы с ним. Таким образом, для реализации алгоритма размещения с оптимизацией задержек было сделано следующее:

- инструмент СВА OpenSTA собран в виде C++ библиотеки, реализован программный интерфейс, обеспечивающий вызов функций инструмента, передачу данных из OpenSTA и загрузку временных ограничений;
- разработана реализация алгоритма размещения, в которой OpenSTA используется для задания задержек, обновления временного графа и получения временных характеристик схемы.

Поскольку любая перестановка элементов влияет на задержки путей, проходящих через перемещенные элементы, временные характеристики, полученные в ходе предыдущих расчетов, утрачивают свою актуальность. Поэтому возникает проблема выбора оптимальной частоты проведения СВА. Исследования разработчиков САПР VPR показали, что выполнение СВА чаще, чем изменяется температура, практически не улучшает качество итогового решения, но значительно увеличивает затраты по времени [1]. Эксперименты на целевой ПЛИС показали, что данное утверждение справедливо, и более частое проведение СВА даст выигрыш в среднем до 2%, но многократно увеличивает время работы алгоритма. По этой причине СВА выполняется только при каждом обновлении температуры.

3.2 Оценка эффективности

Реализация разработанного алгоритма размещения с учетом временных ограничений была протестирована на схемах различной размерности, включая схемы из наборов ISCAS-85 [10], ISCAS-89 [11] и IWLS 2005 [12]. Проведен анализ задержек критических путей для трех вариантов размещения:

- без учета временных ограничений ($\lambda=0$);
- с учетом временных ограничений и стандартной функцией критичности;
- с учетом временных ограничений и разработанной функцией критичности.

Для экспериментов с двумя последними вариантами размещения коэффициент λ устанавливался равным 0,7, поскольку именно такое значение позволило получить наилучшие результаты на исследуемом диапазоне значений от 0 до 1 с шагом 0,1. Значения задержек для сравнения реализаций алгоритма были взяты для наиболее критического пути в схеме после прохождения всего маршрута. Результаты тестирования представлены в табл. 2.

Эксперименты показали, что применение разработанной функции критичности, учитывающей пользовательские временные ограничения, обеспечило сокращение длины критического пути примерно в среднем на 5,6%. Тогда как классическая функция дала сокращение только на 1,28%. Вместе с тем в отдельных случаях наблюдается существенный рост задержки критического пути при использовании разработанного метода. Это обусловлено тем, что в используемом трассировщике [2] нет возможности выполнять трассировку цепей с оптимизацией задержек. Поэтому трассировка выполнялась с

оптимизацией длины цепей, что в ряде случаев негативно сказывается на конечных значениях задержек.

Табл. 2. Результаты тестирования.

Table 2. Testing results.

		Без оптими- зации	Оптимизация с классической функцией критичности		Оптимизация с разработанной функцией критичности	
Схема	Кол-во элемен- тов	Задержка, нс	Задержка, нс	Δ, %	Задержка, нс	Δ, %
ehl_spi_ssr	16	4,12	3,93	-4,45	3,64	-11,47
s382	49	8,33	7,49	-10,05	7,51	-9,77
s526	50	6,08	5,51	-9,37	5,41	-11,04
GRAY_24_BIT	54	3,75	3,76	0,08	3,75	0,00
synchronizer	64	4,11	4,11	0,00	4,11	0,00
ehl_uart_rx	74	7,88	8,03	1,95	8,35	6,04
s713	77	18,17	17,77	-2,19	16,33	-10,12
s641	79	17,06	16,69	-2,17	14,75	-13,52
c1355	85	16,64	15,32	-7,90	14,51	-12,77
c499	86	16,23	15,75	-2,96	15,09	-7,01
b13	102	3,85	3,84	-0,09	3,84	-0,09
s832	127	3,69	3,71	0,51	3,69	0,01
s820	128	3,69	3,69	-0,01	3,73	0,83
timer_module	146	8,87	7,89	-11,04	7,57	-14,64
arbiter	164	57,62	56,78	-1,46	56,57	-1,82
b04	190	3,42	3,89	13,71	3,42	0,05
s1423	206	3,80	3,80	-0,09	3,81	0,26
s1238	240	3,73	3,73	-0,22	3,73	-0,18
b05	262	3,78	3,75	-0,88	3,74	-0,98
srio_arb_tx_mux	268	23,94	26,86	12,22	24,48	2,25
s1488	280	3,72	3,72	0,00	3,22	-13,44
simple_spi	304	9,26	8,37	-9,56	8,86	-4,29
c3540	418	28,12	29,72	5,66	24,08	-14,38
fifo	429	19,65	21,27	8,23	21,08	7,27
i2c_master_top	448	13,48	16,00	18,74	12,86	-4,62
b12	514	3,88	3,83	-1,30	3,88	0,07
test_2	633	12,38	9,44	-23,76	9,86	-20,35
info_handler_top	675	17,41	16,47	-5,42	13,91	-20,10
decimator3_1_1	954	38,38	36,38	-5,20	35,03	-8,73
Средние значения:				-1,28		-5,60

4. Заключение

Учет временных ограничений на этапе размещения элементов на ПЛИС позволяет повысить быстродействие проектируемых схем за счет увеличения эффективности использования ограниченных ресурсов ПЛИС. В данной работе были рассмотрены основные проблемы, возникающие при решении задачи размещения на ПЛИС с оптимизацией задержек. Среди них: выбор подходящего инструмента временного анализа и формирование подходящей модели задержек. Также рассмотрены классические подходы к решению задачи размещения и представлен метод группировки путей, позволяющий учитывать пользовательские временные ограничения при оптимизации быстродействия проектируемых схем.

На основе представленного метода разработан алгоритм размещения с оптимизацией задержек, проведено его тестирование и произведена оценка эффективности предложенного метода. Как показали эксперименты, представленный метод позволил сократить длину критических путей в среднем на 5,6%, что на 4,32% больше, чем позволяет достичь классический подход. Так, метод группировки путей расширяет возможности в оценке критичности цепей, тем самым совершенствуя основу, на которой базируется множество существующих алгоритмов. Также, поскольку представленный метод, в отличие от классического подхода, не требует каких-либо вмешательств в работу инструмента СВА или специальных функций, то позволяет использовать любые инструменты СВА, в том числе и открытые.

Список литературы / References

- [1]. Marquardt A., Betz V., Rose J. Timing-driven placement for FPGAs / Proceedings of the 2000 ACM/SIGDA eighth international symposium on Field programmable gate arrays, 2000, pp. 203-213. DOI: 10.1145/329166.329208.
- [2]. Гаврилов С.В., Железников Д.А., Заплетина М.А., Тиунов И.В., Хватов В.М., Чочаев Р.Ж., Шокарев Д.Б. Разработка доверенных средств проектирования ИС в базе гетерогенных ПЛИС. Труды ИСП РАН, том 35, вып. 5, 2023 г., стр. 107–126. DOI: 10.15514/ISPRAS-2023-35(5)-8. / Gavrilov S.V., Zheleznikov D.A., Zapletina M.A., Tiunov I.V., Khvatov V.M., Chochaev R.Z., Shokarev D.B. Development of the Trusted Tools for IC Design on Heterogeneous FPGAs. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue. 5, 2023, pp. 107–126 (in Russian). DOI: 10.15514/ISPRAS-2023-35(5)-8.
- [3]. Forzan C., Pandini D. Statistical static timing analysis: A survey / Integration, 2009, T. 42, №. 3, pp. 409-435. DOI: 10.1016/j.vlsi.2008.10.002.
- [4]. Ajayi T., Blaauw D. OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain / Proc. Government Microcircuit Applications and Critical Technology Conference, 2019, pp. 1105-1110.
- [5]. Lin Z., Xie Y., Qian G., Chen J., Wang S., Yu J., Chang Y.W. Timing-driven placement for FPGAs with heterogeneous architectures and clock constraints / 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2021, pp. 1564-1569. DOI: 10.23919/DATE51398.2021.9474054.
- [6]. Hung E., Wilton S. J. E., Yu H., Chau T. C. P., Leong. P. H. W. A detailed delay path model for FPGAs / 2009 International Conference on Field-Programmable Technology, IEEE, 2009, pp. 96-103. DOI: 10.1109/FPT.2009.5377673.
- [7]. Karnik T., Kang S.M. An empirical model for accurate estimation of routing delay in FPGAs / Proceedings of IEEE International Conference on Computer Aided Design (ICCAD), IEEE, 1995, pp. 328-331. DOI: 10.1109/ICCAD.1995.480136.
- [8]. Ghannadi P., Kourhli S.S., Mirjalili S. A review of the application of the simulated annealing algorithm in structural health monitoring (1995-2021) / Frattura ed Integrità Strutturale, 2023, T. 17, № 64, pp. 51-76. DOI: 10.3221/IGF-ESIS.64.04.
- [9]. Using the Synopsys Design Constraints Format Application Note, Version 2.1 / Mountain View, CA, USA, 2017.
- [10]. Brglez F., Fujiwara H. A neutral netlist of 10 combinational benchmark circuits and a target simulator in Fortran / International Symposium on Circuits and Systems, 1985, 1985, pp. 695-698.

- [11]. Brglez F., Bryan D., Kozminski K. Combinational profiles of sequential benchmark circuits / 1989 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 1989, pp. 1929-1934. DOI: 10.1109/ISCAS.1989.100747.
- [12]. Albrecht C. IWLS 2005 benchmarks / International Workshop for Logic Synthesis (IWLS), 2005, т. 9.

Информация об авторах / Information about authors

Дмитрий Борисович ШОКАРЕВ – инженер-исследователь отдела САПР ОППМ ЦПМ НИЦ «Курчатовский институт». Область научных интересов: автоматизация проектирования микроэлектроники, САПР ИС, ПЛИС, РСнК.

Dmitry Borisovich SHOKAREV – Research Engineer of Department of Design Problems in Microelectronics of CPM of NRC «Kurchatov institute». Research interests: electronic design automation, VLSI computer-aided design, FPGA, SoC FPGA.

Рустам Жамболатович ЧОЧАЕВ – научный сотрудник отдела САПР ОППМ ЦПМ НИЦ «Курчатовский институт». Область научных интересов: автоматизация проектирования микроэлектроники, САПР ИС, ПЛИС, РСнК.

Rustam Zhambolatovich CHOCHAEV – Research Associate of Department of Design Problems in Microelectronics of CPM of NRC «Kurchatov institute». Research interests: electronic design automation, VLSI computer-aided design, FPGA, SoC FPGA.

Сергей Витальевич ГАВРИЛОВ – доктор технических наук, профессор, руководитель отделения проблем проектирования в микроэлектронике ЦПМ НИЦ «Курчатовский институт». Область научных интересов: автоматизация проектирования микроэлектроники, САПР ИС.

Sergey Vitalievich GAVRILOV – Dr. Sci. (Tech.), Prof., Head of Department of Design Problems in Microelectronics of CPM of NRC «Kurchatov institute». Research interests: electronic design automation, VLSI computer-aided design.