

# Алгоритм исправления нарушений нанометровых топологических правил в СБИС после физического проектирования

М.С. Талалай<sup>1</sup>, Н.В. Рыженко<sup>2</sup>

АО «Интел АО», mikhail.s.talalay@intel.com<sup>1</sup>, nikolai.v.ryzhenko@intel.com<sup>2</sup>

**Аннотация** — В работе предложен алгоритм автоматической правки нарушений правил проектирования в топологии СБИС, возникающих после этапов размещения, трассировки и заполнения пустот в слоях металлизации. Алгоритм основан на решении последовательности задач Булевой выполнимости, сформулированных для объектов в прямоугольном окне вокруг нарушения. Для моделирования технологических правил вводятся ограничения в виде Булевых уравнений над дискретными объектами топологии на решётке трассировки. Исследуется структура формул, а также выявляется оптимальный вариант, позволяющий многократно уменьшить время работы алгоритма и требующую оперативную память. Экспериментальные результаты показали применимость предложенного подхода для устранения нарушений на современных промышленных СБИС.

**Ключевые слова** — правка топологических нарушений, трассировка, заполнение пустот в слоях металлизации, булева выполнимость.

## I. ВВЕДЕНИЕ

В процессе физического проектирования вся СБИС по функциональности и геометрическому положению разбивается на блоки автоматического размещения стандартных ячеек и трассировки соединений между ними (*Auto Place and Route partition, APR*). Стандартная ячейка является наименьшим компонентом иерархии, составленной из нескольких транзисторов и реализующей заданную функциональность. Блоки современных СБИС после логического синтеза могут содержать десятки и сотни тысяч стандартных ячеек. На этапе размещения назначаются координаты для стандартных ячеек в рамках площади, отведенной под данный блок. Далее в процессе трассировки контакты стандартных ячеек, а также входные и выходные порты блока соединяются проводниками, реализованными в доступных металлических слоях.

Важным завершающим этапом физического проектирования современных нанометровых СБИС является заполнение пустот в слоях металлизации (*metal fill*), оставшихся после трассировки, так как наличие таких пустот нарушает правила проектирования [1]. В результате операции заполнения пустот к уже нарисованным сигнальным прямоугольникам добавляются *fill* прямоугольники, которые делятся на два типа: плавающие (*dummy, float*) и активные (*active*). Активные прямоугольники (или антенны) примыкают к исходной

трассировке электрических цепей (рис. 1). Антенны нежелательны, поскольку увеличивают паразитную ёмкость цепей.

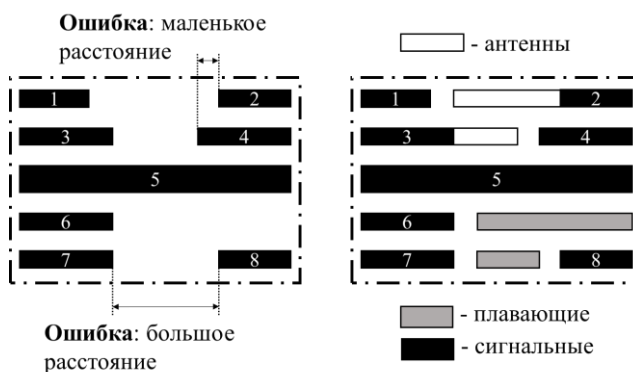


Рис. 1. Заполнения пустот в слоях металлизации

В современных САПР трассировка и процедура заполнения пустот разделены: после трассировки осуществляется заполнение слоев металлизации. Данное разделение этапов приводит к определённым проблемам. Трассировщик не гарантирует возможность заполнить по правилам площадь рядом с созданными сигнальными прямоугольниками. В свою очередь программа, заполняющая пустоты слоёв металлизации, зачастую не может теоретически (или в силу особенностей реализованного алгоритма) создать необходимые прямоугольники без нарушений. Проблема ярко выражена на плотно загруженных трассировкой слоях и на блоках с сильно ограниченной доступной площадью для проектирования. Ещё более осложняет проблему неуклонный рост числа правил проектирования и увеличение их сложности [2, 3]; длина волны в оптической литографии уже много лет остаётся неизменной (193 нм), а требуемая точность печати измеряется десятками нанометров [4].

Среди методов борьбы с указанными нарушениями наиболее универсальным и действенным является увеличение размера блока, а затем новое размещение, трассировка и заполнение пустот в слоях металлизации. Данная процедура помогает понизить число нарушений вплоть до нуля, поскольку с ростом размера блока одновременно уменьшается плотность контактов ячеек, и увеличивается трассировочный ресурс. Однако большая площадь СБИС увеличивает стоимость конечного продукта. Чем больше схема, тем меньше устройств помещается на круглой полупроводниковой

пластине кремния фиксированного размера. Поэтому инженеры определяют пороговый уровень числа нарушений, при котором экономически обоснованно устранять их вручную, без увеличения площади блока в частности, и СБИС в целом.

Целью данной работы является разработка подхода для устранения нарушений, возникающих после трассировки блока и заполнения слоев металлизации. Требования к автоматизированному решению следующие.

1) Алгоритм должен находить решение, если такое решение существует теоретически. 2) Среди возможных решений алгоритм должен находить минимально возможные изменения в исходной топологии трассировки. 3) Решение должно быть пригодным для разных технологических процессов, поэтому алгоритм и абстракция представления правил должны быть разделены.

В силу объективных причин, таких как размер входных данных, промышленные САПР топологии СБИС на уровне блока реализуют эвристические алгоритмы. Условия же локальности проблемного места позволяют развивать подход на основе точных комбинаторных методов. По этим причинам в данной работе рассматриваются подходы на основе решения задач Булевой выполнимости (*ВЫП*, *Boolean Satisfiability*, *SAT*). Правила проектирования моделируются в виде Булевых выражений для дискретных объектов топологии на решётке трассировки [12]. В данной работе также проведено исследование на предмет поиска оптимального вида формул, кодирующих правила проектирования, с точки зрения скорости создания и решения выполнимости.

Содержание работы следующее. В главе 2 представлены место задачи Булевой выполнимости в САПР СБИС и особенности использования для решения поставленной задачи. В главе 3 описана общая схема автоматизированного исправления ошибок на *APR* блоке, а также разобраны вспомогательные этапы. В главе 4 рассмотрен способ моделирования правил проектирования в виде формул над объектами топологии. В главе 5 описаны основные алгоритмы: инкрементальное изменение трассировки, задача выполнимости для заполнения пустот на фиксированной трассировке, а также способ минимизации антенн. Экспериментальные результаты и заключение представлены в главах 6 и 7.

## II. БУЛЕВА ВЫПОЛНИМОСТЬ В ЗАДАЧАХ САПР

Достижения последних лет в области ВЫП решателей позволяют их успешно использовать при проектировании СБИС [5-11]. В [5] задача выполнимости дополняет эвристический алгоритм трассировки схем с регулярной топологией. В [6] ВЫП решатель используется для размещения транзисторов внутри ячейки. В [7] ВЫП используется для трассировки программируемых матриц, в [8] – для трассировки регулярных логических блоков. В [9] с помощью ВЫП решается задача устранения топологий, нежелательных для оптической литографии, при этом начальная трассировка модифицируется за счёт перебора возможных вариантов. В [10] и [11] задача ВЫП используется для трас-

сировки и оптимизации стандартных ячеек. Главной отличительной особенностью ВЫП решателя является то, что он находит допустимое решение (если такое существует) или же определяет, что для заданных ограничений решения не существует теоретически.

Естественным ограничением при формулировании задачи ВЫП для топологии СБИС является число объектов, кодируемых переменными. Поэтому, как правило, рассматривается ограниченная по площади конструкция: небольшое окно [9] или стандартная ячейка [6, 10, 11]. Также число и структура Булевых ограничений, связывающих переменные, важны для эффективного применения ВЫП решателя. Например, регулярные (повторяющиеся) топологические конструкции [5, 7, 8] инициируют повторяющиеся по структуре формулы Булевых ограничений. Обычно такие задачи решаются быстро современными решателями, несмотря на миллионы вовлеченных переменных и ограничений.

Отличительной особенностью предложенного в данной работе решения на основе ВЫП решателя от существующих алгоритмов трассировки ограниченных областей является то, что алгоритм ищет минимальные изменения в исходной топологии, что крайне важно с точки зрения электрических характеристик схемы. Для этого формулируется последовательность локальных изменений в исходной трассировке и соответствующая им последовательность задач Булевой выполнимости. Каждая из задач формулирует условие для правильного заполнения слоев металлизации в рамках текущего варианта трассировки в проблемном месте. Решение такой задачи (если оно существует) соответствует правильной топологии. Такая схема последовательных изменений в трассировке в худшем случае перебирает все возможные варианты изменений, начиная с минимальных.

## III. ОБЩАЯ СХЕМА АЛГОРИТМА

Способ автоматического исправления нарушений правил проектирования на отдельно взятом *APR* блоке разбит на этапы. Предполагается, что в топологии блока построена трассировка и заполнены пустоты в слоях металлизации (рис. 2а, для простоты показан один слой металлизации). Далее ищутся координаты нарушений с локальными прямоугольными областями (окнами) вокруг (рис. 2б). Пересекающиеся окна объединяются в единую область (рис. 2в). Далее в каждом окне производятся попытки исправить топологию через инкрементальные изменения в трассировке с заполнением пустот (см. главу 5). Завершающим этапом процедуры являются вставка правильных топологий в исходный блок и поиск оставшихся нарушений (рис. 2г).

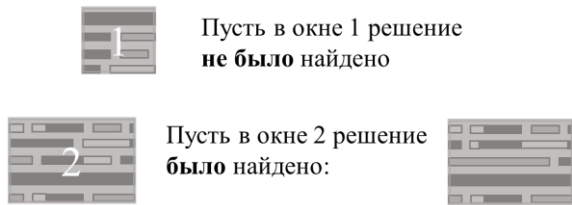
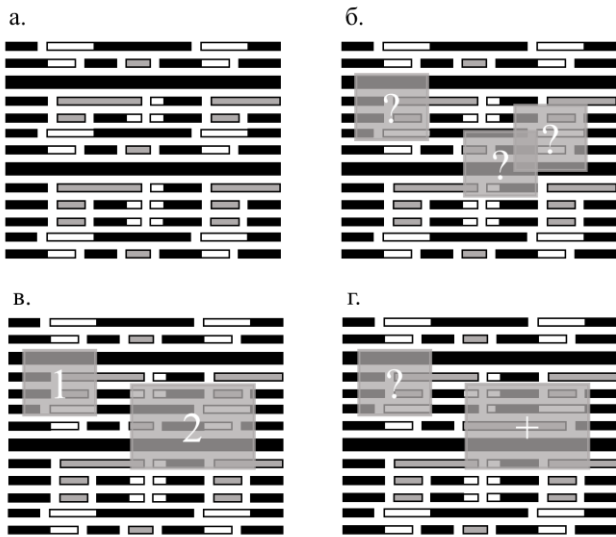


Рис. 2. Общая схема для исправления нарушений

Объединение пересекающихся окон необходимо для независимой работы с каждым отдельным окном. Без такого объединения новая топология в окне *A* повлияет на входную задачу пересекающегося с ним окна *B*. Геометрические объекты в уже разрешенном окне *A* должны оставаться без изменений при работе с окном *B*. Это, в свою очередь, с высокой вероятностью уменьшит свободу в поиске правильной топологии для окна *B*. Если же нарушение в *B* находится очень близко к границе окна *A*, то, как правило, его исправить будет невозможно теоретически. В конечном итоге такой подход может привести к увеличению числа нарушений на всем блоке, а не к уменьшению. Поэтому целесообразней искать правильную топологию одновременно для всех пересекающихся окон, т.е. построив единое окно.

Перед исправлением отдельно взятого проблемного места в центре окна все *fill* прямоугольники удаляются, а на границах с отступом сохраняются, возможно, с обрезанием по отступу. Все сигнальные прямоугольники остаются без изменений (рис. 3). Удаленные прямоугольники освобождают место для инкрементальных изменений в трассировке. В дальнейшем все изменения внутри окна будут выстраиваются таким образом, чтобы не возникало нарушений в центре, а также на участках соприкосновения с «замороженной» топологией. Так как топология на границе не содержит нарушений по построению, то это гарантирует вставку в исходный блок окна без нарушений. Размер необходимого отступа для «заморозки» определяется правилами проектирования, которые в силу своей локальности формулируются для топологических объектов, распо-

ложенных в ограниченной области. В качестве отступа выбирается длина, которая гарантирует, что новая правильная топология внутри окна (в «незамороженной» области) не связана напрямую правилом с топологией за пределами окна.

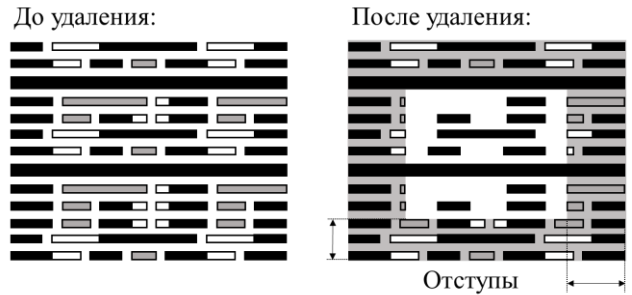


Рис. 3. Подготовка топологии окна для инкрементальных изменений в топологии

#### IV. МОДЕЛЬ ПРАВИЛ ПРОЕКТИРОВАНИЯ

Геометрические правила проектирования формулируются для инженера в виде некоторого текста с рисунками, где объясняется какая топологическая конструкция является правильной, а какая запрещенной. Различные промышленные САПР предполагают более формализованный способ описания таких правил. Обычно это список входных переменных для программы. Выставляя значения для этих параметров, инженер кодирует технологические правила. Для современных нанометровых технологических процессов далеко не всегда представляется возможным перевести правила из текстового описания в значения параметров программы. Поэтому зачастую подбираются усиленные значения параметров, которые гарантируют выполнение правил проектирования, но ограничивают дополнительно пространство решений. Либо формулируются ослабленные требования с идеей правки возможных нарушений после запуска промышленных САПР. В последнем случае это также потенциально ведет к появлению нарушений, обсуждаемых в данной статье.

В данной работе правила моделируются в виде набора запрещенных комбинаций-шаблонов над топологическими объектами в заданной сетке [12]. Для этого вводится решетка, составленная из вертикальных и горизонтальных линий. С узлами решетки (с точками пересечения линий) ассоциируются атомарные дискретные объекты. Каждый такой объект получает координаты на решетке и может находиться в состоянии «*must be present*» (*P*) или «*must be absent*» (*A*), что означает «должен присутствовать» и «должен отсутствовать» соответственно. Множество запрещенных комбинаций в топологии кодируется в виде логического выражения над дискретными объектами с указанием состояний и координат для объектов. При этом важны не сами координаты, а их взаимное расположение относительно друг друга. Т.е. шаблон справедлив для произвольных сдвигов в решетке, поворотов и отражений. Предполагается, что в правильной топологии, которая не нарушает правила проектирования, данные

логические выражения принимают значение «ЛОЖНО» в каждом месте топологии, где данный шаблон применим.

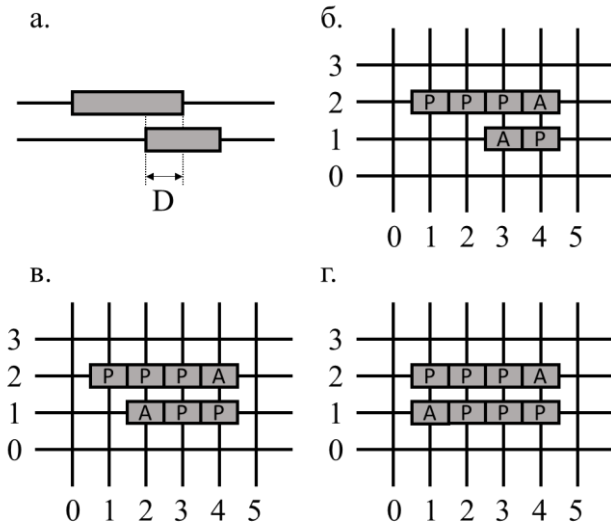


Рис. 4. Пример шаблона, кодирующего правило

Пусть правило проектирования звучит следующим образом: «в слоях металлизации длина области наложения прямоугольников в противоположных направлениях не должна быть менее, либо равной, заданной длины  $D$ » (рис. 4а). Пусть на введенной решетке  $D = 2$ . На рис. 4б-г показаны запрещенные комбинации с указанием позиций атомарных объектов и состояний:  $P$  и  $A$ . Тогда, с учетом того, что  $P_{i,j} = \overline{A_{i,j}}$ , общая формула для данного правила выглядит так:

$$P_{1,2} \wedge P_{2,2} \wedge P_{3,2} \wedge A_{4,2} \wedge P_{4,1} \wedge \overline{(P_{3,1} \wedge P_{2,1} \wedge P_{1,1})}.$$

Данная модель правил позволяет формулировать Булевы ограничения для переменных, кодирующих топологию, тем самым формировать пространство всех правильных топологий. Для этого каждому дискретному объекту, ассоциированному с узлами решетки, ставится в соответствие Булева переменная. Состояние  $P$  соответствует самой переменной, а состояние  $A$  соответствует отрицанию переменной. Решатель задачи выполнимости требует в качестве входа Булеву формулу в виде КНФ (Конъюнктивная Нормальная Форма):

$$F(x_1, x_2, \dots, x_n) = \bigwedge_{i=1}^m \bigvee_{j=1}^n l^i(x_j),$$

где  $m$  – число конъюнкций, а  $l^i(x_j)$  является либо переменной  $x_j$ , либо ее отрицанием  $\overline{x_j}$ , либо логическим нулем. На выходе решатель выдает произвольную комбинацию значений  $(\alpha_1, \alpha_2, \dots, \alpha_n)$ ,  $\alpha_i \in \{0,1\}$ ,  $i = 1, \dots, n$  для переменных  $(x_1, x_2, \dots, x_n)$ , для которой  $F(\alpha_1, \alpha_2, \dots, \alpha_n) = 1$ , если такая комбинация существует.

Для добавления произвольного логического выражения в КНФ существует ряд математических приемов. В [10, 11] используются варианты с добавлением

новой переменной. Например, для того, чтобы добавить ограничение, выраженное в виде

$$x_1 \wedge x_2 \wedge \dots \wedge x_n = 0,$$

вводится дополнительная переменная  $p$ , и КНФ дополняется множителями:

$$(p \vee \overline{x_1} \vee \dots \vee \overline{x_n}) \wedge (\overline{p} \vee x_1) \wedge \dots \wedge (\overline{p} \vee x_n) \wedge \overline{p}.$$

Такой прием гарантирует, что выполнимость исходной КНФ возможна тогда и только тогда, когда

$$x_1 \wedge x_2 \wedge \dots \wedge x_n = p, \text{ и } p = 0.$$

Аналогичный прием существует для перевода выражений вида:

$$x_1 \vee x_2 \vee \dots \vee x_n = 0.$$

Вводится дополнительная переменная  $q$ , и КНФ дополняется множителями:

$$(\overline{q} \vee x_1 \vee \dots \vee x_n) \wedge (q \vee \overline{x_1}) \wedge \dots \wedge (q \vee \overline{x_n}) \wedge \overline{q}.$$

Также гарантируется, что выполнимость исходной КНФ возможна тогда и только тогда, когда

$$x_1 \vee x_2 \vee \dots \vee x_n = q, \text{ и } q = 0.$$

На данных базовых примерах перевод в КНФ можно упростить, т.к. наличие множителя  $\overline{p}$  и  $\overline{q}$  позволяет подставить 0 вместо  $p$  и  $q$  в соответствующих выражениях, так как только в этом случае формула выполнима. Тогда в результате получится:

$$(\overline{x_1} \vee \dots \vee \overline{x_n}) \text{ и } (x_1) \wedge \dots \wedge (x_n).$$

Такой прием иллюстрирует способ перевода выражений общего вида в КНФ для ВПП решателя. Рассмотрим ограничение:

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee x_5 = 0.$$

Для перевода данного выражения в КНФ для ВПП решателя можно ввести переменные:  $p_1, p_2, q$ , и последовательно составить набор новых множителей для исходной КНФ:

$$(p_1 \vee \overline{x_1} \vee \overline{x_2}) \wedge (\overline{p_1} \vee x_1) \wedge (\overline{p_1} \vee x_2),$$

$$(p_2 \vee \overline{x_3} \vee \overline{x_4}) \wedge (\overline{p_2} \vee x_3) \wedge (\overline{p_2} \vee x_4),$$

$$(\overline{q} \vee p_1 \vee p_2 \vee x_5) \wedge (q \vee \overline{p_1}) \wedge (q \vee \overline{p_2}) \wedge (q \vee \overline{x_5}) \wedge \overline{q}.$$

При этом будет гарантироваться, что выполнимость всей КНФ возможна тогда и только тогда, когда:

$$p_1 = x_1 \wedge x_2,$$

$$p_2 = x_3 \wedge x_4,$$

$$q = p_1 \vee p_2 \vee x_5, \text{ и } q = 0.$$

Скорость ВПП решателя зависит от длины КНФ и от числа переменных в ней. При использовании приемов с добавлением новых переменных окончательный вид КНФ для ВПП решателя зависит от выражений, которые задают ограничения. Такие выражения могут быть записаны в очень компактной форме, но требуют много дополнительных переменных. Или могут иметь большую длину, но требовать меньше дополни-

тельных переменных. В рамках данного исследования также ставился вопрос о поиске оптимального вида формул для описания правил и перевода их в КНФ для ВПП решателя. Работа над структурой формул осуществлялась известными открытыми программами для минимизации и логического синтеза булевых формул: *espresso* [13] и *abc* [14]. Были рассмотрены: 1) максимально компактные формулы, представляющие Булеву сеть общего вида; 2) формулы, представляемые деревьями; 3) «двухуровневые» формулы, такие как ДНФ и КНФ, а также 4) специальный вариант без дополнительных переменных. В качестве ВПП решателя использовался *minisat* [15]. Значительное ускорение при построении задачи (т.е. перевод ограничений в КНФ), а также экономия по памяти, были выявлены при использовании формул без дополнительных переменных. Этого можно добиться путем построения всех ограничений в виде инвертированной КНФ:

$$F(x_1, x_2, \dots, x_n) = \overline{\bigwedge_{i=1}^m \bigvee_{j=1}^n l^i(x_j)},$$

где  $m$  – число конъюнкций, а  $l^i(x_j)$  является либо переменной  $x_j$ , либо ее отрицанием  $\overline{x_j}$ , либо логическим нулем. В таком виде каждая переменная в задаче кодирует топологию схемы, а любое ограничение переводится в КНФ путем простой операции отбрасывания отрицания и добавки всех формул без изменений в структуре. Это следует из того, что по определению каждое такое выражение описывает неправильную топологию (нужно, чтобы оно получило значение 0 в решении). Т.е. ограничение в общем виде выглядит:

$$\overline{\bigwedge_{i=1}^t \bigvee_{j=1}^n l^i(x_j)} = 0.$$

А задача для ВПП решателя должна быть в виде:

$$\bigwedge_{i=1}^m \bigvee_{j=1}^n l^i(x_j) = 1.$$

Данный вид формул – инвертированная КНФ – является каноническим, т.е. любая Булева функция может быть записана эквивалентной формулой в таком виде. Для получения таких формул использовался программный пакет *espresso*, с помощью которого строилась компактная таблица истинности для каждого выражения (ограничения). Из таблицы истинности выписывалась КНФ для функции отрицания данного выражения по стандартным правилам. А в завершение добавлялось отрицание для получения формулы для самой функции. Такой вариант моделирования правил проектирования оптимальным образом подходит для использования ВПП решателя для задачи заполнения пустот в слоях металлизации, рассмотренной в следующей главе.

## V. ИСПРАВЛЕНИЕ ТОПОЛОГИИ ОКНА

Исправление топологии в центре окна реализовано в виде последовательности изменений исходной трассировки. После каждого изменения с помощью ВПП решателя алгоритм пробует построить правильное заполнение пустот в слоях металлизации в условиях новой геометрии сигнальных проводов. Если это удастся, то алгоритм завершает работу на данном окне. Если же нет, то трассировка изменяется снова.

Перед построением первого изменения проверяется возможность построить правильное заполнение слоев металлизации для исходной трассировки. Значительная часть окон может быть исправлена уже на этом этапе. Такая ситуация зачастую возникает в период, когда технология совсем новая, и разработчики не успели адаптировать алгоритмы под новые правила. Если же правильного заполнения не существует для исходной трассировки, то определяется набор прямоугольников в слоях металлизации, которые полностью находятся внутри «незамороженной» части окна и могут быть сдвинуты без изменений в «замороженной» части. Далее, путем последовательно перебора, начиная с ближайших к нарушению прямоугольников, осуществляются сдвиги таких прямоугольников в соседние трассировочные треки. При этом также перемещаются и межслойные переходы, касающиеся сдвигаемого прямоугольника, а также «растягиваются» или «сжимаются» прямоугольники в соседних слоях, связанные с данным через межслойные переходы (рис. 6). Сначала алгоритм пробует сдвинуть только один прямоугольник, потом одновременно два и так далее до попыток сдвинуть все прямоугольники одновременно или до установленного ограничения.

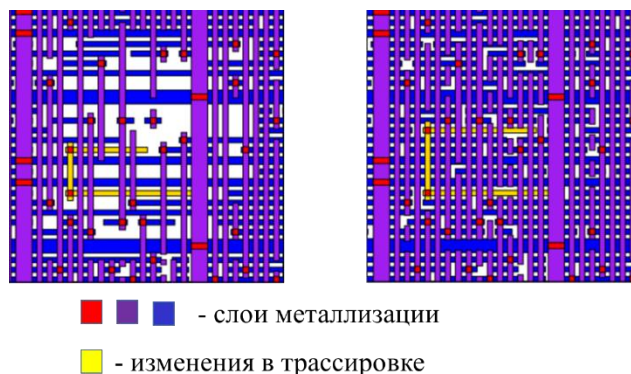


Рис. 6. Пример успешного изменения в трассировке

Для новой геометрии трассировки формулируется задача ВПП на правильное заполнение следующим образом. Для каждого атомарного прямоугольника в слое металлизации на введенной ранее сетке (см. главу 4) ставится в соответствие Булева переменная, кодирующая его наличие или отсутствие в правильной топологии. При этом для атомарных прямоугольников, соответствующих построенной трассировке, переменные не вводятся, так как процедура заполнения пустот не удаляет трассировку. КНФ для решателя ВПП строится путем перевода формул из библиотеки шаблонов, кодирующих правила проектирования таким образом, как показано в главе 4. Также добавляются ограничения, гарантирующие то, что новые прямоугольники не соединят разные электрические цепи.

Если решение задачи заполнения слоев металлизации найдено, то строится последовательность задач второго типа для минимизации длин антенн. Данная последовательность задач реализует механизм двойного поиска минимального значения для максимальной длины антенны. Для этого в полученном решении

определяется максимальная длина антенны  $A_{max}$ . Также вводится переменная  $A_{min} := 0$ , которая задает нижнюю границу поиска. Далее, в текущую задачу добавляются ограничения, гарантирующие, что длина антенн во всех местах не превышает значение  $A := (A_{max} + A_{min})/2$  и запускается решатель. Если решение найдено для такой задачи, то верхняя граница уменьшается:  $A_{max} := A$ ; если решения не существует, то увеличивается нижняя граница:  $A_{min} := A$ . В результате, когда границы поиска будут отличаться на единицу, процедура останавливается, а решение последней решенной задачи берется в качестве окончательного.

## VI. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Рассмотренная в статье схема исправления нарушений правил проектирования после этапов трассировки и заполнения пустот в слоях металлизации на уровне APR блока была реализована на языках C++ и tcl. Программа применяется в компании Intel при проектировании новейшей микроэлектроники и особенно востребована для СБИС класса Система на Кристалле. Разработка именно таких продуктов характеризуется сжатыми сроками по проектированию в целом и, в частности, по выполнению подобных исправлений. Существует развернутая статистика по типу нарушений, проценту успешных исправлений, зависимости качества работы от размера окна и др. для разных СБИС, спроектированных для разных технологических процессов. Однако специфика коммерческой организации не позволяет раскрыть эту информацию. Можно отметить, что программа работает значительно быстрее и точнее, чем группа инженеров, редактирующих топологию вручную, и позволяет автоматически исправить в среднем 80% проблемных мест. Оставшиеся 20% проблемных мест, как правило, требуют глобальных изменений для исправления, включая изменения позиций стандартных ячеек.

## VII. ЗАКЛЮЧЕНИЕ

Рассмотренная в статье схема для автоматического исправления нарушений правил проектирования СБИС разработана для сокращения времени проектирования. Особенности контекста, требования по времени работы и памяти определили выбор алгоритмов. Так, требование высокого качества и локальность проблемных мест позволило применять ВЬП решатель, что дает возможность достичь наилучшего качества. По причинам того, что важно находить минимальное изменение в трассировке за короткое время, выбран инкрементальный подход. На практике показано, что если правильная трассировка с заполнением слоев металлизации существует, то обычно алгоритм ее находит за первые 10 итераций. Для сравнения, использование ВЬП решателя для полной перетрассировки окна в топологии требует в десятки раз больше времени и памяти. В связи с требованиями на разумное время работы особое внимание уделено также структуре за-

дачи, которая формулируется для ВЬП решателя, а именно то, какой вид формул использовать для моделирования правил и как их оптимально переводить в КНФ для ВЬП решателя.

В результате данного исследования была реализована программа, которая применяется в компании Intel при проектировании новейшей микроэлектроники.

## ЛИТЕРАТУРА

- [1] L. Deng, K. Chao, H. Xiang, M. D. F. Wong. Coupling-aware dummy metal insertion for lithography // Proc. of the IEEE Asia South Pacific Des Automation Conference, 2007.
- [2] D. Z. Pan. Lithography friendly routing: from construct-by-correction to correct-by-construction // Proc. of the 21st annual symposium on ICs and system design, 2008.
- [3] Y. Du, Q. Ma, H. Song, J. Shiely, G. Luk-Pat, A. Miloslavsky, and M. D. F. Wong. Spacer-is-dielectric-compliant detailed routing for selfaligned double patterning lithography // Proc. of the 50th Annual Design Automation Conference, pages 93:1–93:6, 2013.
- [4] K. Ronse, P. Jansen, R. Gronheid, E. Hendrickx, M. Maenhoudt, V. Wiaux, A.-M. Goethals, R. Jonckheere, and G. Vandenberghe. Lithography Options for the 32 nm Half Pitch Node and Beyond // IEEE Tran. on Circuits and Systems - I: Regular papers, Vol. 56, No. 8, August 2009.
- [5] Lin Y.-W., Marek-Sadowska M., Maly W. Transistor-level layout of highdensity regular circuits // Proc. of the international symposium on Physical design, 2009.
- [6] T. Iizuka, M. Ikeda, K. Asada. High Speed Layout Synthesis for MinimumWidth CMOS Logic Cells via Boolean Satisfiability // Proc. of the Asia and South Pacific Design Automation Conference, 2004.
- [7] Nam G.-J. et al, Satisfiability-Based Layout Revisited: Detailed Routing of Complex FPGAs via Search-Based Boolean SAT // Proc. of the ACM/SIGDA 7th international symposium on Field programmable gate arrays, 1999.
- [8] B. Taylor, L. Pileggi. Exact Combinatorial Optimization Methods for Physical Design of Regular Logic Bricks // Proc. of the Design Automation Conference, 2007.
- [9] F. Yang, Y. Cai, Q. Zhou, J. Hu. SAT Based Multi-Net Rip-up-and-Reroute for Manufacturing Hotspot Removal // Proc. of the Conference on Design, Automation and Test in Europe, 2010.
- [10] Ryzhenko N., Burns S. Standard cell routing via boolean satisfiability // Proc. of the 49th Annual Design Automation Conference, 2012.
- [11] Ryzhenko N.V., Sorokin A.A., Bykov S.A., Talalay M.S. Minimization of undesired layout patterns during standard cell synthesis // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part I. P. 121-126.
- [12] Suto G. Rule agnostic routing by using design fabrics // Proc. of the 49th Annual Design Automation Conference, 2012.
- [13] URL: [https://en.wikipedia.org/wiki/Espresso\\_heuristic\\_logic\\_minimizer](https://en.wikipedia.org/wiki/Espresso_heuristic_logic_minimizer) (дата обращения: 1.04.2016)
- [14] URL: <http://www.eecs.berkeley.edu/~alanmi/abc/abc.htm> (дата обращения: 1.04.2016)
- [15] URL: <http://www.dwheeler.com/essays/minisat-user-guide.html> (дата обращения: 1.04.2016)

# Algorithm for design rule violation clean-up after physical design

M.S. Talalay<sup>1</sup>, N.V. Ryzhenko<sup>2</sup>

Strategic CAD Labs, Intel Corporation,

mikhail.s.talalay@intel.com<sup>1</sup>, nikolai.v.ryzhenko@intel.com<sup>2</sup>

**Keywords** — DRV clean-up, routing, metal fill, SAT solver.

## ABSTRACT

This paper describes the approach for automated Design Rule Violation (DRV) clean-up of layout after placement, routing and metal fill physical design stages. After routing there are always empty regions in tracks that were not used to implement connections. Such empty regions violate design rules. Metal fill [1] is one of the final steps in physical design which aims to fill empty regions and to fix such violations. After this procedure new float or active fill wires can appear. Float or dummy wires are not connected to any device, active fill wires are the extensions of existing wires. The complexity and number of design rules only grows with each new tech node [2]-[3], and the metal fill process becomes more and more complicated. Routing and metal fill flows are separated: first routing is built and then metal fill is created. It leads to the fact that router can create theoretically unfillable topologies. To fix such places designer edits manually the routing topology near problem place, so that after that it will become fillable. Manual cleanup becomes a bottleneck in physical design. In this paper the automated solution is proposed.

The input of the flow is the Auto Place and Route (APR) block after routing and metal fill, with found DRVs. First, layout windows around problem places are cut from initial block; then tool automatically searches the minimal change in routing topology, so that it will become fillable on each window; and at the last step tool pushes back each fixed window to initial block. The locality of the problem places allow to apply SAT solver which is precise and rather popular today for layout tasks [3]-[11]. The main difference of the proposed approach in comparison to rerouting from scratch of the local areas is that tool builds minimal changes in existing routing topology. This property is important as it helps to mitigate the influence on electrical characteristics of the circuit. It is implemented through the sequence of SAT tasks formulated for discrete layout objects in rectangular local window. Design rules are modeled with Boolean expressions on discrete objects placed on routing grid [12]. In this paper the optimal way to represent such Boolean constraints is proposed to speed up significantly building and solving of SAT tasks for metal fill.

Experiments showed that in average 80% of DRVs can be fixed automatically, other DRVs are not localized and can be fixed only on block level. The solution is used for designing of new microelectronics at Intel Corporation.

## REFERENCES

- [1] L. Deng, K. Chao, H. Xiang, M.D.F. Wong. Coupling-aware dummy metal insertion for lithography // Proc. of the IEEE Asia South Pacific Des Automation Conference, 2007.
- [2] D. Z. Pan. Lithography friendly routing: from construct-by-correction to correct-by-construction // Proc. of the 21st annual symposium on ICs and system design, 2008.
- [3] Y. Du, Q. Ma, H. Song, J. Shiely, G. Luk-Pat, A. Miloslavsky, and M. D. F. Wong. Spacer-is-dielectric-compliant detailed routing for selfaligned double patterning lithography // Proc. of the 50th Annual Design Automation Conference, pages 93:1–93:6, 2013.
- [4] K. Ronse, P. Jansen, R. Gronheid, E. Hendrickx, M. Maenhoudt, V. Wiaux, A.-M. Goethals, R. Jonckheere, and G. Vandenberghe. Lithography Options for the 32 nm Half Pitch Node and Beyond // IEEE Tran. on Circuits and Systems - I: Regular papers, Vol. 56, No. 8, August 2009.
- [5] Lin Y.-W., Marek-Sadowska M., Maly W. Transistor-level layout of highdensity regular circuits // Proc. of the international symposium on Physical design, 2009.
- [6] T. Iizuka, M. Ikeda, K. Asada. High Speed Layout Synthesis for MinimumWidth CMOS Logic Cells via Boolean Satisfiability // Proc. of the Asia and South Pacific Design Automation Conference, 2004.
- [7] Nam G.-J. et al, Satisfiability-Based Layout Revisited: Detailed Routing of Complex FPGAs via Search-Based Boolean SAT // Proc. of the ACM/SIGDA 7th international symposium on Field programmable gate arrays, 1999.
- [8] B. Taylor, L. Pileggi. Exact Combinatorial Optimization Methods for Physical Design of Regular Logic Bricks // Proc. of the Design Automation Conference, 2007.
- [9] F. Yang, Y. Cai, Q. Zhou, J. Hu. SAT Based Multi-Net Rip-up-and-Reroute for Manufacturing Hotspot Removal // Proc. of the Conference on Design, Automation and Test in Europe, 2010.
- [10] Ryzhenko N., Burns S. Standard cell routing via boolean satisfiability // Proc. of the 49th Annual Design Automation Conference, 2012.
- [11] Ryzhenko N.V., Sorokin A.A., Bykov S.A., Talalay M.S. Minimization of undesired layout patterns during standard cell synthesis // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part I. P. 121-126.
- [12] Suto G. Rule agnostic routing by using design fabrics // Proc. of the 49th Annual Design Automation Conference, 2012.
- [13] URL: [https://en.wikipedia.org/wiki/Espresso\\_heuristic\\_logic\\_minimizer](https://en.wikipedia.org/wiki/Espresso_heuristic_logic_minimizer) (access date: 1.04.2016)
- [14] URL: <http://www.eecs.berkeley.edu/~alanmi/abc/abc.htm> (access date: 1.04.2016)
- [15] URL: <http://www.dwheeler.com/essays/minisat-user-guide.html> (access date: 1.04.2016)