

Алгоритм декомпозиции на основе метода имитации отжига для реконфигурируемых систем на кристалле

С.В. Гаврилов, Д.А. Железников, Р.Ж. Чочаев, В.М. Хватов

Институт проблем проектирования в микроэлектронике РАН, г. Москва

zheleznikov_d@ippm.ru, chochaev_r@ippm.ru

Аннотация - В данной работе представлен алгоритм на основе метода имитации отжига для оптимизации результатов декомпозиции в маршруте проектирования схем в базе РСнК «Алмаз-14» с целевой оценочной функцией, основанной на правиле Рента. Приведено численное сравнение предложенного алгоритма со следующими тремя популярными методами: базовым алгоритмом кластеризации, алгоритмом Кернигана-Лина и алгоритмом кластеризации iRAC. Результаты численных экспериментов на наборах тестовых схем ISCAS-85 и ISCAS-89 показывают, что представленный алгоритм оптимизации в комбинации с алгоритмом iRAC имеет лучшую эффективность по сравнению с другими алгоритмами декомпозиции.

Ключевые слова - автоматизация проектирования; кластеризация; правило Рента; алгоритм Кернигана-Лина; имитация отжига.

I. ВВЕДЕНИЕ

За последнее десятилетие реконфигурируемые системы на кристалле (РСнК) прочно заняли нишу в промышленном и телекоммуникационном секторах. Благодаря возможности перепрограммирования и более быстрому процессу проектирования такие системы набирают в мире всё большую популярность.

Для успешного размещения и трассировки на РСнК soft-процессоров или иных больших схем в маршруте проектирования необходимо учитывать этап декомпозиции ЛЭ в группы [1]. Он является одним из ключевых в маршруте проектирования схем в базе реконфигурируемых систем на кристалле. На нём решаются задачи повышения плотности компоновки за счет уменьшения количества создаваемых подсхем, и уменьшения временных задержек за счет локализации критических путей внутри подсхем и использования быстрых локальных коммутационных ресурсов. Применение эффективных алгоритмов декомпозиции позволяет повысить трассируемость схем и понизить потребляемую мощность за счет уменьшения используемых трассировочных ресурсов. Особенно важным этот этап становится в случае ограниченного количества глобальных трасс.

В связи с продолжающимся ростом популярности РСнК перед российской микроэлектроникой встала задача разработки и выпуска отечественных РСнК для

промышленной и космической индустрии. АО «НИИМЭ» и ПАО «Микрон» в 2017 году представили реконфигурируемую систему на кристалле на базе ПЛИС «Алмаз-14». Данная РСнК помимо программируемой логики содержит в себе различные «макро» блоки, такие как блоки памяти, умножители, блоки фазовой автоподстройки частоты (ФАПЧ). Для проектирования схем в базе данной РСнК ИППМ РАН совместно с АО «НИИМЭ» разработали специализированные средства автоматизации проектирования (САПР) для решения задач декомпозиции, размещения элементов и трассировки межсоединений [2], т.к. использование зарубежных средств в данном случае оказывается невозможным в связи с их ориентированностью на собственные технологии.

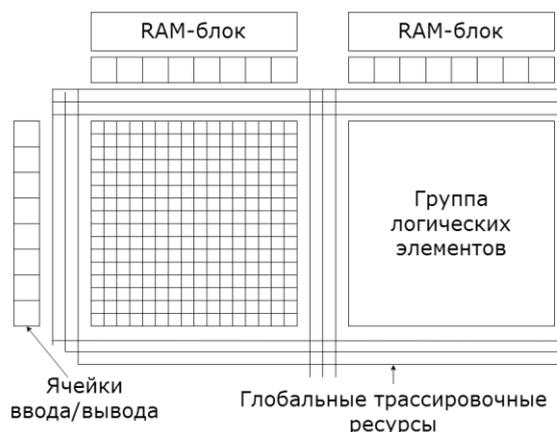


Рис. 1. Глобальные трассировочные ресурсы

РСнК на базе ПЛИС «Алмаз-14» имеет типичную островную архитектуру. В РСнК содержится более 20 тысяч логических элементов, объединенных в группы по 256 элементов (рис. 1). Каждый из логических элементов (ЛЭ) состоит из двух логических ячеек (ЛЯ). Логическая ячейка может реализовывать как логическую функцию до трёх переменных, так и D-триггер. Логический элемент связан со своими соседями по вертикали и горизонтали с помощью быстрых локальных трасс (рис. 2).

Также в составе ПЛИС имеются глобальные трассировочные ресурсы (рис. 1), проходящие вертикально и горизонтально по всему кристаллу. Они

предназначены для соединения логических элементов в разных группах между собой, а также подключения к ячейкам ввода-вывода.

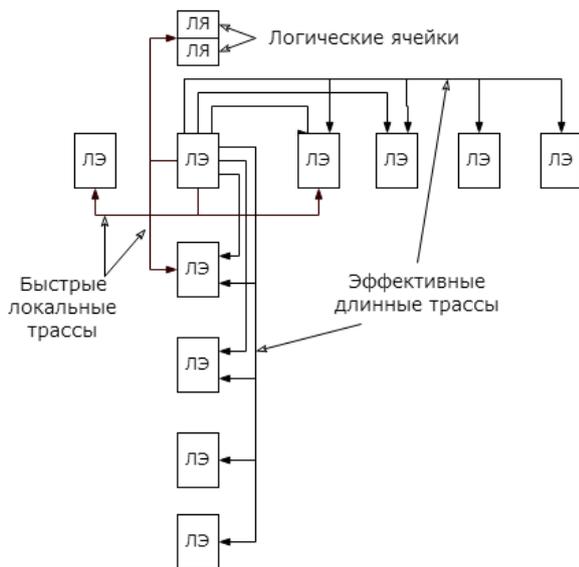


Рис. 2. Локальные трассировочные ресурсы

В данной работе задачу декомпозиции можно сформулировать следующим образом. Исходными данными являются неориентированный граф $G=(V, E)$, где V - множество вершин, соответствующее набору логических элементов проектируемой схемы; $E=V \times V$ - множество рёбер, соответствующее набору соединений между элементами. Необходимо разбить граф G на непересекающиеся подграфы G_1, G_2, \dots, G_n так, чтобы их количество было меньше, чем количество доступных групп логических элементов (ГЛЭ) на кристалле, а в каждой группе было не более N логических элементов. Кроме того, необходимо уменьшить количество межсоединений между ГЛЭ для уменьшения количества используемых глобальных шин. Существующие на данный момент алгоритмы решения задачи декомпозиции можно разделить на следующие группы: (1) нисходящие алгоритмы, (2) восходящие или алгоритмы кластеризации, (3) эвристические алгоритмы.

В данной работе представлен алгоритм оптимизации декомпозиции на основе метода имитации отжига. Также рассматриваются и анализируются следующие алгоритмы декомпозиции: (1) базовый алгоритм кластеризации, (2) алгоритм на основе метода Кернигана-Лина [3], (3) алгоритм iRAC [4]. Результатом исследования является оценка влияния данных алгоритмов на результаты последующих этапов размещения элементов и трассировки межсоединений.

II. ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

A. Базовый алгоритм кластеризации

Базовый алгоритм кластеризации относится к семейству «жадных» алгоритмов. Принцип его работы прост. Каждый раз при формировании нового кластера ищется элемент с наибольшим количеством связей и добавляется в кластер. Затем по принципу наибольшей связности с формируемым кластером происходит добавление свободных логических элементов. Единственное ограничение накладывается на размер кластера, который не должен превышать заданное число N . Алгоритм завершает работу когда не остаётся свободных логических элементов. Также необходимо отметить, что алгоритм не допускает добавление в формируемый кластер логических элементов, принадлежащих уже сформированным кластерам.

Данный алгоритм легко реализуется в САПР и достаточно быстр по сравнению с другими алгоритмами. Главным недостатком базового алгоритма является невозможность учёта различных архитектурных особенностей РСнК.

B. Алгоритм Кернигана-Лина

Алгоритм Кернигана-Лина [3] принадлежит к итерационным дихотомическим алгоритмам и работает по принципу «сверху-вниз». Данный метод делит граф на 2 непересекающихся множества, стараясь уменьшить стоимость разреза с помощью перестановок логических элементов. Алгоритм принимает на вход неориентированный граф $G=(V, E)$, где V - множество вершин, $E=V \times V$ - множество рёбер.

Пусть $E(v)$ - число рёбер, инцидентных вершине v и пересекающих разрез, а $I(v)$ - число рёбер с вершинами внутри группы, которой принадлежит $v \in V$. Тогда ценой перемещения вершины v является функция $D(v) = E(v) - I(v)$. Если $D(v) > 0$, то цена разреза уменьшается и перемещать вершину выгодно, в противном случае - нет.

Стоимость перестановки вершин a и b местами определяется следующим образом:

$$\Delta g(a, b) = D(a) - D(b) - 2c(a, b), \quad (1)$$

где $c(a, b)$ - связность вершин a и b . Чем больше значение Δg , тем меньше стоимость разреза. Таким образом, для того чтобы разделить граф на две группы алгоритм на первой итерации вычисляет для всех вершин графа значение $D(v)$. Затем для всех возможных перестановок пар вершин вычисляется стоимость перестановки Δg и ищется её максимальное значение. Вершины a и b с максимальным Δg переставляются и фиксируются, значения $D(v)$ для связанных с a и b незафиксированных вершин v пересчитываются. Итерации выполняются до тех пор, пока все вершины не станут зафиксированными.

Затем из всей последовательности операций перестановок выбирается такая последовательность длиной m , что $\sum_{i=1}^m \Delta g_i \rightarrow \min$.

Алгоритм Кернигана-Лина был адаптирован для оптимизации решения, полученного после выполнения базового алгоритма кластеризации. Адаптированный алгоритм принимает на вход список из n кластеров. Кластеры сортируются по порядку, а затем последовательно в два прохода для каждой пары кластеров выполняется оптимизация алгоритмом Кернигана-Лина. На первом проходе рассматриваются пары соседних групп $(a_i, a_i + 1)$, где $i = 0, \dots, n$. На втором проходе рассматриваются пары (a_{i-1}, a_i) .

Применение адаптированного алгоритма позволяет уменьшить количество внешних межсоединений групп логических элементов и тем самым понизить количество глобальных трассировочных ресурсов, которые будут использованы при трассировке.

С. Алгоритм кластеризации iRAC

Данный алгоритм впервые был представлен в работе [4]. В его основе лежит правило Рента [5, 6], которое позволяет эвристически оценивать сложность логических схем. Данное правило выражается эмпирической зависимостью в виде степенной функции вида: $N_{io} = KB^p$, где N_{io} - количество внешних терминалов схемы, B - количество элементов в схеме, K - количество соединений, в среднем приходящихся на один элемент в схеме, и p - экспонента Рента.

Экспонента Рента является эвристической оценкой сложности межсоединений архитектуры. Значение экспоненты варьируется от 0 до 1 и позволяет предсказывать результаты размещения и трассировки. Данный алгоритм позволяет минимизировать как количество используемых межсоединений, так и площадь, которую занимает устройство.

Алгоритм на вход принимает граф соединений $G = (V, E, w)$, где V - множество логических элементов проектируемой схемы, $E = V \times V$ - множество соединений между элементами, а $w = 2/r > 0$ - вес, назначаемый каждому соединению $e \in E$, где r - количество терминалов цепи e .

Работа алгоритма состоит из двух этапов. На первом этапе для каждого логического элемента вычисляется показатель связности c_v по формуле $c_v = \text{separation} / \text{degree}^2$, где separation - количество инцидентных ЛЭ цепей, а degree - суммарное количество терминалов этих цепей. Затем список сортируется по убыванию значения параметра degree , а ЛЭ с одинаковыми значениями параметра сортируются по возрастанию показателя связности c_v .

На втором этапе происходит формирование кластеров. В качестве первого элемента каждого нового кластера выбирается ЛЭ с максимальным значением параметра degree и минимальным значением показателя связности c_v . Это позволяет

уменьшить количество внешних цепей формируемого кластера.

После выбора начального элемента для каждого логического элемента, связанного с формируемым кластером, вычисляется коэффициент притяжения. ЛЭ с наибольшим значением коэффициента добавляется в кластер. Допустим, что элемент $v \in V$ - один из кандидатов в кластер C , имеющий с ним одну общую цепь $e \in E$. Тогда коэффициент притяжения есть функция от веса цепи и количества терминалов этой цепи, находящихся внутри кластера $\text{Gain}(v, C, e) = 2 \cdot n \cdot w(e) \cdot (1 + p_e)$, где p_e - количество терминалов цепи $e \in E$, находящихся внутри кластера, $w(e)$ - вес цепи, n - текущий размер формируемого кластера.

Если $v \in V$ и C имеют больше одной общей цепи, то коэффициент притяжения равен сумме коэффициентов каждой из общих цепей.

В том случае, если после добавления ЛЭ в кластер все терминалы цепи $e \in E$ окажутся внутри кластера, то коэффициент притяжения $\text{Gain}(v, C, e)$ умножается на константу $\alpha = 100$. Это позволяет гарантировать, что в кластер будут добавлены элементы с небольшими цепями, которые имеют большую вероятность впоследствии стать внутренними.

Каждый новый ЛЭ может быть добавлен в кластер только в том случае, если при этом удовлетворяются ограничения на размер и количество внешних терминалов кластера. Для этого количество внешних терминалов ограничивается с помощью правила Рента. Главная задача заключается в том, чтобы значение экспоненты Рента формируемого кластера было меньше глобального значения экспоненты для архитектуры кристалла. Максимально возможное количество внешних терминалов кластера равно:

$$j = (k+1)N^{P_a}, k+1 \leq j < 9N, \quad (2)$$

где k - максимальное количество входов ЛЭ, N - заданный размер кластера, $0 < P_a \leq 1$ - экспонента Рента для данной архитектуры. В данной работе параметр P_a для архитектуры РСнК «Алмаз-14» по умолчанию был выбран равным 0.85. Также был определен параметр N - максимальное количество ЛЭ внутри кластера, равный 256. Изменяя данные параметры можно контролировать плотность заполнения кластеров и, соответственно, количество внешних связей каждой ГЛЭ.

Д. Алгоритм оптимизации на основе метода имитации отжига

Метод имитации отжига является эвристическим алгоритмом, предназначенным для решения задач оптимизации. В алгоритме используется упорядоченный случайный поиск, основанный на моделировании процесса кристаллизации вещества, например, отжига металлов. Алгоритм был впервые описан в 1953 году в работе [12]. В 1983 году С. Киркпатрик впервые предложил использовать метод

График охлаждения отжига

Доля принятых перестановок $R_{accepted}$	α
(0.96; 1]	0.5
(0.8; 0.96]	0.9
(0.15; 0.8]	0.95
[0; 0.15]	0.8

имитации отжига для решения задач оптимизации [13]. Основной особенностью данного метода является то, что в процессе поиска глобального экстремума он с некоторой вероятностью принимает изменения, которые могут ухудшить значение целевой оценочной функции. Это позволяет избежать «застывания» в локальных экстремумах целевой функции и продолжить поиск глобального.

Основными параметрами алгоритма являются:

- 1) начальное решение,
- 2) целевая функция,
- 3) график охлаждения,
- 4) начальное и конечное значения температуры.

Основной идеей разработанного алгоритма оптимизации декомпозиции является использование метода имитации отжига с целевой функцией на основе правила Рента для улучшения некоторого начального решения.

Целевая оценочная функция рассчитывается по формуле [4]:

$$Cost = (\sum_{i=1}^{N_c} p_i N_i) / N_g, \quad (3)$$

где p_i - экспонента Рента i -го кластера, N_i - количество ЛЭ в кластере, N_g - общее количество ЛЭ в схеме и N_c - общее количество кластеров. Экспонента Рента для каждого отдельного кластера вычисляется из классической формулы $N_{io} = KB^p$.

Начальная и конечная температуры для алгоритма получены экспериментально. Значение начальной температуры:

$$T_{initial} = 10000 \cdot Cost_{initial}, \quad (4)$$

где $Cost_{initial}$ - стартовое значение целевой оценочной функции после начальной декомпозиции. Конечная температура: $T_{final} = 0.00001$. Количество итераций L при температуре T : $L = 100 \cdot N_g$.

График охлаждения рассчитывается по формуле: $T_{new} = \alpha \cdot T_{old}$, где T_{old} - предыдущее значение температуры, α является параметром, который зависит от доли принятых решений (R_{accept}) при T_{old} , как показано в табл. 1.

На каждой итерации алгоритма выполняется перестановка двух случайных ЛЭ из разных кластеров, после чего производится оценка значения целевой функции. Уменьшение значения целевой функции ведет к увеличению структурной однородности кластеров, что, в свою очередь, приводит к уменьшению количества требуемых трассировочных ресурсов на последующем этапе трассировки межсоединений.

III. РЕЗУЛЬТАТЫ

Представленные алгоритмы тестировались на наборах схем ISCAS-85 [7], ISCAS-89 [8] и схеме UART [9]. После декомпозиции тестовых схем выполнялись этапы размещения и трассировки межсоединений. Размещение элементов на РСнК осуществлялось на двух уровнях иерархии с помощью алгоритма на основе метода имитации отжига [10] с модифицированной целевой функцией. На первом проходе выполнялось размещение на уровне ГЛЭ, на втором - внутри ГЛЭ. Трассировка выполнялась с помощью модифицированного алгоритма A^* [11] со встроенным механизмом разрыва и перетрассировки.

В табл. 2 приведены результаты проведенных экспериментов. Значения в таблице были получены для случая, когда начальная декомпозиция выполнялась с помощью описанных алгоритмов, после чего выполнялась оптимизация с помощью предложенного метода на основе имитации отжига. Как видно из таблицы применение метода имитации отжига позволило уменьшить количество внешних межсоединений кластеров в среднем на 42%. Уменьшение количества внешних межсоединений позволяет также уменьшить и количество используемых глобальных шин, что в свою очередь приводит к повышению трассируемости схем на РСнК «Алмаз-14» на первой итерации алгоритма трассировки. Из таблицы также видно, что применение алгоритма оптимизации в комбинации с алгоритмом iRAC имеет лучшую эффективность по сравнению с другими алгоритмами декомпозиции.

IV. ВЫВОДЫ

В данной работе представлен новый подход на основе метода имитации отжига для оптимизации декомпозиции сложных схем при проектировании в базисе РСнК. Был проведен анализ и сравнение с несколькими существующими алгоритмами декомпозиции: (1) базовым алгоритмом кластеризации, (2) алгоритмом Кернигана-Лина, (3) алгоритмом iRAC. Все алгоритмы были реализованы в САПР и протестированы на наборах тестовых схем ISCAS-85, ISCAS-89 и схеме UART. Было показано, что представленный метод с целевой оценочной функцией на основе правила Рента для РСнК «Алмаз-14» позволяет повысить трассируемость проектируемых схем по сравнению с рассмотренными алгоритмами.

Результаты экспериментов

Алгоритм	Схема	Кол-во ЛЭ / ГЛЭ	Среднее кол-во внешних цепей до отжига	Среднее кол-во внешних цепей после отжига	Улучшение (%)	Кол-во неразведенных цепей до отжига	Кол-во неразведенных цепей после отжига	Улучшение (%)
Базовый	c1908	220/3	75	51	32	17	8	53
	s1196	313/5	104	47	55	85	37	56
	s1238	308/5	106	46	57	82	33	60
	UART	987/14	132	51	61	425	173	59
Керниган-Лин	c1908	220/3	73	52	29	16	5	69
	s1196	313/5	101	47	53	78	35	55
	s1238	308/5	105	44	58	67	37	45
	UART	987/14	131	48	63	424	172	59
iRAC	c1908	220/3	70	52	26	13	12	8
	s1196	313/5	65	45	31	46	29	37
	s1238	308/5	66	44	33	44	33	25
	UART	987/14	56	47	16	173	152	12

ЛИТЕРАТУРА

- [1] Hauck S., DeHon A. Reconfigurable computing: the theory and practice of FPGA-based computation. CA: Morgan Kaufmann Publishers Inc., San Francisco, 2007. - P. 944.
- [2] Garbulina T., Khvatov V. Methods and algorithms of design on domestic FPGAs with given restrictions on routing resources // Moscow Workshop on Electronic and Networking Technologies, MWENT-2018.
- [3] Kernighan and Lin. An efficient heuristic procedure for partitioning graphs // The Bell System Technical Journal, vol. 49, no. 2, 1970, P. 291 - 307.
- [4] A. Singh and M. Marek-Sadowska. Efficient circuit clustering for Area and Power Reduction in FPGAs // ACM Transactions on Design Automation of Electronic Systems, Vol. 7, No. 4, October 2002, P. 643-663.
- [5] B.S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs // IEEE Trans. on Comput., 1971, vol. C-20: P. 469-479.
- [6] H. Van Marck, D. Stroobandt, and J. Van Campenhout. Towards an extension of Rent's rule for describing local variations in interconnect complexity // Proceedings of the Fourth International Symposium for Young Computer Scientists, 1994, P. 136-141.
- [7] Bryan D. The ISCAS '85 Benchmark Circuits and Netlist Format. North-Carolina State University, 1985.
- [8] Brglez F., Bryan D., and Kozminski K. Notes on the ISCAS '89 Benchmark Circuits. North-Carolina State University, 1989.
- [9] Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier. Introduction to Embedded Systems: Using Microcontrollers and the MSP430. Springer Science & Business Media, 2013, P. 648.
- [10] V. Betz and J. Rose. VPR: A new packing, placement and routing tool for FPGA research // Proceedings of the Seventh International Workshop on Field-Programmable Logic and Applications. Springer-Verlag, 1997, P. 213-222.
- [11] Gavrilov S.V., Zheleznikov D.A., Khvatov V.M. Solution of Interconnect Trace Tasks with Resynthesis for Reconfigurable Systems-on-Chip // Proc. of universities. Electronics. - 2017. - Vol.22. - №3. - P. 266-275.
- [12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computer Machines // J. Chemical Physics. 21. 6. June. 1953, P. 1087-1092.
- [13] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi, Optimization by Simulated Annealing // Science. 220. 1983, P.671-680.

Partitioning Algorithm Based on Simulated Annealing for Reconfigurable Systems-on-Chip

S.V. Gavrilov, D.A. Zheleznikov, R.Z. Chochaev, V.M. Khvatov

Institute for Design Problems in Microelectronics of RAS, Moscow

zheleznikov_d@ippm.ru, chochaev_r@ippm.ru

Abstract - Partitioning is one of the most significant steps in the reconfigurable systems-on-chip design flow. High-quality partitioning ensures effective placement and routing steps. The goals of circuit partitioning are following: a) achieving the high density by minimizing the number of groups; b) decreasing time delays by localizing time-critical connections

within a group and using fast local routing resources. There are several popular partitioning problem solutions, such as top-down decomposition algorithms, bottom-up clustering and different heuristic algorithms. In this paper we present a simulated annealing approach for partitioning optimization for the reconfigurable system-on-chip (RSoC) based on the

“Almaz-14” FPGA. This RSoC contains more than twenty thousand logic elements arranged into groups of size 256. It has typical island architecture with rich local and poor global routing resources therefore partitioning step becomes vital for successful placement and routing. In our work we use simulated annealing for optimization the initial solution generated by other clustering algorithms. The cost function is based on Rent's rule and it is a function of: (1) Rent's exponent of each cluster, (2) number of LE in each cluster, (2) total number of LE in the circuit and (3) total number of clusters. Such parameters as annealing schedule, initial and final temperature values were obtained experimentally. We analyze and compare our algorithm with three popular approaches: basic clustering; Kernighan-Lin partitioning algorithm; clustering algorithm iRAC. Each method was verified on the set of benchmark circuits ISCAS-85 and ISCAS-89. The algorithms were configured in such way to divide the benchmark circuits into an equal number of clusters. Benchmark circuits were placed using the algorithm based on the simulated annealing method with modified cost function and routed by using the modified algorithm A* with stop at the first iterations of rip-up and reroute procedure. Experimental results demonstrate that presented algorithm with initial placement generated by iRAC has better effectiveness compared to other partitioning algorithms.

Keywords - design automation; clustering; Rent's rule; Kernighan-Lin algorithm; Simulated Annealing.

REFERENCES

- [1] Garbulina T., Khvatov V. Methods and algorithms of design on domestic FPGAs with given restrictions on routing resources // Moscow Workshop on Electronic and Networking Technologies, MWENT-2018.
- [2] Hauck S., DeHon A. Reconfigurable computing: the theory and practice of FPGA-based computation. CA: Morgan Kaufmann Publishers Inc., San Francisco, 2007. - P. 944.
- [3] Kernighan and Lin. An efficient heuristic procedure for partitioning graphs // The Bell System Technical Journal, vol. 49, no. 2, 1970, P. 291 - 307.
- [4] A. Singh and M. Marek-Sadowska. Efficient circuit clustering for Area and Power Reduction in FPGAs // ACM Transactions on Design Automation of Electronic Systems, Vol. 7, No. 4, October 2002, P. 643-663.
- [5] B.S. Landman and R. L. Russo. On a pin versus block relationship for partitions of logic graphs // IEEE Trans. on Comput., 1971, vol. C-20: P. 469-1479.
- [6] H. Van Marck, D. Stroobandt, and J. Van Campenhout. Towards an extension of Rent's rule for describing local variations in interconnect complexity // Proceedings of the Fourth International Symposium for Young Computer Scientists, 1994, P. 136-141.
- [7] Bryan D. The ISCAS '85 Benchmark Circuits and Netlist Format. North-Carolina State University, 1985.
- [8] Brglez F., Bryan D., and Kozminski K. Notes on the ISCAS '89 Benchmark Circuits. North-Carolina State University, 1989.
- [9] Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier. Introduction to Embedded Systems: Using Microcontrollers and the MSP430. Springer Science & Business Media, 2013, P. 648.
- [10] V. Betz and J. Rose. VPR: A new packing, placement and routing tool for FPGA research // Proceedings of the Seventh International Workshop on Field-Programmable Logic and Applications. Springer-Verlag, 1997, P. 213-222.
- [11] Gavrilov S.V., Zheleznikov D.A., Khvatov V.M. Solution of Interconnect Trace Tasks with Resynthesis for Reconfigurable Systems-on-Chip // Proc. of universities. Electronics. - 2017. - Vol.22. - №3. - P. 266-275.
- [12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of State Calculations by Fast Computer Machines // J. Chemical Physics. 21. 6. June. 1953, P. 1087-1092.
- [13] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi, Optimization by Simulated Annealing // Science. 220. 1983, P.671-680.