

DOI: 10.51790/2712-9942-2020-1-1-2

КОНЦЕПЦИЯ МНОГОУРОВНЕВОЙ ТРАССИРОВКИ ЦЕПЕЙ ИНТЕГРАЛЬНЫХ СХЕМ С ИСПОЛЬЗОВАНИЕМ ВИРТУАЛЬНЫХ КАНАЛОВ**С. Е. Власов¹, М. М. Годовицын^{2,a}, Н. В. Старостин^{2,b}**¹ *Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук», г. Москва, Российская Федерация, vlasov@niisi.ru*² *Нижегородский государственный университет, г. Нижний Новгород, Российская Федерация*
^a *maxim.godovitsyn@gmail.com*, ^b *nvstar@mail.ru*

Аннотация: рассматривается проблема трассировки цепей интегральной схемы. Предлагается многоуровневый метод, в основу которого положены идеи редукции сетки трассировки, а также расширения коммутационных ресурсов кристалла за счет введения дополнительных каналов. Редукция сетки трассировки обеспечивает существенное сокращение времени работы известных алгоритмов трассировки. Введение дополнительных каналов дает возможность свести задачи поиска трассы в сложной топологии коммутационного пространства к задаче вытеснения построенных трасс с дополнительных (виртуальных) каналов.

Ключевые слова: трассировки цепей, интегральная схема, многоуровневый метод, виртуальные каналы.

Для цитирования: Власов С. Е., Годовицын М. М., Старостин Н. В. Концепция многоуровневой трассировки цепей интегральных схем с использованием виртуальных каналов. *Успехи кибернетики*. 2020;1(1):8–15. DOI: 10.51790/2712-9942-2020-1-1-2.

THE CONCEPT OF MULTISTAGE INTEGRATED CHIP ROUTING WITH VIRTUAL CHANNELS**Sergey E. Vlasov¹, Maksim M. Godovitsyn^{2,a}, Nikolay V. Starostin^{2,b}**¹ *Federal State Institution “Scientific Research Institute for System Analysis of the Russian Academy of Sciences”, Moscow, Russian Federation, vlasov@niisi.ru*² *Lobachevsky State University of Nizhni Novgorod, Nizhniy Novgorod, Russian Federation*
^a *maxim.godovitsyn@gmail.com*, ^b *nvstar@mail.ru*

Abstract: the study solves the problem of circuit routing in ICs. We propose a multistage approach based on interconnection mesh reduction. It expands the chip interconnecting capability by introducing extra channels. The interconnection mesh reduction significantly accelerates the existing routing algorithms. The introduction of more channels makes it possible to reduce the routing problem in a complex interconnect space topology by removing the new routes from the extra (virtual) channels.

Keywords: circuit routing, integrated circuit, multistage method, virtual channels.

Cite this article: Vlasov S. E., Godovitsyn M. M., Starostin N. V. The Concept of Multistage Integrated Chip Routing with Virtual Channels. *Russian Journal of Cybernetics*. 2020;1(1):8–15. DOI: 10.51790/2712-9942-2020-1-1-2.

Введение

Этап трассировки цепей входит в типовой маршрут проектирования интегральной схемы на стандартных ячейках и заключается в проектировании топологии проводящих соединений, связывающих контактные площадки компонент схемы. Каждая цепь должна реализовываться своей связной системой проводников, которую принято называть трассой. В допустимой топологии проводящих соединений никакие проводники, принадлежащие разным трассам, не должны контактировать друг с другом.

Главные проблемы трассировки обусловлены размерностью, сложностью схем, а также миниатюризацией геометрических размеров элементов микросхем, что в результате приводит к увеличению

числа объектов и плотности их упаковки в пространстве кристалла. С учетом NP-полноты [1] проблемы трассировки и ориентируясь на характерную большую размерность практических задач, требования к прикладным алгоритмам трассировки постоянно возрастают. Ситуация усугубляется тем, что для субмикронных топологических норм требуется учитывать взаимное влияние соседних трасс друг на друга — в этих обстоятельствах, невзирая на существующее многообразие методов и средств решения рассматриваемого класса задач, построение быстрого и качественного трассировщика является актуальной проблемой.

В данной работе представлен многоуровневый метод трассировки, который, используя не одну, а множество сеток трассировки, обеспечивает необходимый баланс между скоростью поиска решения и его качеством.

Задача трассировки

Исходными данными для задачи трассировки являются: слои, горизонтальные и вертикальные треки, образующие сетку трассировки; зоны запрета металлизации на сетке трассировки, включая металл трасс заземления и питания; закрепленные за узлами сетки трассировки контактные площадки компонентов схемы; перечень цепей схемы, с указанием для каждой цепи набора контактов; реализованные металлом трассы некоторого подмножества цепей. Под трассой цепи понимается связанная система металлических проводников, связывающая все контакты цепи, проходящая исключительно в рамках заданной структуры сетки трассировки и не имеющая контактов с зонами запрета и трассами иных цепей. Главная задача трассировки состоит в реализации трасс всех цепей.

Точные и приближенные алгоритмы трассировки представляют главным образом академический интерес и в реальных условиях для решения практических задач не используются из-за экспоненциального роста вычислительных издержек при росте размерности задачи.

Основные практически значимые эвристические алгоритмы трассировки можно разделить на два класса. Первый класс составляют алгоритмы, осуществляющие поиск решения непосредственно на сетке трассировки. К таким алгоритмам относятся канальные и магистральные [2, 3], волновые [4, 5] и др. Главная проблема данных алгоритмов – преждевременная фиксация трасс, что приводит к быстрому исчерпанию ресурсов коммутационного поля. В результате нередко ситуации, когда неудачно проведенная трасса перекрывает возможность реализации других цепей.

Ко второму классу относятся алгоритмы [6, 7], которые выполняют расчет системы трасс в два этапа: вначале осуществляют укладку трасс на непрерывной плоскости (дает возможность между любой парой непересекающихся трасс всегда провести третью трассу), а затем решают задачу переноса решения из непрерывного пространства в дискретное, которое определяется сеткой трассировки. Фактически первый этап позволяет задать, в некотором смысле, план реализации цепей, но при этом базовая проблема преждевременного исчерпания ресурсов коммутационного поля на втором этапе все также имеет место.

Оба класса описанных алгоритмов нередко на практике сталкиваются с проблемой получения недостроенных трасс, которая решается многократной перетрассировкой системы трасс во всей или в отдельно взятой области в автоматизированном режиме (с участием эксперта). Подобные решения характеризуются большими издержками, приводящими в итоге к удорожанию всего процесса конструкторского проектирования изделий микроэлектроники.

Концепция быстрой многоуровневой трассировки

Для решения проблемы ускорения счета на практике нормой стало использование приемов декомпозиции трассируемого пространства на крупные фрагменты (блоки, области) и транзитные участки, связывающие фрагменты сеткой из макродискретов. На этапе глобальной трассировки реализуются только транзитные цепи, связывающие элементы из разных блоков. На этапе детальной трассировки реализуются локальные цепи в рамках одного фрагмента. Главной проблемой подобного двухуровневого подхода [8] является последовательная фиксация транзитных трасс, что резко ограничивает возможности пространства для трассировки следующих трасс.

Предлагается использование известной многоуровневой концепции [9] для решения задач трассировки в предположении, что данная концепция в значительной степени будет свободна от подобных негативных явлений за счет манипуляций с сеткой трассировки, включая ее редукцию и расширение.

В основе многоуровневой схемы лежит построение для всей сетки трассировки её грубой версии (редукция сетки трассировки). Она получается объединением соседних треков (вертикальные или горизонтальные линейки металлизации в рамках всего кристалла) в группы, с одинаковым, по возможности, числом треков в группах. Такие группы фактически представляют собой каналы в исходной сетке трассировки. В результате получается новая огрубленная сетка трассировки, где каждый узел и каждое ребро соответствуют определенному фрагменту исходной сетки трассировки. Припишем каждому ребру сетки число, характеризующее потенциальное число трасс, которое можно реализовать в рамках соответствующего фрагмента. Это число будем называть трассируемая способность ребра, и оно определяется как произведение числа треков в группе и числа слоев металлизации.

По аналогии на базе первой редукции сетки трассировки можно построить вторую редукцию и так далее до получения необходимого числа редукций. Результатом первого этапа будет построенная система редукций сетки трассировки, которые описываются графами различных размеров и характеризуются разной степенью детализации.

Для расчета одной трассы предлагается использовать многоуровневый волновой алгоритм, который вначале выполняет расчет трасс на грубой сетке без учета топологических пересечений проводников разных цепей (псевдотрассировка), но с учетом трассируемых способностей ребер. В этом случае полученная трасса, в терминах грубой сетки трассировки, есть связный фрагмент исходной сетки трассировки, в рамках которого волновой алгоритм реализует трассу.

Вполне очевидна ситуация, когда алгоритм на некотором шаге не сможет найти связную трассу в рамках выделенного фрагмента. В этом случае фрагмент расширяется (в предельном случае может включить все элементы данной сетки) и решение формируется уже за пределами фрагмента сетки, описываемого трассой, найденного на более грубой версии сетки трассировки.

В результате найденное решение на исходной сетке трассировки фиксируется и корректируются соответствующие трассируемые способности ребер грубых версий сетки трассировки.

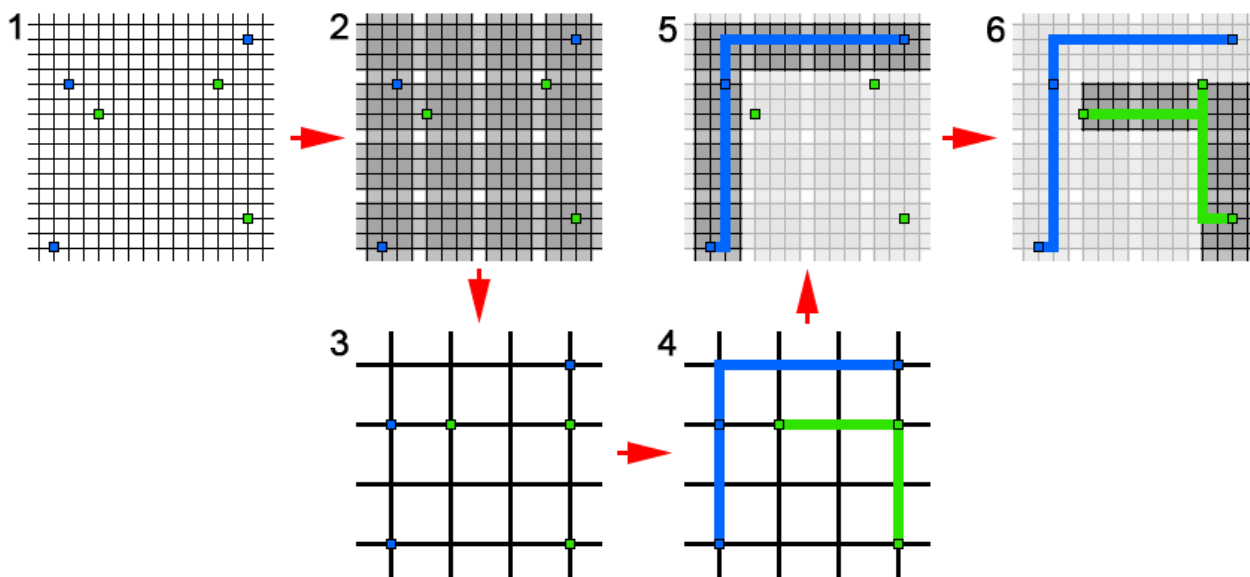


Рис. 1. Этапы многоуровневой трассировки

На рисунке (см. рис. 1) схематично представлены основные шаги многоуровневой трассировки на примере двух уровней. Исходные данные (этап 1) включают сетку трассировки, перечень цепей для каждой цепи, перечень контактов и их расположение в терминах треков сетки трассировки. Далее (этап 2) треки исходной сетки отождествляются в группы (каналы) и подготавливаются редукции сетки трассировки. На рисунке представлена одна редукция (этап 3), на практике их может быть несколько, и их число — параметр алгоритма.

Трассировка цепей осуществляется на редуцированной сетке (этап 4) — фактически на этом этапе подготавливается схема построения общего решения, которая выражается в системе трасс на редуцированной сетке. При этом каждая такая трасса описывает систему фрагментов исходной сетки,

в рамках которой с высокой вероятностью будет реализована цепь. На этом этапе трассы могут пересекаться, а их бесконфликтная реализация предполагается на этапе переноса редуцированного решения на исходную сетку (этап 5).

Общее решение последовательно переносится на исходную сетку трассировки (этапы 5 и 6), далее корректируется и оптимизируется с учетом требований, обусловленных топологией трассы, а также влиянием соседствующих токопроводящих соединений.

Концепция качественной трассировки на основе виртуальных каналов

Описанная выше многоуровневая схема трассировки позволяет ускорить процесс расчета трасс, в ее основу заложен механизм равномерного использования коммуникационных ресурсов интегральной схемы. Однако, как уже отмечалось выше, данная многоуровневая схема вполне допускает ситуацию, при которой волновой алгоритм в рамках выделенного фрагмента на исходной сетке трассировке не сможет построить искомую трассу. Сфокусируемся на данной проблеме.

Анализ различных вариантов решения проблемы недостроенных трасс позволяет сформулировать следующие тезисы. Во-первых, перетрассировка на исходной сетке, как правило, малоэффективна. Во-вторых, использование топологических методов перетрассировки в метрическом пространстве сопряжено с проблемой отображения непрерывного решения в дискретное пространство. Однако реализация цепей в непрерывном пространстве ограничивается только топологией графа (гиперграфа) цепей. Если решение существует, то существует эффективный алгоритм отыскания решения задачи. Это обеспечивается свойством непрерывного пространства, которое дает возможность между любой парой непересекающихся трасс всегда провести третью. Воспользуемся этим аспектом для решения задачи перетрассировки.

Для расширения возможностей коммутационного поля предлагается ввести в имеющуюся сетку трассировки новые треки — будем их называть виртуальными каналами. Для чего между каждой парой соседних имеющихся треков вставим дополнительный виртуальный канал, который фактически, с некоторой степенью условности, моделирует особенности непрерывного пространства.

В результате технология перетрассировки выглядит следующим образом: на первом этапе коммутационное поле с частично построенным решением дополняется виртуальными каналами; на втором этапе выбирается недостроенная цепь и находится ее трасса; на третьем этапе вытесняются фрагменты трасс, попавшие на виртуальные каналы. Второй и третий этапы выполняются до тех пор, пока не будут перебраны все проблемные цепи. Рассмотрим подробно все этапы представленной технологии.

Дополнение сетки трассировки виртуальными каналами

Для временного расширения возможностей коммутационного поля введем между каждой парой соседних имеющихся треков дополнительный (виртуальный) канал. На рисунке (см. рис. 2) представлена сетка трассировки до (а) и после (б) введения виртуальных каналов.

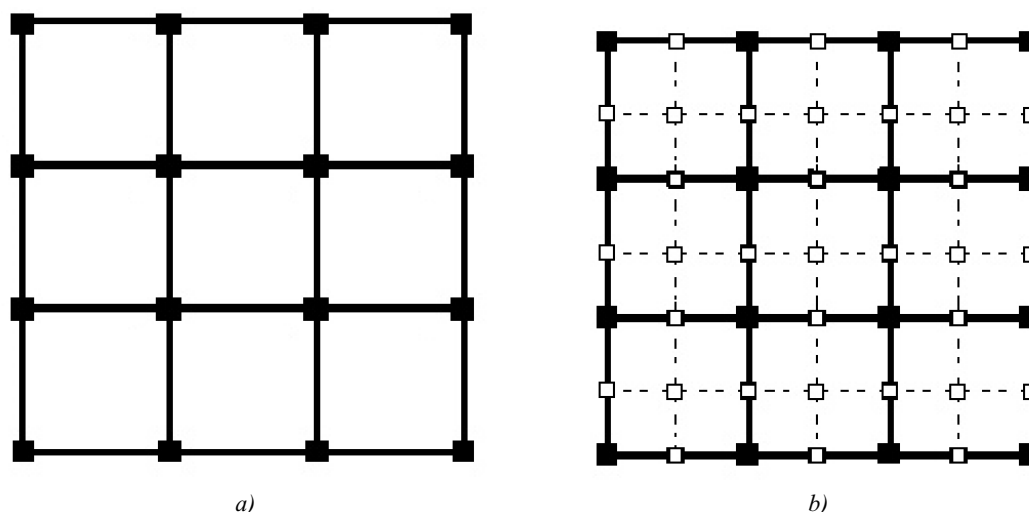


Рис. 2. Дополнение сетки трассировки виртуальными каналами

В результате получим новую сетку трассировки (см. рис. 2), в которой помимо реальных узлов (закрашенные) есть виртуальные (незакрашенные) — они опираются на вертикальные и/или горизонтальные виртуальные каналы. Ребра сетки трассировки, которые идентичны виртуальным узлам, опирающимся одновременно на два (вертикальный и горизонтальный) виртуальных канала, будем называть виртуальными (отмечены пунктиром).

Трассировка на сетке с виртуальными каналами

Трассировку одной цепи в данной сети предлагается осуществлять на базе волнового алгоритма Ли [5]. Этот алгоритм представляет собой метод поиска путей на сетке, основанный на алгоритме поиска в ширину (BFS). С учетом характера BFS алгоритм Ли гарантирует получение результата с минимальной длиной пути от исходной точки до целевой. Идея алгоритма Ли заключается в моделировании волны из исходных узлов в целевые, которые соединяются впоследствии общим проводником.

Трассировку многоконтактной цепи предлагается осуществлять последовательно. Выбирается некоторый начальный контакт — он фиксируется. Далее осуществляется волновая трассировка от всех неметаллизированных контактов до металла этой цепи — найденный фрагмент добавляется в трассу. Последний шаг продолжается до тех пор, пока все контакты цепи не будут соединены между собой, либо окажется, что их соединение невозможно.

Классический волновой алгоритм предлагается расширить модификацией [8], которая учитывает веса ребер сетки трассировки, которые могут интерпретироваться как условные расстояния между соседними узлами. Ограничим использование виртуальных каналов, для чего искусственно увеличим веса для виртуальных ребер сетки трассировки.

Вытеснение трасс с виртуальных каналов

После успешной трассировки на сетке с виртуальными каналами требуется вытеснить металл построенной трассы со всех виртуальных каналов без нарушения целостности всех реализованных трасс. В целях упрощения изложения материала будем рассматривать случай задачи трассировки с одним слоем металлизации.

Рассмотрим частный случай, когда требуется освободить от металла единственный узел сетки трассировки с сохранением целостности трассы, которой принадлежит металл. Будем рассматривать только варианты корректировки трассы, связанные с перемещением в заданном направлении металла из заданного узла в свободные соседние узлы сетки трассировки. Для определенности на данном этапе изложения будем рассматривать только 3 направления перемещения металла «вверх» (см. рис. 3).

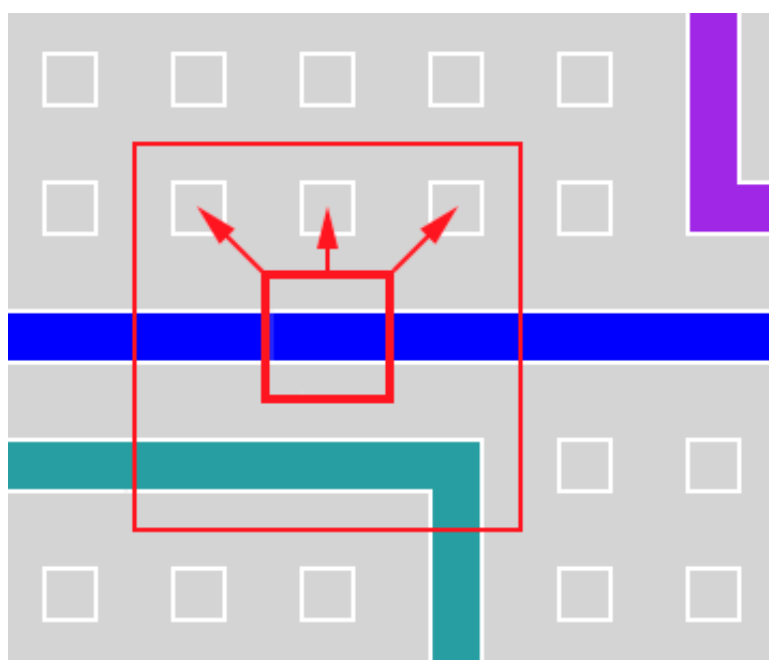


Рис. 3. Металлизированный узел сетки и направления перемещения металла

Предлагается решать рассматриваемую частную задачу алгоритмом УЗЕЛ.

Алгоритм УЗЕЛ

1. Построить новую конфигурацию трассы — из рассматриваемого узла перенести металл во все соседние узлы сетки трассировки в заданных направлениях.
2. Проверить целостность трассы. Если целостности трассы нет, то частная задача решения не имеет, восстановить исходную конфигурацию трассы и осуществить выход с результатом «РЕШЕНИЯ НЕТ». При положительном ответе переход к пункту 3.
3. Оптимизируем трассу методом отбрасывания лишнего металла — исключение данного металла не влияет на целостность трассы.
4. Проверяем на пересечение новой трассы с металлом других трасс. Если такие пересечения имеются, то восстанавливается исходная конфигурация трассы и осуществляется выход с результатом «РЕШЕНИЕ НАЙДЕНО» и с перечнем конфликтующих узлов сетки трассировки.
5. Осуществляется выход с результатом «РЕШЕНИЕ РЕАЛИЗОВАНО» и с перечнем металлизированных узлов сетки трассировки.

ЗАМЕЧАНИЕ. Алгоритм УЗЕЛ можно значительно оптимизировать по времени работы. Суть оптимизации в построении базы данных всевозможных вариантов конфигурации трасс (всего 256 вариантов) и готовых решений для каждого варианта. В этом случае алгоритм заключается в тривиальном извлечении готового решения по заданной конфигурации.

Рассмотрим более общий случай задачи, когда требуется освободить от металла единственный виртуальный канал сетки трассировки с сохранением целостности всех трасс. Для решения данной задачи воспользуемся алгоритмом УЗЕЛ решения частной задачи. Для упрощения изложения предположим, что требуется освободить горизонтальный трек. В этом случае перемещать металл нужно от виртуального канала — это означает, что если металлизированный узел сетки трассировки располагается выше виртуального канала, то металл нужно перемещать «вверх», в противном случае «вниз».

Алгоритм ТРЕК

1. Обозначим через *QUEUE* очередь узлов, требующих перемещения металла. Добавить в очередь *QUEUE* все металлизированные узлы, которые находятся в соседстве к металлу виртуального канала.
2. Проверить возможность переноса металла из виртуальных каналов, для чего:
 - а) Проверить возможность переноса металла в направлении «вверх»: для каждого металлизированного узла виртуального канала запускаем алгоритм УЗЕЛ. Если для каждого узла результатом является «РЕШЕНИЕ РЕАЛИЗОВАНО», то выход из алгоритма с результатом «РЕШЕНИЕ РЕАЛИЗОВАНО».
 - б) Проверить возможность переноса металла в направлении «вниз»: для каждого металлизированного узла виртуального канала запускаем алгоритм УЗЕЛ. Если для каждого узла результатом является «РЕШЕНИЕ РЕАЛИЗОВАНО», то выход из алгоритма с результатом «РЕШЕНИЕ РЕАЛИЗОВАНО».
3. Если очередь пуста, то выход из алгоритма с результатом «РЕШЕНИЕ НЕ НАЙДЕНО».
4. Извлечь из очереди *QUEUE* узел — если узел лишен металлизации (не исключается такая ситуация в процессе работы алгоритма), то переход к шагу 3. В противном случае переход к шагу 5.
5. Для узла выполнить алгоритм УЗЕЛ. Если возвращен результат «РЕШЕНИЯ НЕТ», то переход к шагу 3. Если возвращен результат «РЕШЕНИЕ НАЙДЕНО», то добавить в очередь *QUEUE* конфликтующие узлы и переход к шагу 3. Если возвращен результат «РЕШЕНИЕ РЕАЛИЗОВАНО», то переход к шагу 6.
6. Для узлов металлизации реализованного решения сформировать множество соседних узлов в направлении виртуального канала.
7. Добавить в очередь *QUEUE* множество соседних узлов и переход к пункту 2.

Конечность алгоритма ТРЕК очевидна — либо все металлизированные узлы виртуального канала окажутся освобождены (задача решена), либо очередь узлов рано или поздно опустеет (ввиду конечности сетки трассировки). Представленный алгоритм ТРЕК тривиально трансформируется для случая вертикального виртуального канала.

Самый общий случай задачи предполагает вытеснение металла с нескольких треков сетки трассировки с сохранением целостности всех трасс. Решения данной задачи предлагается построить последовательным вытеснением металла с виртуальных каналов.

Алгоритм ТРЕКИ

1. Свободные от металла виртуальные каналы исключаются из сетки трассировки.
2. Занятые металлом виртуальные просматриваются последовательно согласно некоторому порядку.
Для каждого виртуального канала:
 - а) Выполнить алгоритм ТРЕК, если на результат алгоритма «РЕШЕНИЕ НЕ НАЙДЕНО», то выход с результатом «РЕШЕНИЕ НЕ НАЙДЕНО».
 - б) Исключить виртуальный канал из сетки трассировки.
3. Выход с результатом «РЕШЕНИЕ РЕАЛИЗОВАНО».

Вычислительный эксперимент

Описанный выше алгоритм был реализован на языке C#. Создано тестовое приложение, главной задачей которого была апробация предложенной многоуровневой технологии решения задачи трассировки.

В таблице 1 представлены результаты работы многоуровневого алгоритма трассировки на искусственно сгенерированных тестовых задачах. Эксперименты проводились на ПК с конфигурацией Intel i7 2 ГГц / 16 Гб ОС MS Windows 7x64.

Таблица

Тестовый пример				Время работы (сек)	
Цепей	Контактов	Узлов в слое	Слоев	Волновой алгоритм	Многоуровневый алгоритм
10^4	5×10^4	10^6	3	1841	310
10^4	10^5	10^6	3	6099	855
10^4	5×10^4	10^7	3	2721	322
10^4	5×10^4	10^6	5	4155	582

Сравнение алгоритмов трассировки

Многоуровневый алгоритм осуществлял редукцию сетки трассировки один раз, при этом её размер уменьшался в 100 раз. Как результат, многоуровневый алгоритм в среднем более чем в 7 раз выполнил трассировку быстрее классического волнового алгоритма. Следует отметить, что качество полученных решений представленными алгоритмами различались незначительно.

Далее представлены скриншоты, демонстрирующие пример алгоритма трассировки на виртуальных каналах.

На рисунке (см. рис. 4) показан пример результата работы алгоритма трассировки на виртуальных каналах. На изображении (а) представлены исходные данные задачи. Заметим, что имеется нетрассированная цепь, для которой волновой алгоритм не способен найти решение. Изображение (б) демонстрирует результат работы алгоритма трассировки на виртуальных горизонтальных каналах. Изображение (с) демонстрирует результат работы алгоритма трассировки на виртуальных вертикальных каналах. Как результат, в обоих случаях была реализована не трассированная ранее цепь, при сохранении целостности изначально реализованных трасс.

Подводя итог, заметим, что экспериментальные исследования подтвердили предположения о том, что предложенные технологии многоуровневой трассировки имеют высокий потенциал к применению в практике решения проблемы синтеза интегральных схем с субмикронными топологическими нормами.

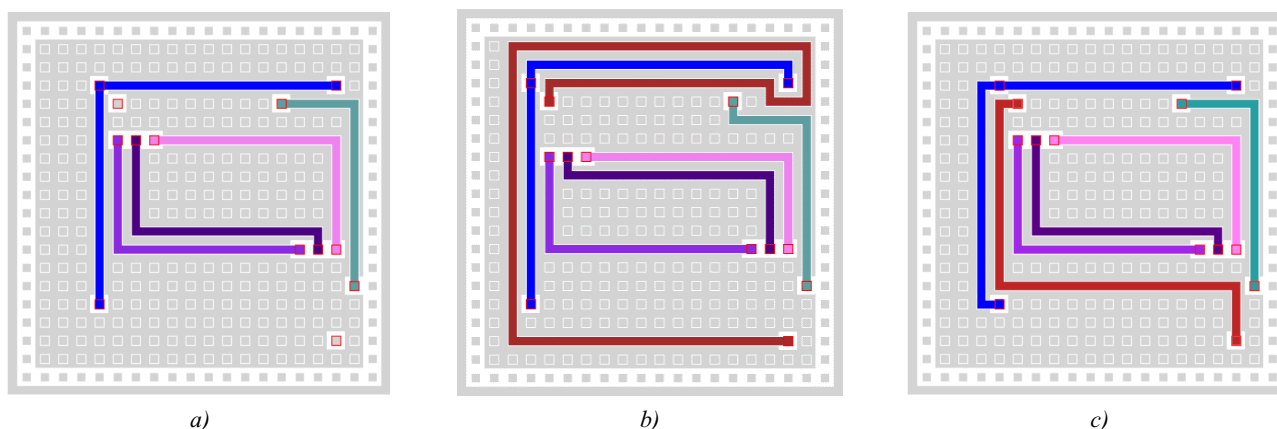


Рис. 4. Демонстрация результатов работы трассировки на основе виртуальных каналов

Заключение

В работе рассмотрена проблема повышения эффективности трассировки цепей интегральных схем. Предложена многоуровневая схема трассировки, обеспечивающая экономию временных вычислительных ресурсов. Предложена многоуровневая трассировка на основе виртуальных каналов, предназначенная для рационального использования ресурсов коммутационного поля. Планируется апробация и настройка представленных алгоритмов в контексте решения практических задач для интегральных схем с субмикронными топологическими нормами.

ЛИТЕРАТУРА

1. Гэри М., Джонсон Д. *Вычислительные машины и трудно решаемые задачи*. М.: Мир; 1982. 584 с.
2. Mikami K., Tabuchi K. A Computer Program for Optimal Routing of Printed Circuit Conductors. *Proc. IFIP Congress*. 1968;1475–1478.
3. Hightower D. W. A Solution to Line-Routing Problems on the Continuous Plane. *Proc. 6th Annual Design Automation Conference (DAC '69)*. 1969;1–24.
4. Hadlock F. O. A Shortest Path Algorithm for Grid Graphs. *Networks*. 1977;7(4):323–334.
5. Lee C. Y. An Algorithm for Path Connections and Its Applications. *IRE Transactions on Electronic Computers*. 1961;EC–10(3):346–365.
6. Hama T., Etoh H. Topological Routing Path Search Algorithm with Incremental Routability Test. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1999;Feb:142–150.
7. Старостин Н. В., Балашов В. В. Использование гиперграфов для решения задачи ортогональной трассировки больших интегральных схем с нерегулярной структурой. *Радиотехника и электроника*. 2008;53(5):618–623.
8. Батищев Д. И., Старостин Н. В., Филимонов А. В. Двухуровневая эволюционно-генетическая трассировка электрических цепей на графовой модели. *Проблемы разработки перспективных микро- и наноэлектронных систем (МЭС)*. 2008;1:61–64.
9. Батищев Д. И., Старостин Н. В., Филимонов А. В. Многоуровневый генетический алгоритм решения задачи декомпозиции гиперграфа. *Известия СПбГЭТУ ЛЭТИ*. 2007;1:3–13.