

# Планирование топологии СБИС в условиях ограничений методами нелинейной оптимизации

Локинский С.Н., Марченко А.М., Попов Е.А., Савельев В.И., Якимов М.Ю.

Mentor Graphics Corporation,

sergey\_lokinskiy@mentor.com, alexander\_marchenko@mentor.com, evgeniy\_popov@mentor.com,  
vasily\_savelyev@mentor.com, mikhail\_yakimov@mentor.com

**Аннотация** — В работе представлен эффективный алгоритм планирования СБИС. В рамках предлагаемого подхода задача планирования элементов СБИС, представляемых прямоугольными многоугольниками, сформулирована в терминах нелинейного программирования. При этом подход позволяет учитывать любые непротиворечивые топологические и технологические требования и ограничения. В частности, поддерживаются различные типы блоков: жесткие, мягкие, с фиксированным положением; объектов трассировки: контакты, переходы, сегменты трасс, а также их относительные и абсолютные взаимные положения. Для получения топологического плана используется алгоритм нелинейной оптимизации, основанный на широко известном методе штрафных функций, в котором каждой из них соответствует некоторое топологическое или технологическое ограничение. В работе показано, что такое представление задачи планирования позволяет обеспечить гибкость при добавлении дополнительных ограничений и получить решение за приемлемое время.

**Ключевые слова** — планировка, упаковка, метод нелинейного программирования, целевая функция, штрафная функция.

## I. ВВЕДЕНИЕ

В отличие от методов планирования СБИС, использующих алгоритмы линейной оптимизации [1] или алгоритмы на графах, описанные в [2], предлагаемый подход имеет ряд преимуществ. В частности, он позволяет легко учитывать новые технологические или топологические ограничения и задавать практически любую целевую функцию. В то же время задача размещения решается без ограничений на относительное расположение топологических элементов, требуя лишь отсутствия их перекрытия. Предлагаемый подход также позволяет размещать объекты сложной прямоугольной формы, в том числе невыпуклой, наряду с обычными прямоугольниками, а также с учётом трасс между этими объектами.

В первой части статьи формулируется задача планирования СБИС, а также даются основные определения. Во второй части приводится описание алгоритма планирования. В третьей части

продемонстрирована эффективность предлагаемого алгоритма на различных примерах. Основные результаты работы представлены в заключительной части.

## II. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Задача планировки состоит в упаковке блоков и определении их формы. Одновременно может оптимизироваться длина проводов и другие параметры схемы. Кроме того, на объекты могут накладываться дополнительные ограничения в виде симметрии, относительного положения, отсутствия перекрытий и других.

В задаче рассматриваются блоки двух типов. Если фиксирована только площадь, то такие блоки называются мягкими. Если фиксированы размеры и форма – жесткими.

Жесткий блок – это произвольный прямоугольный многоугольник, примеры жестких блоков представлены на Рис. 1. Координаты жесткого блока задаются парой переменных  $(x, y)$  для геометрического центра фигуры и списком смещений для ее ребер.

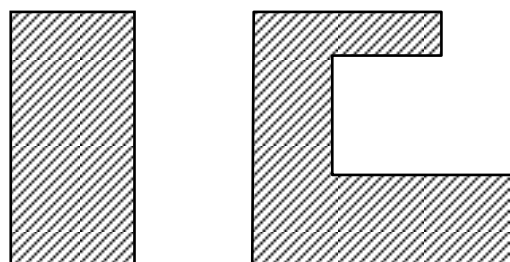


Рис. 1. Пример жестких блоков

Мягкий блок – это фигура, составленная из двух прямоугольников с одним общим ребром; частным случаем мягкого блока является прямоугольник. Для мягкого блока из примера Рис. 2 выполняются соотношения  $h_1 \cdot w_1 = h_{21} \cdot w_{21} + h_{22} \cdot w_{22}$ .

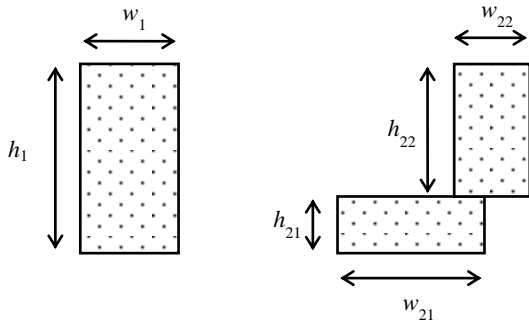


Рис. 2. Пример мягких блоков одинаковой площади

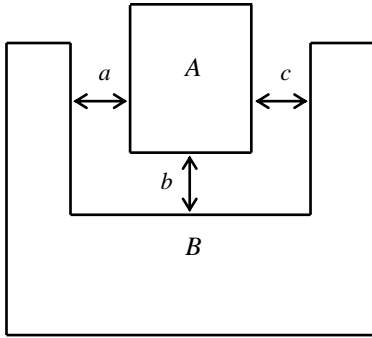


Рис. 3. Пример топологических отношений между объектами

Прямоугольный мягкий блок фиксированной площади  $S$  можно задать тремя переменными  $(x, y, w)$ , где  $x$  и  $y$  – координаты центра блока, а  $w$  – ширина блока, тогда  $h = S/w$ .

Мягкий блок сложной формы с фиксированной площадью  $S$  можно задать шестью переменными  $(x, y, w_1, w_2, \alpha, \Delta)$ , где  $x$  и  $y$  – координаты центра первой части блока,  $w_1$  и  $w_2$  – ширины первой и второй части блока,  $\alpha$  таково, что  $h_1 = \frac{S \cdot \alpha}{w_1 \cdot \alpha + w_2 \cdot (1 - \alpha)}$  и  $h_2 = \frac{S \cdot (1 - \alpha)}{w_1 \cdot \alpha + w_2 \cdot (1 - \alpha)}$ , а  $\Delta$  – задает смещение первой части блока относительно второй.

Топологические отношения между объектами  $A$  и  $B$  описываются следующими неравенствами:

- $A$  левее  $B$ ,  $(A, B, left)$ :  

$$\min(RE(A)) \leq \max(LE(B)),$$
- $A$  правее  $B$ ,  $(A, B, right)$ :  

$$\max(LE(A)) \geq \min(RE(B)),$$
- $A$  ниже  $B$ ,  $(A, B, bottom)$ :  

$$\min(TE(A)) \leq \max(BE(B)),$$
- $A$  выше  $B$ ,  $(A, B, top)$ :  

$$\max(BE(A)) \geq \min(TE(B)),$$

где  $RE(A)$  – список всех правых ребер блока  $A$ ,  $LE(A)$  – список всех левых ребер блока  $A$ ,  $BE(A)$  – список всех нижних ребер блока  $A$ ,  $TE(A)$  – список всех верхних ребер блока  $A$ .

Для примера на Рис. 3 будут выполнены следующие отношения между блоками:

- $a : (A, B, right),$
- $b : (A, B, top),$
- $c : (A, B, left).$

Для сведения проблемы планировки к задаче нелинейного программирования потребуются функции  $\min(\bar{x})$ ,  $\max(\bar{x})$ ,  $|x|$ ,  $positive(x)$ . Эти функции не являются гладкими, и вместо них будут использоваться аппроксимации с помощью логарифма суммы экспонент (LSE [3]):

$$\begin{aligned} \max'(\bar{x}) &= c \cdot \ln\left(\sum e^{x_i/c}\right), \\ \min'(\bar{x}) &= -\max'(-\bar{x}), \\ \text{abs}'(x) &= \max'(-x, x), \\ \text{pos}'(x) &= \max'(0, x), \end{aligned}$$

где  $c$  – коэффициент сглаживания, который характеризует точность аппроксимации.

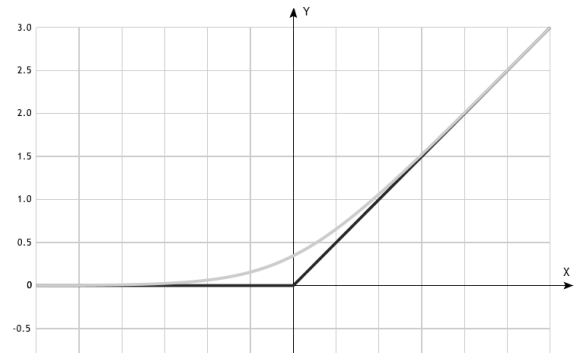


Рис. 4. Функция  $positive(x)$  и её аппроксимация  $pos'(x)$

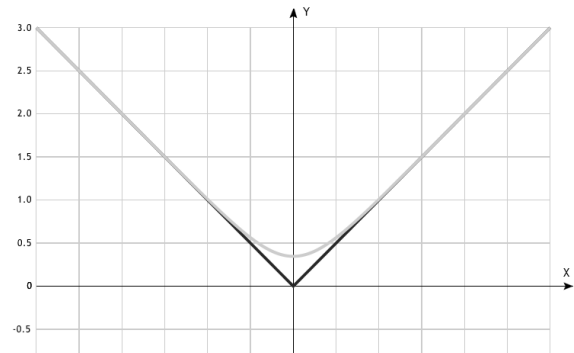


Рис. 5. Функция  $|x|$  и её аппроксимация  $\text{abs}'(x)$

На Рис. 4 и Рис. 5 показан пример аппроксимации функций  $positive(x)$  и  $|x|$ .

### III. ОПИСАНИЕ АНАЛИТИЧЕСКОГО ПЛАНИРОВЩИКА

Во время решения задачи планировки одновременно определяются положения блоков, форма мягких блоков, положения контактных площадок и трассировка. Также возможно задание различных ограничений на объекты:

- 1) задание относительного положения блоков позволяет частично или полностью зафиксировать топологический порядок объектов на плоскости;
- 2) ограничения на высоту и ширину блоков позволяют задавать необходимые минимальные размеры для корректного размещения содержимого блоков;
- 3) ограничения на размеры области планировки позволяют решать задачу с ограничениями, установленными на верхних уровнях иерархии.

Задачу планировки можно свести к набору целевых и штрафных функций. Результирующая функция оптимизации – это взвешенная сумма данных функций. Коэффициенты целевых функций фиксированы, задаются до решения задачи и регулируют соотношение между различными параметрами оптимизации. Коэффициенты штрафных функций могут меняться в процессе решения задачи для удовлетворения заданных топологических и технологических ограничений.

#### A. Целевая функция минимизации площади

$$S = (\max'(VE) - \min'(VE)) \cdot (\max'(HE) - \min'(HE)),$$

где  $VE$  – это вертикальные ребра всех объектов, а  $HE$  – горизонтальные ребра всех объектов.

#### B. Целевая функция минимизации суммарной длины проводов

$$L = \sum_{nets} (\Delta_x + \Delta_y),$$

где  $\Delta_x = (\max'(PinXs) - \min'(PinXs))^2$ ,  
 $\Delta_y = (\max'(PinYs) - \min'(PinYs))^2$ ,  $PinXs$  – это  $x$ -координаты всех контактных площадок цепи, а  $PinYs$  – это  $y$ -координаты всех контактных площадок цепи.

#### C. Целевая функция минимизации изгибов проводов

$$J = \sum_{nets} \sum_{segments} \sqrt{L_{segment}},$$

где  $L_{segment} = |L_1 - L_2|$  – длина сегмента цепи с координатами концов  $L_1$  и  $L_2$ .

#### D. Штрафная функция перекрытия блоков

Для прямоугольных блоков функция перекрытия выглядит следующим образом:

$$O = \sum_{b_1} \sum_{b_2} O_x(b_1, b_2) \cdot O_y(b_1, b_2),$$

где функции  $O_x(b_1, b_2)$  и  $O_y(b_1, b_2)$  – функции перекрытия  $x$ - и  $y$ -проекций блоков  $b_1$  и  $b_2$ :

$$O_x(b_1, b_2) = pos'(MinDist_x(b_1, b_2)),$$

$$O_y(b_1, b_2) = pos'(MinDist_y(b_1, b_2)),$$

где:

$$MinDist_x(b_1, b_2) = \min'(b_{1R}, b_{2R}) - \max'(b_{1L}, b_{2L}),$$

$$MinDist_y(b_1, b_2) = \min'(b_{1T}, b_{2T}) - \max'(b_{1B}, b_{2B}),$$

где  $b_{1L}$ ,  $b_{1R}$ ,  $b_{1B}$  и  $b_{1T}$  – соответственно левая, правая, нижняя и верхняя границы блока  $b_1$ .

В случае блоков сложной формы они разбиваются на прямоугольники, и вычисляется пересечение этих прямоугольников.

#### E. Штрафные функции максимальной ширины и высоты области упаковки

$$W = pos'(\max'(VE) - \min'(VE) - MaxW),$$

где  $MaxW$  – максимальная допустимая ширина области.

Штрафная функция максимальной высоты области упаковки  $H$  определяется аналогично.

#### F. Штрафная функция относительного положения блоков и задание жесткого относительного положения блоков

В представленном методе реализованы два способа задания относительного положения объектов.

Способ 1: с помощью штрафной функции – это мягкое ограничение, т.е. возможно его нарушение в процессе поиска решения. Штрафная функция выглядит следующим образом:

$$R = \sum_{(b_1, b_2, R)} pos'(V(b_1, b_2, r)),$$

где:

$$V(A, B, r) = \begin{cases} \min'(RE(A)) - \max'(LE(B)), & \text{если } r = left; \\ \max'(LE(A)) - \min'(RE(B)), & \text{если } r = right; \\ \min'(TE(A)) - \max'(BE(B)), & \text{если } r = bottom; \\ \max'(BE(A)) - \min'(TE(B)), & \text{если } r = top. \end{cases}$$

Способ 2: с помощью смещения координаты одного объекта относительно другого – это жесткое ограничение, которое в процессе поиска решения не нарушается. Жесткое ограничение относительного положения точек  $(A, B, left)$  выражается следующим образом:

$$B_x = A_x + \Delta^2,$$

где  $A_x$  –  $x$ -координата точки  $A$ ,  $B_x$  –  $x$ -координата точки  $B$ , а  $\Delta$  – переменная. Аналогично выражаются относительные положения  $right$ ,  $bottom$  и  $top$ . Также возможно определить жесткие отношения между блоками.

#### G. Описание алгоритма

Задача планировки разбивается на три шага:

- Начальная упаковка блоков, минимизация общей длины проводов, оптимизация положения контактных площадок.
- Построение модели проводов в соответствии с расположением блоков и контактных площадок.
- Заключительная совместная оптимизация положения блоков и трассировки.

В начале первого и третьего шага устанавливаются коэффициенты целевых функций  $\alpha_i$  и начальные коэффициенты штрафных функций  $\beta_j$ .

Результирующая функция оптимизации выглядит следующим образом:

$$F = \alpha_1 S + \alpha_2 L + \alpha_3 J + \beta_1 O + \beta_2 W + \beta_3 H + \beta_4 R.$$

Также задаются начальные коэффициенты сглаживания  $c_j$  функций аппроксимации.

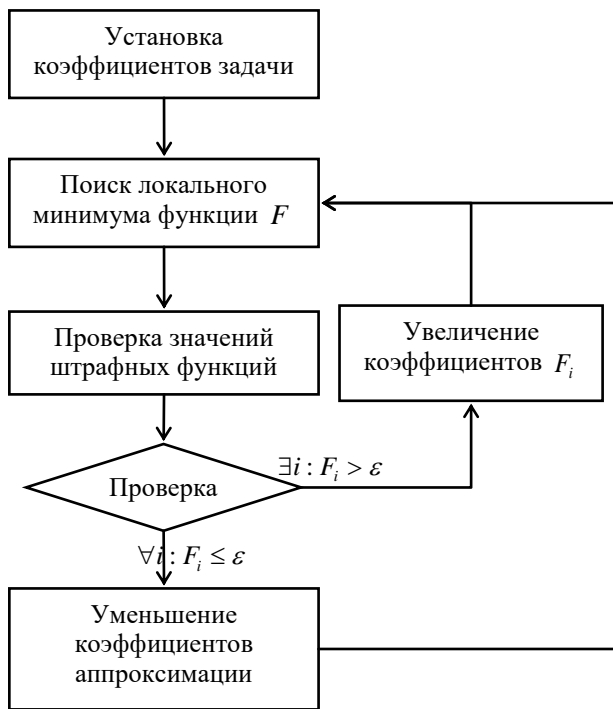


Рис. 6. Алгоритм поиска локального минимума целевой функции

Блок-схема первого и третьего шагов задачи планировки представлена на Рис. 6.

Для решения задачи поиска локального минимума функции  $F$  используется метод квази-Ньютона из библиотеки нелинейной оптимизации ТАО [4].

#### IV. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Описанный метод планировки используется в системе размещения и трассировки компании Mentor Graphics. Результаты работы алгоритма получены на сервере с процессорами 2 x Intel® Xeon® CPU E5-2680@2.70GHz, 128 Gb оперативной памяти, под управлением ОС RHEL 5. Каждый процессор имеет 8

ядер и включенную систему гиперпоточности, таким образом, можно задействовать до 32 логических ядер.

#### A. Время работы алгоритма

Время работы алгоритма определяется используемыми целевыми и штрафными функциями. Одной из самых сложных с вычислительной точки зрения является функция перекрытия блоков. На Рис. 7 приведен результат размещения 1000 прямоугольных блоков различной площади.

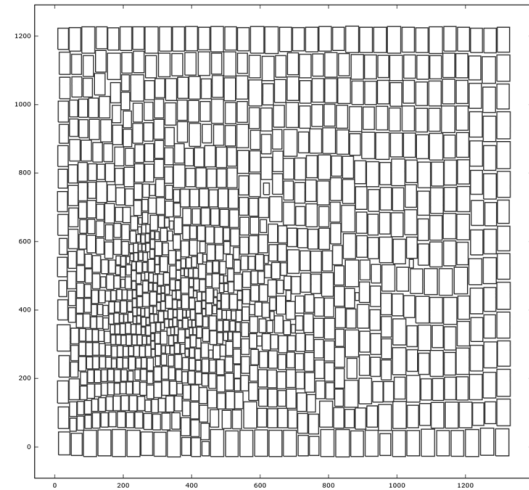


Рис. 7. Пример упаковки 1000 прямоугольников

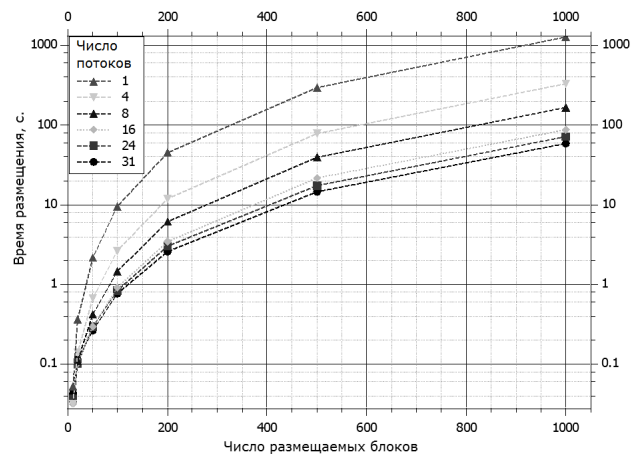


Рис. 8. Зависимость времени выполнения от количества вычислительных ресурсов и размерности задачи

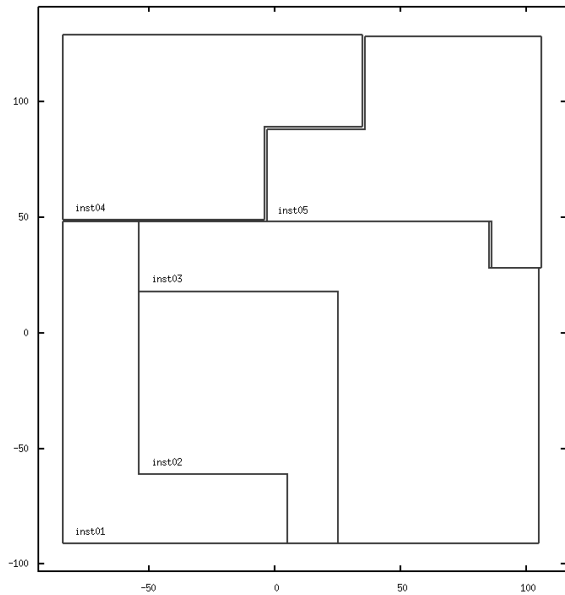
Алгоритм планировки реализован с использованием возможностей многопоточных вычислений OpenMP [5]. На Рис. 8 показана зависимость времени решения задачи от числа используемых потоков и размера задачи.

#### B. Результаты упаковки с использованием мягкого ограничения относительного положения

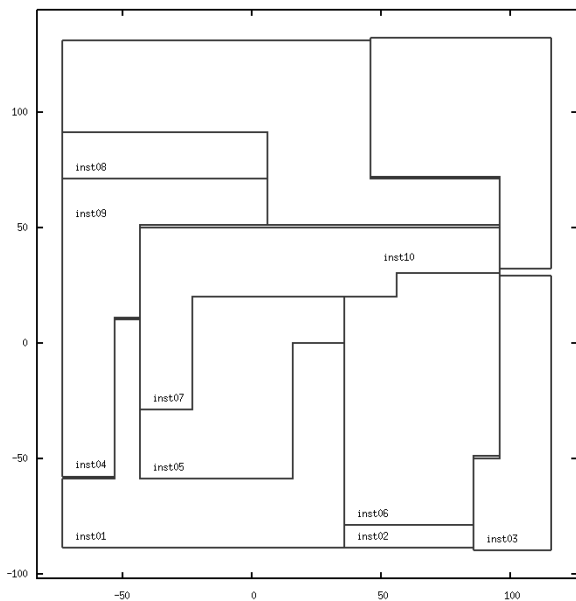
Применение ограничения взаимного расположения блоков позволяет пользователю управлять процессом планировки.

На Рис. 9 и Рис. 10 показаны примеры оптимальной упаковки 5 и 10 фигур сложной формы в квадратную

область при заданном относительном взаимном расположении.



**Рис. 9. Оптимальная упаковка пяти блоков сложной формы**



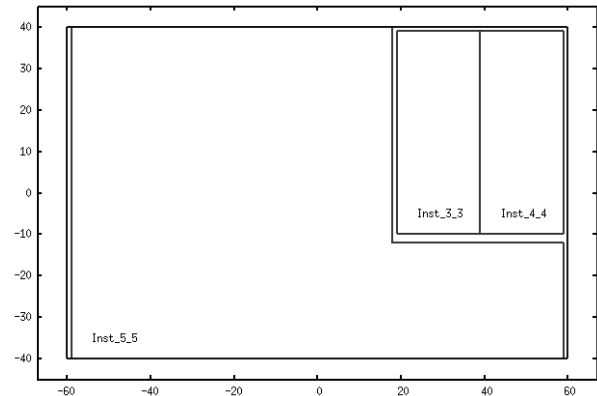
**Рис. 10. Оптимальная упаковка 2 прямоугольников и 8 сложных блоков в квадратную область**

### C. Результаты упаковки мягких блоков

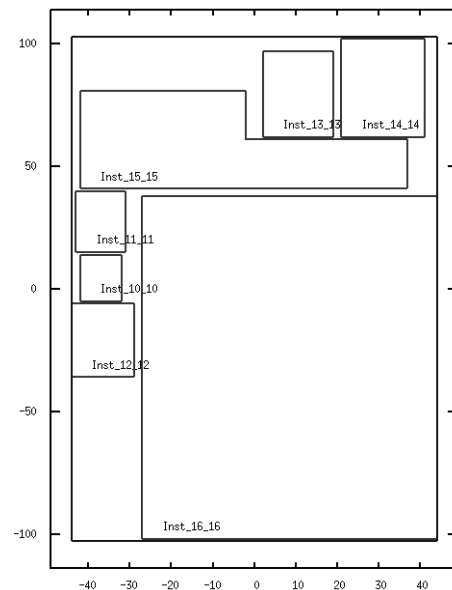
Использование мягких блоков при планировке размещения позволяет эффективно использовать ограниченную область размещения. Начальная форма мягких блоков – квадрат заданной площади.

На Рис. 11 показан результат упаковки, при которой была найдена форма мягкого блока Inst\_5\_5, обеспечивающая оптимальное использование области размещения.

На Рис. 12 показан другой пример упаковки мягких и жестких блоков в заданную прямоугольную область. Для удовлетворения ограничений на высоту и ширину области размещения граница мягкого блока Inst\_16\_16 была оптимизирована до прямоугольной формы, в то время как для мягкого блока Inst\_15\_15 удалось найти только сложную форму границы.



**Рис. 11. Упаковка одного мягкого и двух жестких блоков в заданную прямоугольную область**



**Рис. 12. Упаковка двух мягких и пяти жестких блоков в заданную прямоугольную область**

### D. Совместная оптимизация размещения и трассировки

Изложенный подход позволяет одновременно оптимизировать различные целевые функции.

На Рис. 13 и Рис. 14 показаны начальное размещение блоков и результат оптимизации положения блоков, общей длины и количества изломов проводов.

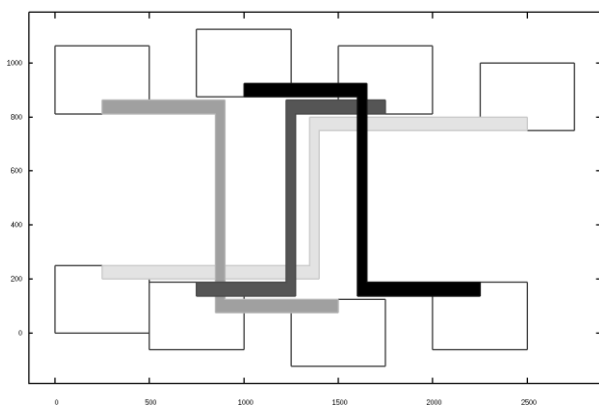


Рис. 13. Начальная расстановка объектов

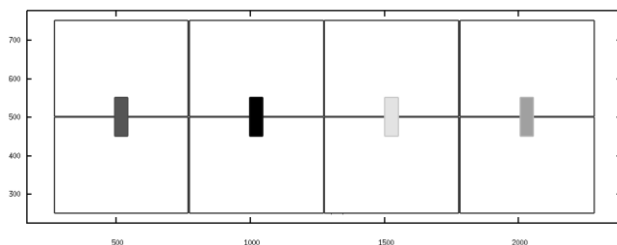


Рис. 14. Результат оптимизации размещения и трассировки

## V. ЗАКЛЮЧЕНИЕ

В работе рассмотрен метод планирования топологии СБИС в условиях топологических и технологических ограничений, в основе которого лежит решение задачи нелинейной оптимизации на множестве непрерывно дифференцируемых функций. В аналитической форме описаны наиболее часто используемые целевые функции планирования, а также некоторые виды ограничений. Предложен оригинальный алгоритм планировки, учитывающий помимо блоков также расположение контактов и трасс.

# Constraint Driven VLSI Floorplanning by Non-Linear Optimization

S. N. Lokinskiy, A. M. Marchenko, E. A. Popov, V. I. Savelyev, M. Y. Yakimov

Mentor Graphics Corporation,

sergey\_lokinskiy@mentor.com, alexander\_marchenko@mentor.com, evgeniy\_popov@mentor.com,  
vasily\_savelyev@mentor.com, mikhailevich\_yakimov@mentor.com

**Keywords** — floorplan, packing, non-linear programming, sizing, cost function, penalty function, non-convex function, smooth approximation.

## ABSTRACT

In this paper, we present an effective algorithm for VLSI floorplanning. Given any topology or technology constraints between either rectangles or complex rectilinear

polygons, representing floorplan objects, we formulate floorplanning problem as constrained NLP problem with area-like cost function. This approach can handle any consistent topological/technological constraints, such as soft/hard/preplaced blocks represented by rectilinear shapes, routing objects, such as pins, vias, routing segments, their relations and positions. Unlike linear programming, the proposed approach can be dealt with

В рамках данного алгоритма подробно описан поиск оптимального решения, базирующийся на широко известном методе штрафных функций, позволяющий решить исходную задачу с ограничениями путем рассмотрения последовательности задач без ограничений, решения которых сходятся к решению исходной задачи. Предложенный подход позволяет одновременно оптимизировать планировку по многим критериям, а также расширять список поддерживаемых оптимизационных параметров и ограничений и демонстрирует высокую производительность. Так, в работе показано, что план для 1000 прямоугольников может быть получен за время в пределах 1 минуты, что является достаточно эффективным для практического применения предложенного алгоритма. Реализация алгоритма используется в системе размещения и трассировки компании Mentor Graphics®.

## ЛИТЕРАТУРА

- [1] Pinhong Chen, Ernest S. Kuh (2000). Floorplan Sizing by Linear Programming Approximation. *DAC* (P. 468-471). Los Angeles, California: ACM.
- [2] Martin D.F. Wong, Data Structures for Analog Placement, International Workshop on Design Automation for Analog and Mixed-Signal Circuits, 2014 URL: [https://users.ece.cmu.edu/~xinli/2014\\_ams/iinde.html](https://users.ece.cmu.edu/~xinli/2014_ams/iinde.html) (дата обращения: 07.04.2016)
- [3] Log-Sum-Exp (LSE) Function and Properties on [inst.eecs.berkeley.edu](http://inst.eecs.berkeley.edu) URL: [https://inst.eecs.berkeley.edu/~ee127a/book/login/default\\_lse\\_fcn.html](https://inst.eecs.berkeley.edu/~ee127a/book/login/default_lse_fcn.html) (дата обращения: 07.04.2016)
- [4] Toolkit for Advanced Optimization on [www.mcs.anl.gov](http://www.mcs.anl.gov) URL: <http://www.mcs.anl.gov/research/projects/tat/tao-deprecated/> (дата обращения: 07.04.2016)
- [5] OpenMP description on [openmp.org](http://openmp.org) URL: <http://openmp.org/wp/> (дата обращения: 07.04.2016).

polygons, representing floorplan objects, we formulate floorplanning problem as constrained NLP problem with area-like cost function. This approach can handle any consistent topological/technological constraints, such as soft/hard/preplaced blocks represented by rectilinear shapes, routing objects, such as pins, vias, routing segments, their relations and positions. Unlike linear programming, the proposed approach can be dealt with

non-convex cost function, which is common for VLSI floorplanning tasks, and is flexible to add new topological/technological constraints in the form of specific analytical functions without changing the optimization algorithm. The optimization algorithm itself is based on well-known penalty functions method where penalty functions corresponds to topological/technological constraints. The algorithm replaces an original non-linear constrained optimization problem by a series of unconstrained non-linear problems whose solutions iteratively converge to the solution of the original constrained problem. In the paper, we have shown empirically, that our method needs not much iteration and finds the reasonable sub-optimum solution in reasonable run time.

## REFERENCES

- [1] Pinhong Chen, Ernest S. Kuh (2000). Floorplan Sizing by Linear Programming Approximation. *DAC* (P. 468-471). Los Angeles, California: ACM.
- [2] Martin D.F. Wong, Data Structures for Analog Placement, International Workshop on Design Automation for Analog and Mixed-Signal Circuits, 2014  
URL: [https://users.ece.cmu.edu/~xinli/2014\\_ams/iinde.html](https://users.ece.cmu.edu/~xinli/2014_ams/iinde.html) (accessed: 07.04.2016)
- [3] Log-Sum-Exp (LSE) Function and Properties on inst.eecs.berkeley.edu  
URL: [https://inst.eecs.berkeley.edu/~ee127a/book/login/def\\_lse\\_fcn.html](https://inst.eecs.berkeley.edu/~ee127a/book/login/def_lse_fcn.html) (accessed: 07.04.2016)
- [4] Toolkit for Advanced Optimization on [www.mcs.anl.gov](http://www.mcs.anl.gov)  
URL: <http://www.mcs.anl.gov/research/projects/tat/tao-deprecated/> (accessed: 07.04.2016)
- [5] OpenMP description on [openmp.org](http://openmp.org)  
URL: <http://openmp.org/wp/> (accessed: 07.04.2016)