

# Методы ускорения работы модифицированного алгоритма трассировки Pathfinder для ПЛИС островного типа

М. А. Заплетина

Институт проблем проектирования в микроэлектронике РАН, zapletina\_m@ippm.ru

**Аннотация** — В статье представлены два метода ускорения этапа трассировки межсоединений в рамках маршрута топологического проектирования на основе программируемых логических интегральных схем с архитектурой островного типа. Базовый алгоритм, используемый в работе, представляет собой алгоритм Pathfinder, модифицированный для поддержки представления трассировочных ресурсов базового кристалла в форме смешанного графа. Первый метод, основанный на уменьшении числа перестроений трассировочных деревьев, позволяет добиться среднего ускорения времени трассировки 38% при среднем увеличении задержки критического пути 5%. Второй метод, развивающий идею направленного поиска пути на графе, при корректном выборе подстроечного коэффициента способен обеспечить двукратное ускорение трассировки без потери трассируемости.

**Ключевые слова** — трассировка, межсоединение, ПЛИС, разрыв и перетрассировка, размещение и трассировка, направленный поиск, Pathfinder.

## I. ВВЕДЕНИЕ

Трассировка является заключительным и, как правило, наиболее длительным этапом в маршруте проектирования на основе программируемых логических интегральных схем (ПЛИС) и реконфигурируемых систем на кристалле. Назначение этих классов схем (например, прототипирование заказных ИС) предполагает возможность их быстрой настройки на выполнение новых функций. В связи с этим, на время, затрачиваемое на каждый этап маршрута проектирования, накладываются жесткие ограничения. Уменьшение технологических норм ведет к повышению степени интеграции базовых кристаллов ПЛИС и в конечном итоге к росту размерности  $NP$ -полной [1] задачи трассировки межсоединений. В связи с этим, разработка новых эффективных алгоритмов трассировки не теряет актуальности.

Анализ существующих методов и способов, применяемых на этапе трассировки ПЛИС, позволяет выделить три основных направления их развития. Первое из них включает в себя разработку **новых** математических основ, абстракций для формализации и решения задачи трассировки. Ко второму направлению относятся традиционные методы поиска кратчайшего пути и эвристические методы трассировки на их основе. Третье направление подразумевает обязательное наличие аппаратной реализации предлагаемого

алгоритма трассировки на основе программируемой или заказной логической схемы или специализированного вычислителя.

В авангарде первого направления стоят методы машинного обучения, набирающие популярность в области автоматизации проектирования со второй половины 2010-х гг. Наиболее часто их применение связано с оценкой разводимости проектной схемы во время этапа размещения или по его завершению перед началом трассировки [2], [3]. Вместе с тем, самые последние работы затрагивают и непосредственное решение задачи трассировки межсоединений. Так, авторы [4] предлагают алгоритм трассировки с устранением перегрузок коммутационных элементов путем обучения с подкреплением [4] как замену традиционной эвристике известного алгоритма Pathfinder [5]. Из более ранних исследований, к первому направлению следует отнести методы целочисленного линейного программирования [6], а также методы доказательства выполнимости булевых формул, применяемые при поиске корректного трассировочного решения в детальной трассировке [7], назначении цепей синхросигналов [8], [9] и при оценке разводимости размещения [10].

Второе направление, методы которого применяются наиболее часто, в настоящее время является также и самым развитым. В его состав входят классические методы поиска кратчайшего пути на графе (алгоритм Дейкстры [11], волновой [12] (или алгоритм Ли [13]),  $A^*$  [14]) для одной цепи, а также эвристические алгоритмы, позволяющие находить пути для набора цепей. Наиболее известным из этих алгоритмов является Pathfinder [5] и многочисленные его модификации [15]-[17], в том числе версии с использованием параллелизма [18]-[21]. Известно о его применении в составе академических [16], [22] и коммерческих [23], [24] САПР.

Третье направление, как отмечено выше, включает методы и подходы к решению задачи трассировки, неотъемлемой частью которых является аппаратная реализация в виде самостоятельного устройства либо в виде программы для специализированных вычислителей. Так, в работе [25] предложен аппаратный трассировщик на платформе Intel DE1-SoC с процессором ARM Cortex A9, а в [26], [27] программы трассировки разработаны для запуска на графическом процессоре (GPU).

Данная работа развивает второе из обозначенных направлений. Базовым алгоритмом представленного далее метода трассировки является известный алгоритм Pathfinder [5], модифицированный для работы на смешанном графе коммутационных ресурсов [28]. Предшествующие работы [15]-[18], [29] свидетельствуют в пользу большого потенциала идеи развития классических методов трассировки без кардинальной смены подхода. Так, благодаря внедрению ряда модификаций в классический алгоритм Pathfinder, в [15] суммарное время трассировки было сокращено в 1,3-1,8 раза, в [18] - в 3,5 раза в среднем, что является хорошим результатом.

Использование техники перетрассировки, предложенной в данной работе, по сравнению с [29], позволило в среднем сократить время трассировки на 38% для ПЛИС с островной архитектурой. Кроме того, представленные в работе варианты реализации направленного поиска в составе эвристического алгоритма трассировки [29] при сохранении полной трассируемости позволяют сократить время трассировки более чем в 2 раза.

Идея применения направленного поиска в эвристических алгоритмах трассировки с устранением перегруженности не нова [16]. В то же время, работы, посвященные специфике этой техники для поиска пути в составе алгоритма Pathfinder на смешанном графе трассировки, не были найдены.

Дальнейшее содержимое статьи организовано следующим образом. Раздел II посвящен краткому описанию математической модели трассировочных ресурсов, используемой для формализации задачи трассировки. В разделе III даны обновленный механизм разрыва и перетрассировки и его сравнение с предыдущей работой [29]. Раздел IV посвящен деталям и результатам реализации направленного поиска в рамках эвристического алгоритма трассировки на основе Pathfinder. В разделе V представлено заключение.

## II. МОДЕЛЬ ОПИСАНИЯ ТРАССИРОВОЧНЫХ РЕСУРСОВ

В рамках данной работы для описания доступных для трассировки ресурсов базового кристалла ПЛИС (или реконфигурируемой СнК) использовалась модель смешанного графа трассировочных ресурсов [29]  $G = \{\mathbf{V}, \mathbf{E}\}$ ,  $\mathbf{E} = \mathbf{A} \cup \mathbf{U}$ , где вершина  $v_i \in \mathbf{V}$  представляет электрический узел базового кристалла, каждая дуга  $a_j = (v_k, v_l)$ ,  $a_j \in \mathbf{A}$ ,  $\mathbf{A} = \mathbf{V} \times \mathbf{V}$ ,  $v_k, v_l \in \mathbf{V}$  - направленное программируемое соединение между двумя узлами, а ребра  $u_j \in \mathbf{U}$  представляют множество ненаправленных коммутационных элементов кристалла. Далее для краткости элементы  $e_j \in \mathbf{E}$  будут именоваться ребрами, независимо от направления распространения сигнала.

Для каждой вершины  $v_i$  и каждого ребра  $e_j$  графа трассировочных ресурсов задаются фиксированные начальные веса  $w(v_i)$  и  $w(e_j)$ . Кроме того, каждому

ребру  $e_j$  ставится в соответствие функция прохождения сигнала  $F_{e_j}$ , позволяющая описать возможное направление его распространения, инверсии входов и выходов соответствующего ребру схемотехнического элемента, а также учесть различные способы управления открытием элемента (коммутацию по нагрузке, общий управляющий сигнал для группы коммутационных элементов и пр.).

В рамках процедур размещения и трассировки проектная цепь в приведенных терминах описывается как  $n_m = \{s_m, \{t_{m,1}, \dots, t_{m,l}\}\}$ , где  $s_m \in \mathbf{V}$  - источник, а  $\mathbf{T} = \{t_m^k\}$ ,  $\mathbf{T} \subset \mathbf{V}$ ,  $k = 1, \dots, l$  - множество приемников, а  $l = |\mathbf{T}|$  - число приемников сигнала в цепи  $n_m$ . Цепь  $n_m \in \mathbf{N}$  из множества проектных цепей  $\mathbf{N}$  является трассируемой, если в ходе процедуры трассировки на графе  $G$  для неё может быть найдено дерево трассировки  $\mathbf{RT}(n_m)$  с корневым элементом в вершине  $s_m$  и листьями в вершинах  $t_m^1, \dots, t_m^l$ , не пересекающиеся с деревьями других проектных цепей. Таким образом, трассируемость (или разводимость)  $R$  пользовательской схемы  $ckt$  в базе ПЛИС определяется следующим образом (1):

$$R(ckt) = \begin{cases} 1, & \forall n_i \in \mathbf{N}, \forall n_j \in \mathbf{N} : \mathbf{RT}(n_i) \cap \mathbf{RT}(n_j) = \emptyset \\ 0 & \end{cases} \quad (1)$$

## III. ОБНОВЛЕННЫЙ МЕТОД РАЗРЫВА И ТРАССИРОВКИ

Ранее в работе [29] был представлен метод разрыва и перетрассировки, призванный сократить число очисток и перестроений трассировочных деревьев проектных цепей, выполняемых в ходе устранения перегруженных участков трассировочных ресурсов. Это было реализовано, во-первых, за счет перетрассировки только цепей  $n_m \in \mathbf{M}$ , для которых выполняется условие перегруженности (2):

$$\begin{aligned} & \forall n_m \in \mathbf{M} : \\ & \exists n_k \in \mathbf{M}, \mathbf{RT}(n_m) \cap \mathbf{RT}(n_k) = \\ & = \{\mathbf{V}_{mk} \cup \mathbf{E}_{mk}, \mathbf{V}_{mk} \in \mathbf{V}, \mathbf{E}_{mk} \in \mathbf{E}\} \neq \emptyset. \end{aligned} \quad (2)$$

Во-вторых, в каждой паре перегруженных цепей перетрассировке подвергалась только цепь с меньшим количеством приемников (3):

$$\begin{aligned} & \forall (n_k, n_m), \mathbf{RT}(n_m) \cap \mathbf{RT}(n_k) \neq \emptyset : \\ & \begin{cases} \mathbf{RT}^{i+1}(n_k) = reroute(\mathbf{RT}^i(n_k)) \\ \mathbf{RT}^{i+1}(n_m) = \mathbf{RT}^i(n_m) \end{cases} \Leftrightarrow |T_k| < |T_m|, \end{aligned} \quad (3)$$

где  $i$  - номер итерации алгоритма трассировки.

В настоящей работе исследовалась модификация этого метода, при которой условие (3) было устранено в пользу перетрассировки всех цепей, имеющих общие трассировочные ресурсы, независимо от числа приемников сигнала. Согласно данным табл.1, при сохранении среднего размера дерева трассировки эта

модификация приводит к уменьшению среднего времени трассировки и росту среднего значения задержки критического пути. Данные табл. 1 были получены при прохождении маршрута проектирования

для наборов тестовых схем ISCAS'85, ISCAS'89 и LGSynth'89 на основе ПЛИС островного типа (программируемая цифровая часть ПАЦИС 5400ТР094 [30]).

Таблица 1

Результаты применения модификации метода разрыва и перетрассировки относительно (3)

| Группа тестовых схем        | Уменьшение времени трассировки, % | Уменьшение дерева трассировки, % | Увеличение задержки критического пути, % |
|-----------------------------|-----------------------------------|----------------------------------|--|
| $ T_m _{\max} \leq 20$      | 4.23                              | 1.72                             | 13.71                                    |
| $ T_m _{\max} \in [21; 60]$ | 14.76                             | 1.62                             | 10.61                                    |
| $ T_m _{\max} > 60$         | 20.965                            | 1.74                             | 6.46                                     |

Таблица 2

Результаты применения модификации механизма разрыва и перетрассировки относительно стандартной трассировки

| Группа тестовых схем        | Уменьшение времени трассировки, % | Уменьшение дерева трассировки, % | Увеличение задержки критического пути, % |
|-----------------------------|-----------------------------------|----------------------------------|--|
| $ T_m _{\max} \leq 20$      | 38.03                             | -0.01                            | -8.6                                     |
| $ T_m _{\max} \in [21; 60]$ | 37.65                             | 0.76                             | -5.38                                    |
| $ T_m _{\max} > 60$         | 32.53                             | -0.4                             | -9.85                                    |

Табл. 1 демонстрирует различия во влиянии модифицированного механизма (3) на трассировочное решение для схем, в состав которых входят проектные цепи разной степени разветвленности. Полученную закономерность можно объяснить следующим образом. Запрет (3) фактически означает фиксирование принадлежности определенного набора трассировочных ресурсов цепям со средним ( $|T_m|_{\max} \in [21; 60]$ ) и большим ( $|T_m|_{\max} > 60$ ) числом приемников. Это ведет к формированию деревьев трассировки большей высоты для менее разветвленных цепей, поскольку ресурсы, составляющие их кратчайший путь, оказываются заняты. При наличии средне и сильно разветвленных цепей в составе пользовательского проекта значительная часть конфликтов, как показывает практика, может быть устранена перетрассировкой этих цепей, и тогда снятие запрета (3) ведет к ускорению решения задачи трассировки и снижению задержки критического пути. Вместе с тем, на трассировку схем, в состав которых входят только слабо разветвленные цепи, смена механизма не оказывает значительного влияния.

С учетом приведенных данных (табл. 1, 2), можно сделать вывод о разнице в назначении метода перетрассировки [29] и его модификации, представленной в данной работе. Так, метод из работы [29] следует применять при необходимости ускорения процедуры трассировки с наименьшими потерями во временных характеристиках проектной схемы. В то время как модификация метода позволяет получить трассировочное решение менее высокого качества, но за более короткое время.

#### IV. НАПРАВЛЕННЫЙ ПОИСК

Использование направленного поиска является классическим приемом для ускорения поиска пути на базовой сети или графе при наличии информации об их пространственных характеристиках. В данной работе такими характеристиками стали прямоугольные координаты центров трассировочных элементов и их входов/выходов, приведенные в соответствие ребрам  $e_j \in \mathbf{E}$  и вершинам  $v_i \in \mathbf{V}$  трассировочного графа  $G$ .

Предложенный вариант направленного поиска предполагает введение добавочного веса вершины графа, зависящего от подстроечного коэффициента  $\alpha$  и нормализованной оценки расстояния от рассматриваемой вершины до ближайшего приемника сигнала, путь до которого ещё не найден. Так, стандартная формула для расчета стоимости вхождения вершины  $v_i \in \mathbf{V}$  в дерево трассировки  $\mathbf{RT}(n_m)$  из [29] преобразуется в вид (4):

$$\text{cost}(v_i) = \text{cost}_{\text{hist}}^i + c(v_i) + w(e_i) + \alpha \text{Cost} , \quad (4)$$

где  $\text{cost}_{\text{hist}}^i$  - накопленный вес вершин, предшествующих  $v_i$  в строящемся дереве трассировки  $\mathbf{RT}(n_m)$ ;  $c(v_i)$  - обновляемый вес вершины, зависящий от текущей итерации трассировки  $iterN$  и числа проектных цепей, перегружающих вершину  $v_i$ ;  $\alpha$  - подстроечный коэффициент;  $\text{Cost}$  - нормализованное Манхэттенское расстояние от анализируемой вершины графа  $v_i \in \mathbf{V}$  до ближайшего от источника приемника

сигнала, путь до которого следует найти. При  $\alpha = 0$  (4) сводится к формуле стандартной трассировки [29].

Подстроечный коэффициент  $\alpha$  регулирует степень влияния компоненты направленного поиска на общую эвристику построения трассировочного дерева. В табл. 3 представлены результаты применения различных способов расчета и фиксированных значений этого

коэффициента на примере трассировки программируемой цифровой части ПАЦИС [30] в форме ПЛИС с архитектурой островного типа. Доля разведенных пользовательских схем рассчитывалась как  $\sum_{i=1}^C R(ckt_i) / C$ , где  $C$  – количество схем в используемом тестовом наборе.

Таблица 3

Влияние подстроечного коэффициента направленного поиска

| № | Формула расчета коэффициента $\alpha$                        | Уменьшение длительности трассировки, раз | Доля разведенных схем |
|---|--|--|-----------------------|
| 1 | $\alpha = 1.0$   | 2.05                                     | 1.0                   |
| 2 | $\alpha = e^{Cost/costMax}$                                  | 2.43                                     | 0.74                  |
| 3 | $\alpha = e^{1-Cost/costMax}$                                | 2.53                                     | 0.59                  |
| 4 | $\alpha = e^{(e/iterN)}$                                     | 3.22                                     | 0.37                  |
| 5 | $\alpha = e^{(e/\sqrt{iterN})}$                              | 3.65                                     | 0.3                   |
| 6 | $\alpha = \frac{e^{Cost/maxCost}}{\sqrt{iterN}}$             | 1.72                                     | 0.7                   |
| 7 | $\alpha = 1.0$ , но $Cost$ рассчитана по Евклидовой метрике. | 1.81                                     | 0.78                  |

Согласно табл. 3 и рис. 1, наилучший результат с точки зрения сохранения трассируемости возможен при фиксированном значении коэффициента  $\alpha = 1.0$ . В этом случае длительность поиска трассировочного решения уменьшается в 2.05 раза в сравнении со стандартной трассировкой (формула (4) при  $\alpha = 0$ ). Зависимость степени влияния направленного поиска от номера текущей итерации (табл. 3, №№ 4-6), с одной стороны, приводит к ускорению прохождения процедуры трассировки, а с другой – становится причиной падения суммарной трассируемости пользовательских схем в составе тестового набора.

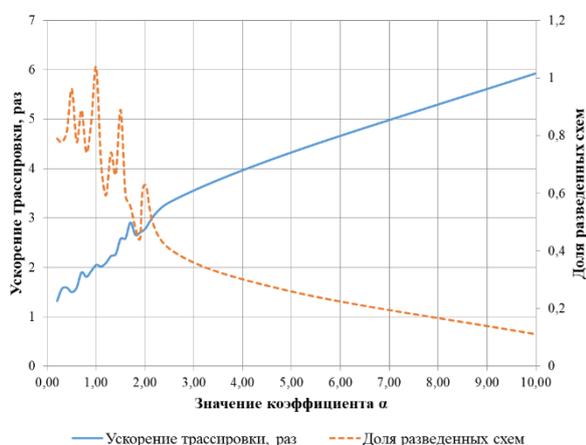


Рис. 1. Влияние фиксированного значения коэффициента  $\alpha$  на результаты трассировки

Для полноты выводов необходимо также провести сравнение временных характеристик пользовательских схем, получаемых с помощью двух вариантов поисковой трассировочной эвристики. Данный вопрос станет направлением дальнейших исследований по теме этой работы.

## V. ЗАКЛЮЧЕНИЕ

В работе представлено исследование двух методов, позволяющих ускорить прохождение этапа трассировки межсоединений в рамках маршрута проектирования на основе ПЛИС островного типа.

Первый метод, являющийся модификацией метода из работы [29], позволяет получить трассировочное решение при небольшом увеличении длины критического пути, но за более короткое время, чем при использовании оригинального метода. При этом выигрыш по времени трассировки тем значительнее, чем выше доля сильно разветвленных цепей в составе рассматриваемой пользовательской схемы.

Второй метод, развивающий идею направленного поиска, обеспечивает возможность двукратного уменьшения времени трассировки при сохранении полной суммарной трассируемости тестового набора пользовательских схем.

## БЛАГОДАРНОСТИ

Автор выражает благодарность к.т.н. Железникову Д.А. за ценные методические и технические замечания.

ЛИТЕРАТУРА

- [1] Kramer M. R., van Leeuwen J. Wire-Routing is NP-Complete / Technical. Report RUU-CS-82-4.
- [2] Al-Hyari A. и др. Novel Congestion-estimation and Routability-prediction Methods based on Machine Learning for Modern FPGAs // ACM Transactions on Reconfigurable Technology and Systems. 2019. Т. 12. № 3. PP. 1-25.
- [3] Alawieh M. B. и др. High-Definition Routing Congestion Prediction for Large-Scale FPGAs // 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020, pp. 26-31.
- [4] Farooq U., Ul Hasan N., Baig I., Zghaibeh M. Efficient FPGA Routing using Reinforcement Learning // 2021 12th International Conference on Information and Communication Systems (ICICS), 2021, pp. 106-111.
- [5] McMurchie L., Ebeling C. PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs // Third International ACM Symposium on Field-Programmable Gate Arrays, 1995, pp. 111-117.
- [6] Hu J., Roy J.A., Markov I.L. Sidewinder: a scalable ILP-based router // Proc. of the 2008 international workshop on System level interconnect prediction (SLIP '08). Association for Computing Machinery, New York, NY, USA, 73–80.
- [7] Nam G.-J., Sakallah K.A., Rutenbar R.A. A new FPGA detailed routing approach via search-based Boolean satisfiability // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, vol. 21, no. 6, pp. 674–684.
- [8] Fraisse H., Joshi A., Gaitonde D., Kaviani A. Boolean satisfiability-based routing and its application to Xilinx ultrascale clock network // Proc. of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '16), 2016, pp. 74–79.
- [9] Fraisse H. Incremental routing for circuit designs using a SAT router. Patent USA no. 10445456, 2019.
- [10] Park D., Kang I., Kim Y., Gao S., Lin B., Cheng C.-K. ROAD: Routability Analysis and Diagnosis Framework Based on SAT Techniques // Proc. of the 2019 International Symposium on Physical Design (ISPD '19). PP. 65–72.
- [11] Dijkstra E. W. A Note on Two Problems in Connection with Graphs // Numer. Math., 1, 269-271, 1959.
- [12] Moore E. F. The shortest path through a maze // Proc. of an International Symposium on the Theory of Switching, 1959. Vol. 2. PP. 285–292.
- [13] Lee C.Y. An Algorithm for Path Connections and Its Applications // IRE Transactions on Electronic Computers, vol. EC-10, number 2, pp. 364–365, 1961
- [14] Hart P. E., Nilsson N. J., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths // IEEE Transactions on Systems Science and Cybernetics, 1968. Vol. 4, no.2. PP. 100–107.
- [15] Murray K. E., Zhong Sh., Betz V. AIR: A Fast but Lazy Timing-Driven FPGA Router // 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, 2020, pp. 338-344
- [16] Murray K.E., et al. VTR 8: High-performance CAD and Customizable FPGA Architecture Modelling // ACM Trans. Reconfigurable Technol. Syst., 2020. Vol.13, no.2. Article 9. 55 p.
- [17] Vercruyce D., Vansteenkiste E., Stroobandt D. CRoute: a fast high-quality timing-driven connection-based FPGA router // 2019 27th IEEE Annual International Symposium on Field-programmable Custom Computing Machines (FCCM), San Diego, CA, 2019, pp. 53–60.
- [18] Gort M., Anderson J. H. Accelerating FPGA Routing Through Parallelization and Engineering Enhancements // IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 2012. Vol. 31, no. 1. PP. 61–74.
- [19] Shen M., Luo G., Xiao N. Combining Static and Dynamic Load Balance in Parallel Routing for FPGAs // in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 14 p.
- [20] Zhou Y., Vercruyce D., Stroobandt D. Accelerating FPGA Routing Through Algorithmic Enhancements and Connection-aware Parallelization // ACM Trans. Reconfigurable Technol. Syst., 2020. Vol. 13, no.4. Article 18, 26 p.
- [21] He J., Burtscher M., Manohar R., Pingali K. SPRoute: A Scalable Parallel Negotiation-based Global Router // 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2019. PP. 1-8.
- [22] Заплетина М.А., Железников Д.А., Гаврилов С.В. Иерархический подход к трассировке реконфигурируемой системы на кристалле островного типа // Проблемы разработки перспективных микро- и нанoeлектронных систем (МЭС). 2020. Выпуск 3. С. 16-21. doi:10.31114/2078-7707-2020-3-16-21
- [23] Intel Quartus Suite: <https://www.intel.ru/content/www/ru/ru/software/programmable/quartus-prime/overview.html> (дата обращения: 05.08.2021).
- [24] Vivado Design Suite HLx Editions: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2020-3.html> (дата обращения: 05.08.2021).
- [25] Korolija D., Stojilović M. FPGA-Assisted Deterministic Routing for FPGAs // 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2019. PP. 155-162
- [26] Shen M., Luo G. Corolla: GPU-Accelerated FPGA Routing Based on Subgraph Dynamic Expansion // Proc. of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '17), 2017. PP. 105–114.
- [27] Shen M., Luo G., Xiao N. Exploring GPU-Accelerated Routing for FPGAs // IEEE Transactions on Parallel and Distributed Systems, 2019. Vol. 30, no. 6. PP. 1331-1345.
- [28] Железников Д. А., Заплетина М. А., Хватов В. М. Решение задачи трассировки межсоединений для реконфигурируемых систем на кристалле с различными типами коммутационных элементов // Электронная техника. Серия 3: Микроэлектроника, 2018. № 4(172). С. 31-36.
- [29] Zapletina M. A., Zheleznikov D. A., Gavrilov S. V. Improving Pathfinder Algorithm Performance for FPGA Routing // Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2021, Moscow, 2021. PP. 2054-2057.
- [30] ДЦ «Союз» - 5400TP094 ПАЦИС: <https://dcsoyuz.ru/products/pais/art/1605> (дата обращения 29.08.2021).

# Methods for Speeding up the Modified Pathfinder Routing Algorithm for Island-Style FPGA

M. A. Zapletina

Institute for design problems in microelectronics of RAS, Moscow, zapletina\_m@ippm.ru

**Abstract** — The paper presents two methods of speeding up the interconnection routing stage within the layout design flow for field-programmable gate arrays with island-style architecture. The basic algorithm used in this work is Pathfinder algorithm modified to support representation of routing resources of an FPGA chip in the form of a mixed route graph. The first method is based on reducing the number of rip-up & reroute iterations. It was found that a reroute of congested nets only allows reducing the routing time by 38% on average while increasing the critical path delay up to 10% only. The second method expands the idea of directed pathfinding on a route graph with spatial characteristics known. Various directed search adjustment factors were investigated and the optimal one was discovered. It was found that decrease in the adjustment factor depending on the iteration of the routing algorithm leads to acceleration of a routing task solution but causes significant drop in routes quality at the same time. The introduced method adjusted by an optimal factor is capable of doubling routing speedup without routability degradation.

**Keywords** — routing, interconnect, FPGA, rip-up and reroute, place & route, directed search, Pathfinder.

## REFERENCES

- [1] Kramer M. R., van Leeuwen J. Wire-Routing is NP-Complete / Technical. Report RUU-CS-82-4.
- [2] Al-Hyari A. и др. Novel Congestion-estimation and Routability-prediction Methods based on Machine Learning for Modern FPGAs // ACM Transactions on Reconfigurable Technology and Systems. 2019. T. 12. № 3. PP. 1-25.
- [3] Alawieh M. B. et al. High-Definition Routing Congestion Prediction for Large-Scale FPGAs // 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020, pp. 26-31.
- [4] Farooq U., Ul Hasan N., Baig I., Zghaibeh M. Efficient FPGA Routing using Reinforcement Learning // 2021 12th International Conference on Information and Communication Systems (ICICS), 2021, pp. 106-111.
- [5] McMurchie L., Ebeling C. PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs // Third International ACM Symposium on Field-Programmable Gate Arrays, 1995, pp. 111-117.
- [6] Hu J., Roy J.A., Markov I.L. Sidewinder: a scalable ILP-based router // Proc. of the 2008 international workshop on System level interconnect prediction (SLIP '08). Association for Computing Machinery, New York, NY, USA, 73–80.
- [7] Nam G.-J., Sakallah K.A., Rutenbar R.A. A new FPGA detailed routing approach via search-based Boolean satisfiability // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2002, vol. 21, no. 6, pp. 674–684.
- [8] Fraisse H., Joshi A., Gaitonde D., Kaviani A. Boolean satisfiability-based routing and its application to Xilinx ultrascale clock network // Proc. of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '16), 2016, pp. 74–79.
- [9] Fraisse H. Incremental routing for circuit designs using a SAT router. Patent USA no. 10445456, 2019.
- [10] Park D., Kang I., Kim Y., Gao S., Lin B., Cheng C.-K. ROAD: Routability Analysis and Diagnosis Framework Based on SAT Techniques // Proc. of the 2019 International Symposium on Physical Design (ISPD '19). PP. 65–72.
- [11] Dijkstra E. W. A Note on Two Problems in Connection with Graphs // Numer. Math., 1, 269-271, 1959.
- [12] Moore E. F. The shortest path through a maze // Proc. of an International Symposium on the Theory of Switching, 1959. Vol. 2. PP. 285—292.
- [13] Lee C.Y. An Algorithm for Path Connections and Its Applications // IRE Transactions on Electronic Computers, vol. EC-10, number 2, pp. 364—365, 1961
- [14] Hart P. E., Nilsson N. J., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths // IEEE Transactions on Systems Science and Cybernetics, 1968. Vol. 4, no.2. PP. 100–107.
- [15] Murray K. E., Zhong Sh., Betz V. AIR: A Fast but Lazy Timing-Driven FPGA Router // 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, 2020, pp. 338-344
- [16] Murray K.E., et al. VTR 8: High-performance CAD and Customizable FPGA Architecture Modelling // ACM Trans. Reconfigurable Technol. Syst, 2020. Vol.13, no.2. Article 9. 55 p.
- [17] Vercruyce D., Vansteenkiste E., Stroobandt D. CRoute: a fast high-quality timing-driven connection-based FPGA router // 2019 27th IEEE Annual International Symposium on Field-programmable Custom Computing Machines (FCCM), San Diego, CA, 2019, pp. 53–60.
- [18] Gort M., Anderson J. H. Accelerating FPGA Routing Through Parallelization and Engineering Enhancements // IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 2012. Vol. 31, no. 1. PP. 61–74.
- [19] Shen M., Luo G., Xiao N. Combining Static and Dynamic Load Balance in Parallel Routing for FPGAs // in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 14 p.
- [20] Zhou Y., Vercruyce D., Stroobandt D. Accelerating FPGA Routing Through Algorithmic Enhancements and Connection-aware Parallelization // ACM Trans. Reconfigurable Technol. Syst., 2020. Vol. 13, no.4. Article 18, 26 p.
- [21] He J., Burtscher M., Manohar R., Pingali K. SPRoute: A Scalable Parallel Negotiation-based Global Router // 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2019. PP. 1-8.
- [22] Zapletina M.A., Zheleznikov D.A., Gavrilov S.V. The Hierarchical Approach to Island Style Reconfigurable System-on-a-chip Routing // Problems of Perspective Micro-

- and Nanoelectronic Systems Development - 2020. Issue 3. P. 16-21. doi:10.31114/2078-7707-2020-3-16-21
- [23] Intel Quartus Suite: <https://www.intel.ru/content/www/ru/ru/software/programmable/quartus-prime/overview.html> (accessed on 05.08.2021).
- [24] Vivado Design Suite HLx Editions: <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2020-3.html> (accessed on 05.08.2021).
- [25] Korolija D., Stojilović M. FPGA-Assisted Deterministic Routing for FPGAs // 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2019. PP. 155-162
- [26] Shen M., Luo G. Corolla: GPU-Accelerated FPGA Routing Based on Subgraph Dynamic Expansion // Proc. of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '17), 2017. PP. 105–114.
- [27] Shen M., Luo G., Xiao N. Exploring GPU-Accelerated Routing for FPGAs // IEEE Transactions on Parallel and Distributed Systems, 2019. Vol. 30, no. 6. PP. 1331-1345.
- [28] Zheleznikov D. A., Zapletina M. A., Hvatov V. M. Reshenie zadachi trassirovki mezhsoedinenij dlja rekonfiguriruemih sistem na kristalle s razlichnymi tipami kommutacionnyh jelementov (Solving the problem of interconnects routing for reconfigurable systems on a chip with various types of switching elements) // Elektronnaja tehnika. Serija 3: Mikroelektronika, 2018. № 4(172). S. 31-36.
- [29] Zapletina M. A., Zheleznikov D. A., Gavrilov S. V. Improving Pathfinder Algorithm Performance for FPGA Routing // Proceedings of the 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, ElConRus 2021, Moscow, 2021. PP. 2054-2057.
- [30] Design Cenetr “Soyuz” - Programmable analog-digital integrated circuit 5400TP094: <https://dcsoyuz.ru/products/pais/art/1605> (accessed on 29.08.2021).