

Синтез блоков памяти с использованием представления правил в виде булевых функций от топологических объектов

Н.В. Рыженко, А.А. Сорокин, С.А. Быков

ЗАО «Интел А/О», nikolai.v.ryzhenko@intel.com

Аннотация — В данной работе представлены аспекты физического проектирования блоков памяти для перспективных нанометровых технологий. Предложена система универсального представления технологических правил в виде булевых функций. Предложен алгоритм построения полностью безошибочной детальной трассировки сбалансированных деревьев с минимизацией сегментации проводников. Экспериментальные результаты показали применимость предложенных подходов для синтеза промышленных блоков управления памяти.

Ключевые слова — трассировка, стандартные ячейки, память, булева выполнимость, оптическая литография.

I. ВВЕДЕНИЕ

Отладка каждого нового технологического производства полупроводниковых интегральных схем традиционно производится на регулярных массивах памяти. При физическом проектировании памяти на данном этапе можно отметить несколько отличительных особенностей. Во-первых, в процессе проектирования технологические правила непрерывно претерпевают изменения и уточнения. Во-вторых, велика доля ручного проектирования, поскольку коммерческие инструменты автоматического синтеза на стандартных ячейках не поддерживают новые технологические правила.

На рис. 1 представлена примерная структура организации физической памяти. Блоки ячеек памяти перемежаются вертикальными и горизонтальными полосами. Каждая полоса составлена из идентичных логических устройств (макроблоков), управляющими записью данных в память и чтением их из памяти. Каждый макроблок состоит из нескольких стандартных ячеек. Число типов стандартных ячеек, входящих в макроблок, невелико. Синтез и корректировка ячеек в условиях изменяющихся правил является отдельной задачей, и решение её выходит за рамки данной работы.

Размещение такого небольшого количества стандартных ячеек не составляет больших трудностей. Зачастую это операция выполняется вручную или несложными программами на основе существующего размещения аналогичных логических блоков, спроектированных для предыдущих технологических процессов. Размещение ячеек в блоке не меняется

вообще или не меняется значительно при корректировке технологических правил.

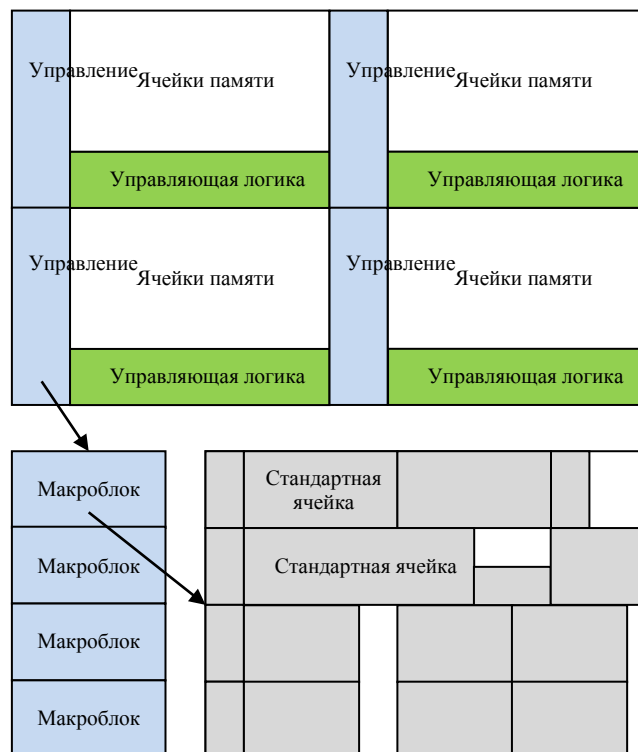


Рис. 1. Примерная структура памяти

Детальная трассировка – более трудоемкий процесс, сложность которого возрастает с переходом на всё более передовые технологии производства интегральных схем. Размеры топологических объектов современных интегральных полупроводниковых технологий в разы меньше длины волны лазера 193 нм. Как результат, от процесса к процессу растёт сложность технологических правил, а именно растёт число объектов и числовых значений, вовлечённых одновременно в одно правило. Для изготовления современных интегральных схем на технологиях 22 нм и меньше используется технология последовательного экспонирования двух и более масок [1]. Технологические правила для объектов, принадлежащих одной маске, и для объектов, принадлежащих разным маскам, отличаются друг от друга. Также данная технология рождает новый класс задач – распределение топологических объектов по маскам [2].

В данных условиях использование ручного режима для трассировки даже относительно небольших блоков стандартных ячеек становится нецелесообразным. Блоки приходится заново трассировать как при изменении технологических правил, так и при неоднократном уточнении архитектуры: решеток земли и питания, ширин проводников, количества и местоположение треков, размеров межслойных переходов. Число и сложность правил не позволяет выполнять эту работу в ручном режиме качественно и в разумные сроки.

Таким образом, становится актуальной задача автоматической трассировки блоков, состоящих из нескольких стандартных ячеек, с возможностью быстрой адаптации к изменяющимся технологическим правилам. При этом трассировщик:

- должен быть в высокой степени абстрагирован от всех типов правил;
- должен находить приемлемое и полностью легальное решение, исключаящее последующую ручную доводку.

Известные промышленные алгоритмы сводятся к традиционным последовательным итерационным подходам в различных вариациях канальной и волновой трассировки [3]-[5]. Последовательные алгоритмы многократно трассируют одни и те же соединения. Инкрементальная трассировка используется как для обеспечения полной связности электрической схемы, так и для достижения заданных электрических параметров. Переход к нанометровым технологиям производства интегральных схем делает использование последовательных подходов нецелесообразным. Поочерёдное построение и перестроение цепей не гарантирует сходимости и не позволяет эффективно учитывать всё многообразие современных правил, включающих сразу до нескольких объектов, лежащих на смежных слоях.

Для решения поставленных задач предлагается использовать систему представления технологических правил в виде булевых функций от топологических объектов [6] и точный алгоритм трассировки на основе задачи булевой выполнимости SAT [7].

Модель данных [6] позволяет полностью абстрагировать технологические правила от форм объектов и различных расстояний.

В работе [7] задача SAT для трассировки стандартных ячеек ставится совместно для всех электрических соединений. Это фундаментальное отличие от последовательных алгоритмов: определив допустимое решение SAT, мы автоматически определяем полную трассировку; перетрассировка отдельных сегментов цепей не требуется по определению. Также SAT позволяет задавать с помощью булевых выражений произвольные правила и ограничения между проводниками и межслойными переходами. Третье, и немаловажное, преимущество заключается в том, что SAT относится к классу точных

алгоритмов. Он или находит некое решение задачи, или же определяет с абсолютной точностью, что допустимого решения не существует. Область применимости [7] – трассировка транзисторов на уровне стандартной ячейки, что по числу терминалов совпадает по размерности с задачей двухслойной трассировки макроблоков, состоящих из нескольких десятков стандартных ячеек.

Содержание работы следующее. В главе 2 описано представление технологических правил в виде булевых функций. В главе 3 сформулирована задача булевой выполнимости в применении к задаче трассировки макроблоков. В главе 4 описаны проблемы, возникающие при данном подходе и способы их решения. Результаты представлены в главе 5.

II. МОДЕЛЬ ПРЕДСТАВЛЕНИЯ ПРАВИЛ

В работе [6] представлена модель данных, которая описывает произвольные геометрические правила с использованием булевых выражений для заданных дискретных прямоугольных шаблонов топологии. В данной модели для всех слоёв трассировки строится общая двумерная сетка. Для каждого узла сетки известен конечный список шаблонов топологии, которые могут быть построены в данном узле. Для каждого шаблона вводится соответствующая булева переменная. Шаблон в узле имеет два состояния: «присутствует» (соответствующая переменная принимает значение булевой единицы `1`) или «отсутствует» (переменная принимает значение булевого нуля `0`). Для переменных вводятся булевы выражения, применение которых описывается ниже.

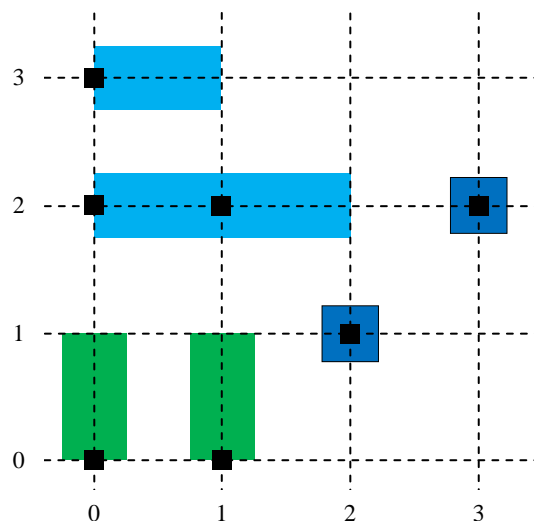


Рис. 2. Сетка и дискретные шаблоны топологии: отрезки проводников и межслойные переходы

На рис. 2 показана регулярная сетка трассировки для горизонтального и вертикального слоёв металлизации и соответствующего межслойного перехода. Отрезки вертикального слоя металлизации Mv присутствуют в узлах сетки (0,0) и (1,0) и отсутствуют во всех остальных узлах сетки. Отрезки горизонтального слоя металлизации Mh присутствуют

в узлах (0,2), (1,2), (0,3) и отсутствуют в остальных узлах сетки. Межслойные переходы *Via* присутствуют в узлах сетки (2,1) и (3,2).

Мы предлагаем использовать данные булевы выражения для описания нелегальных топологий. Предположим, что одновременное присутствие межслойных переходов в узлах сетки (2,1) и (3,2) на рис. 2 запрещено. Тогда допустима только такая комбинация значений переменных, при которой выражение (1) принимает значение булевого нуля `0`.

$$Via(2,1) \wedge Via(3,2) \quad (1)$$

Практика показывает, что любое геометрическое правило построения топологии можно перевести в соответствующее булево выражение, описав все возможные запрещённые ситуации данного правила. Ниже приведены примеры нескольких базовых правил.

Так, формула $\overline{Mv(1,0)} \wedge Mv(1,1) \wedge \overline{Mv(1,2)}$ запрещает появление отдельно стоящего вертикального проводника в узле сетки (1,1) (рис. 3). Данное выражение принимает запрещённое значение `1` тогда, когда дискретный отрезок вертикального проводника отсутствует в узлах (1,0) и (1,2), но присутствует в узле (1,1). При этом физически образуется слишком короткий провод, что запрещено.

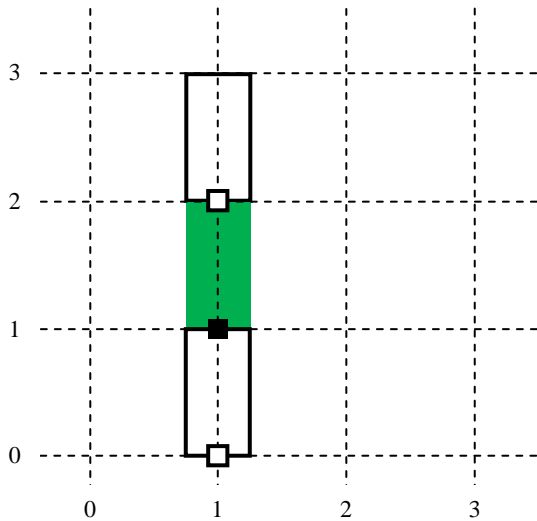


Рис. 3. Данная топология нарушает правило: Минимальная длина вертикального проводника

Аналогично формулируется выражение для минимального расстояния между концами двух проводников в одном трассировочном треке (рис. 4). Выражение $Mv(1,0) \wedge Mv(1,1) \wedge Mv(1,2)$ принимает запрещённое значение `1` тогда, когда дискретный отрезок вертикального проводника присутствует в узлах (1,0) и (1,2), но отсутствует в узле (1,1). При этом физически образуется слишком короткое расстояние между концами проводов, что запрещено.

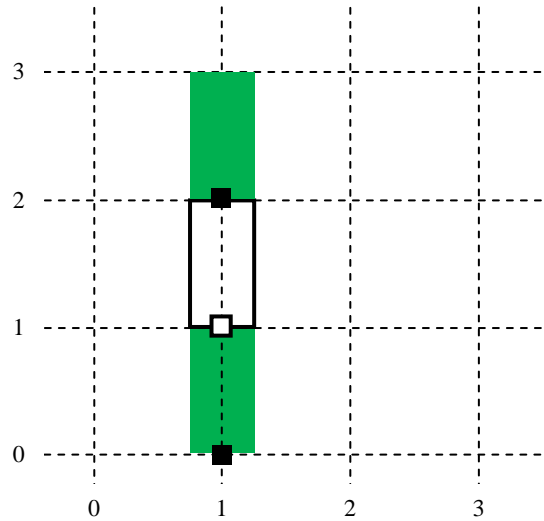


Рис. 4. Данная топология нарушает правило: Минимальное расстояние между концами двух проводников в одном треке

Минимальное перекрытие двух противоположно направленных проводников в соседних треках представляется аналогичным способом: $\overline{Mv(1,0)} \wedge Mv(1,1) \wedge Mv(2,1) \wedge \overline{Mv(2,2)}$ (рис. 5). Неважно, есть ли отрезки проводника в узлах (2,0) и (1,2), взаимное перекрытие двух проводников слишком мало и поэтому нелегально.

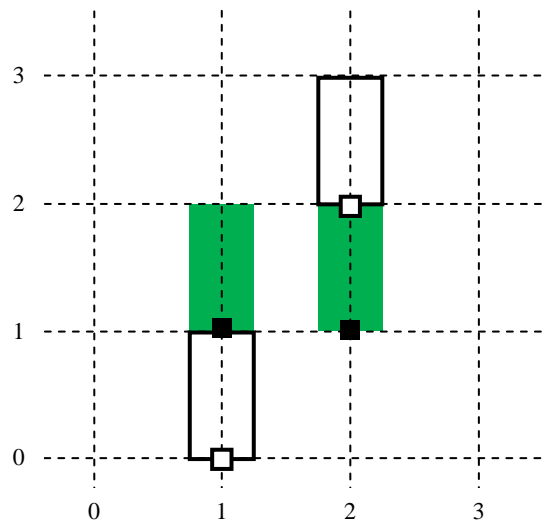


Рис. 5. Данная топология нарушает правило: Минимальное перекрытие противоположно направленных проводников в соседних треках

Отметим, что эффективным дополнением к функциям дизъюнкции, конъюнкции и отрицания является функция типа $GTE(N, \{x\})$, где N – неотрицательное число, $\{x\}$ – конечное множество булевых переменных, и функция принимает значение булевой единицы тогда, когда число булевых единиц среди переменных больше либо равно N .

III. ЗАДАЧА БУЛЕВОЙ ВЫПОЛНИМОСТИ В ПРИЛОЖЕНИИ К ДЕТАЛЬНОЙ ТРАССИРОВКЕ

В работе [7] предложен способ трассировки с использованием задачи булевой выполнимости. На предварительном этапе все электрические цепи разбиваются на двухточечные соединения. Для каждого соединения создаётся список из нескольких возможных топологий. Затем специальная процедура определяет конфликты между каждой парой возможных топологий. Связность трассировки, возможные реализации соединений, парные конфликты между ними и другие дополнительные ограничения переводятся в общую булеву формулу, представленную в конъюнктивной нормальной форме. Далее запускается программа поиска решения (*SAT solver*), ищется допустимое решение. Поскольку SAT относится к классу точных методов, то, определив допустимое решение SAT для общей формулы, автоматически определяется полная трассировка, при этом выполняются все заданные ограничения.

Мы предлагаем расширить данную постановку задачи. Для каждого дискретного объекта топологии (отрезка провода или межслойного перехода) и для каждой электрической цепи вводится дополнительный булевый литерал. Когда булевый литерал принимает значение булевой единицы, это значит, что этот объект топологии присутствует в трассировке, и он принадлежит конкретной электрической цепи. За счёт введения дополнительных булевых выражений, описывающих нелегальные топологии, как это описано в главе 2, ни одно технологическое правило не может быть нарушено в процессе трассировки.

На рис. 6 показан пример задачи трассировки в однородном слое S и три типа запрещённой топологии.

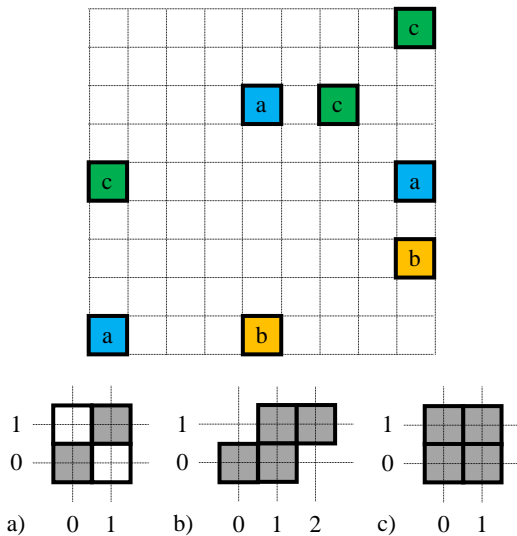


Рис. 6. Терминалы цепей и запрещённые топологии

Выражение (2) определяет запрещённую топологию, в которой два проводника касаются друг друга в одной точке (рис. 6b). Выражения (3) и (4) определяют нелегальные топологии с рис. 6b и 6c.

$$S(0,0) \wedge S(1,1) \wedge \overline{S(0,1)} \wedge \overline{S(1,0)} \quad (2)$$

$$S(0,0) \wedge S(1,0) \wedge S(1,1) \wedge S(2,1) \quad (3)$$

$$S(0,0) \wedge S(1,0) \wedge S(0,1) \wedge S(1,1) \quad (4)$$

На рис. 7 представлен один из вариантов возможной трассировки для данной задачи.

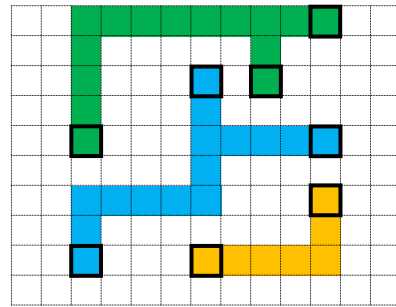


Рис. 7. Вариант трассировки

Предположим, появилось новое правило, которое дополнительно к старым правилам определяет минимальное допустимое расстояние между концом одного проводника и стороной другого проводника, идущего в перпендикулярном направлении (рис. 8).

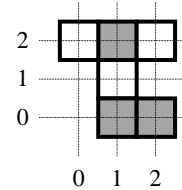


Рис. 8. Дополнительная запрещённая топология

Описав данную запрещённую топологию в виде выражения (5) и добавив его в соответствующий установочный файл, мы запускаем трассировку заново.

$$S(1,2) \wedge \overline{S(0,2)} \wedge \overline{S(2,2)} \wedge \overline{S(1,1)} \wedge S(1,0) \wedge S(2,0) \quad (5)$$

Если существует легальное решение для данной постановки задачи SAT, т.е. существует такая трассировка, которая соединяет все терминалы цепей, не создаёт электрические замыкания и удовлетворяет всем четырём описанным запрещённым комбинациям дискретных отрезков проводников, то такое решение будет непременно найдено (рис. 9). Это является принципиальной особенностью задачи булевой выполнимости формул.

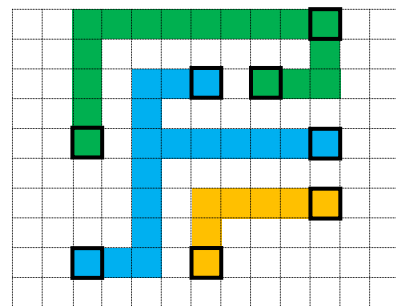


Рис. 9. Новый вариант трассировки

IV. ОПТИМИЗАЦИЯ ДЕТАЛЬНОЙ ТРАССИРОВКИ

Эволюция индустрии в направлении регулярной топологии [8] и прогресс в разработке программных комплексов для решения задач SAT [13] позволяют использовать SAT для трассировки макроблоков из нескольких десятков стандартных ячеек. При этом достигается высокая степень абстракции программы трассировки от технологических правил. Все возможные правила описываются в виде булевых выражений и хранятся в отдельном технологическом файле. Добавление, удаление и редакция правил не требуют какой-либо модификации программного кода. Технология SAT позволяет получить трассировку без единого нарушения. Однако особенность SAT программ такова, что они находят *первое возможное* допустимое решение. В данной главе представлены проблемы, возникающие при поиске оптимальной трассировки, и возможные пути их решения.

На рис. 10 изображены три ячейки А, В и С с терминалами на нижнем слое металлизации. Для построения трассировки необходимо реализовать 2 из 3-х возможных соединений АВ, АС и ВС, используя вышележащие горизонтальный и вертикальный слои металлизации.

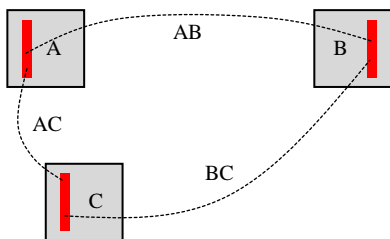


Рис. 10. Терминалы электрической цепи

Сразу отметим, что построение леса деревьев Штейнера в качестве глобальной трассировки малоэффективно из-за малой размерности задачи. Размер сетки глобальной трассировки, который обычно равен высоте одной-двух ячеек, по порядку сравним с размерами всего блока. Поэтому для каждого из соединений определяются все возможные реализации в некотором заданном ограничивающем прямоугольнике и с заданным числом сегментов проводников. На рис. 11 показан один такой вариант; характерное число всех Z-соединений для цепей подобных блоков – два порядка.

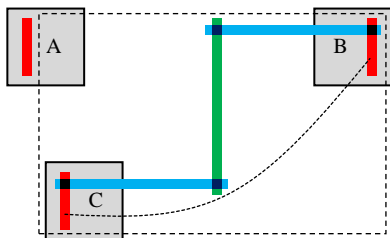


Рис. 11. Одно из возможных соединений

При решении задачи SAT будут выбран такой набор соединений, который позволяет построить полную трассировку цепи, удовлетворив всем

наложенным ограничениям. Для каждого соединения будет реализован полностью легальный, но абсолютно случайный вариант (рис. 12). Это является проблемой и требует наложения дополнительных ограничений.

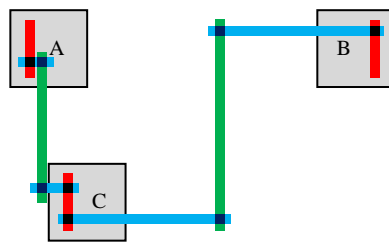


Рис. 12. Трассировка без ограничений

В общем случае терминал ячейки имеет несколько возможных точек пересечения с вышележащим слоем. Наложив строгие ограничения (6), где $\{v\}$ конечное множество межслойных переходов, примыкающих к некому терминалу, получим, что этого терминала будет касаться один и только один межслойный переход (рис. 13).

$$V_{i \neq j}(v_i \wedge v_j) = 0; \bigvee_{i=1}^n v_i = 1 \quad (6)$$

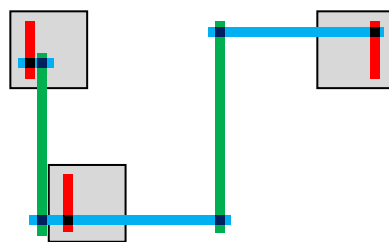


Рис. 13. Каждого терминала может касаться один и только один межслойный переход

В работе [7] показано, что, используя булевы функции определённого типа, можно устанавливать для конечного множества литералов произвольное число литералов, принимающих заданное значение. Это позволяет минимизировать число вертикальных трек для каждой отдельной электрической цепи. Для этого все соединения цепи разбиваются на группы. В каждой группе каждое соединение имеет отрезок проводника на определённом треке. Если соединение вообще не использует вертикальный слой металлизации, то оно никак не ограничивается в данной постановке.

Далее, используя встроенный механизм внешних ограничений [9], сначала накладывается ограничение $V = 0$, т.е. мы запрещаем появление любого соединения из любой группы. SAT находит, что данное ограничение недостижимо, поэтому оно отбрасывается и накладывается следующее $V = 1$, т.е. мы позволяем использовать любое количество соединений, но только из одной группы. Одновременно это запрещает появление соединений из любой другой группы. Для этого набора ограничений решение существует (рис. 14), т.е. будет построена такая трассировка из такого набора соединений, которые используют произвольный, но один и только один вертикальный трек.

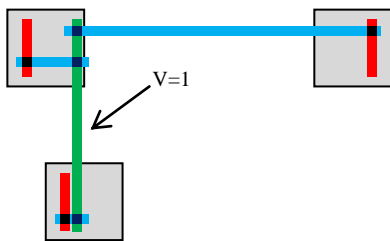


Рис. 14. Добавлено ограничение на число используемых вертикальных трексов $V = 1$

Аналогичный подход последовательных ограничений накладывается на число используемых горизонтальных трексов H . Последовательно накладывая ограничения $H = 0$, $H = 1$, $H = 2$, находим новое решение для $H = 2$ (рис. 15).

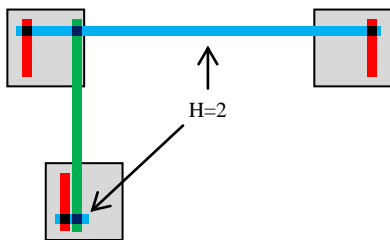


Рис. 15. Добавлено ограничение на число используемых горизонтальных трексов $H = 2$

Сбалансированность трассировки – важная характеристика, особенно для цепей синхронизации. На рис. 16 изображена электрическая цепь, состоящая из четырёх терминалов. Наложение вышеописанных ограничений не позволяет задать оптимальное положение вертикального проводника. На рис. 16 реализована комбинация АВ, AD и СВ. При данной трассировке задержка распространения сигнала от D до B больше, чем задержка $D \rightarrow A$ и $D \rightarrow C$.

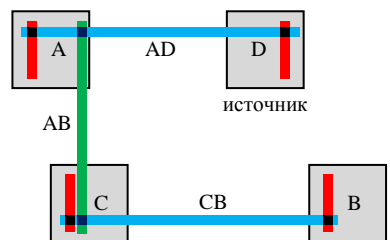


Рис. 16. Наложённые ограничения на число сегментов не определяют позицию вертикального проводника

В качестве четвёртого ограничения мы максимизируем число соединений между источником и потребителями сигнала. Вкупе с предыдущими тремя ограничениями: 1) на число подключений к терминалам, 2) число используемых вертикальных трексов и 3) число используемых горизонтальных трексов, получается трассировка, где вертикальный проводник локализован возле источника сигнала (рис. 17). Пунктиром показаны ограничивающие прямоугольники соединений DA, DB, DC, штриховкой показана область появления вертикального отрезка проводника.

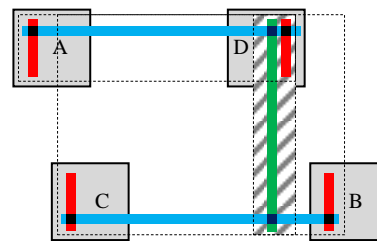


Рис. 17. Финальная трассировка

V. РЕЗУЛЬТАТЫ И ЗАКЛЮЧЕНИЕ

При отладке литографического процесса в первую очередь проектируется память, при этом характерными особенностями являются недоступность коммерческих программ физического проектирования, высокая доля ручного труда и корректировка правил. Детальная трассировка – наиболее трудоемкий процесс, сложность которого возрастает с переходом на всё более передовые технологии. Предложенный метод описания правил в виде булевых функций позволил полностью абстрагировать трассировщик от всех технологических правил произвольного типа. Это позволило редактировать, изменять и добавлять правила в сторонней программе с графическим интерфейсом без настройки и перекомпиляции самого трассировщика. Данное представление правил позволило использовать точный метод построения трассировки на основе задачи булевой выполнимости.

Трассировщик на основе программы [9] показал следующие типовые результаты. Финальная, полностью легальная, оптимизированная по числу сегментов и сбалансированная трассировка для блока из 25 стандартных ячеек, 19 цепей, 52 терминалов и 132 запрещённых конфигураций топологии была получена за ~9 часов. Было определено 13,026 возможных Z-соединений (рис. 10), создано 13,029,266 литералов; пиковое число неопределённых литералов – 3,789,968; максимальное число дизъюнкций литералов – 47,792,252. При использовании практиковались ночные запуски на нескольких машинах с разными вариантами правил, архитектуры блока и оптимизаций.

ЛИТЕРАТУРА

- [1] Kurt Ronse, et al. Lithography Options for the 32 nm Half Pitch Node and Beyond // IEEE Tran. on Circuits and Systems - I: Regular papers. August 2009. Vol. 56. № 8.
- [2] H. Tian, et al. Constrained pattern assignment for standard cell based triple patterning lithography // ICCAD. 2013.
- [3] Sanjay Rekh, J. Donald Trotter, Daniel H. Linder. Automatic layout synthesis of leaf cells // DAC. 1995.
- [4] Gupta A., J. P. Hayes. CLIP: Integer-Programming-Based Optimal Layout Synthesis of 2D CMOS Cells // ACM Tr. on DA El. Sys. July 2000. Vol. 5. № 3.
- [5] Mohan G. et al. CELLERITY: a fully automatic layout synthesis system for standard cell libraries // DAC. 1997.
- [6] Guszti Suto. Rule agnostic routing by using design fabrics // Proc. of DAC. 2012.
- [7] N. Ryzhenko, S. Burns. Standard cell routing via boolean satisfiability // Proc. of DAC. 2012.
- [8] Rasit O. Topaloglu. Design with FinFETs: design rules, patterns, and variability // Proc. of ICCAD. 2013.
- [9] URL: <http://minisat.se> (дата обращения: 07.04.2014).