УДК 004.023

Подсистема планировки цепей в стандартных ячейках нанометровых СБИС

В. А. Борисов[1], Д. Ю. Кривошеин[2], А. М. Марченко[1,2], Е. А. Попов[1], В. Ю. Савченко[1,3]

[1]ООО «Нангейт»

[2]Московский государственный университет имени М. В. Ломоносова

[3]Национальный исследовательский университет «МИЭТ», valentine.savchenko@gmail.com

***Аннотация* — В работе предложены подход к построению системы трассировки в стандартных ячейках и модификации методов решения подзадач, составляющих задачу глобальной трассировки (планировки цепей) стандартной ячейки: построение модели, общий алгоритм глобальной трассировки, эвристики для упорядочивания цепей, инкрементальный алгоритм пересчета кратчайших путей в графе глобальной трассировки.**

***Ключевые слова* — синтез топологии, трассировка, стандартные ячейки.**

I. Введение

Современные технологии производства нанометровых СБИС определяют систему сложных требований к топологии стандартной ячейки. С одной стороны, противоречивость этих требований сужает множество решений задачи трассировки цепей, с другой – делает решения плохо формализуемыми.

Практика использования библиотечного метода для проектирования нанометровых СБИС показывает тенденцию к увеличению размеров библиотек. Увеличение обусловлено прямой зависимостью между размером библиотеки и числом проектов, в которых библиотека может быть применена. Увеличение размеров библиотек снижает экономическую привлекательность полностью ручного проектирования библиотек и повышает актуальность разработки САПР, способных за приемлемое время синтезировать библиотеки больших размеров (от нескольких тысяч до нескольких десятков тысяч элементов). Время и корректность решения задачи трассировки являются критичными для таких САПР.

Задача трассировки стандартной ячейки на транзисторном уровне состоит в нахождении соединений между элементами множества полигонов, моделирующих истоки, затворы и стоки транзисторов, на заданном наборе слоёв в области ячейки.

II. Система синтеза топологии

На вход системы подаётся шаблон библиотеки, упорядоченный список транзисторов и список соединений ячейки (Рис. 1). На основе списка транзисторов рассчитывается оптимальная ширина ячейки, размещаются элементы шаблона и транзисторы.

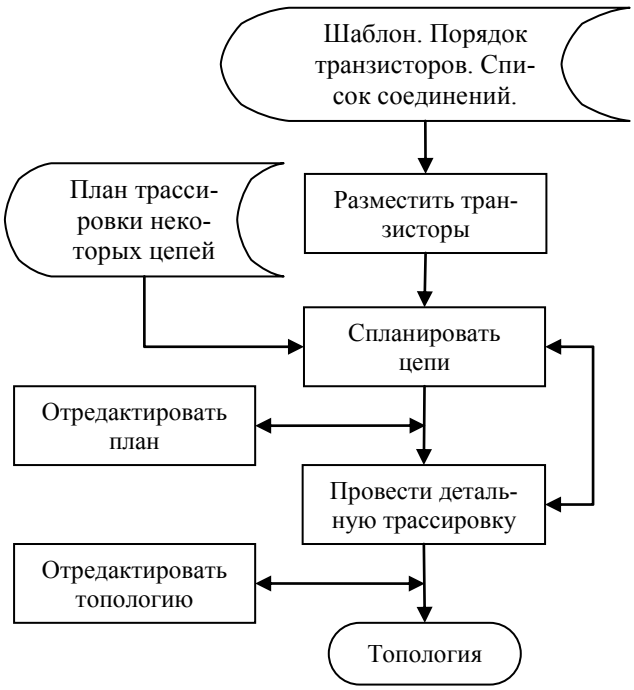

Рис. 1. Маршрут системы трассировки

Далее выполняется автоматическое планирование цепей. Однако на практике встречаются ячейки, для которых невозможно найти планировку за приемлемое время, поэтому пользователю предоставлен графический редактор, позволяющий задавать и вносить правки в план трассировки ячейки. Планы трассировки в этом редакторе представлены с помощью расширенной штрих диаграммы (stick diagram).

Затем производится отображение плана трассировки на топологию. В процессе отображения учитываются базовые технологические ограничения, что позволяет создавать топологию ячейки без нарушений большинства правил проектирования (DRC-aware routing).

Оставшиеся нарушения устраняются на завершающем этапе синтеза – сжатии топологии (проверке и устранении нарушений правил проектирования). Однако не всегда возможно автоматически разрешить все нарушения на этом этапе. Практика показывает, что для успешного сжатия части таких ячеек требуется внесение изменений в исходные топологии (полученные после отображения). Для этого в распоряжение пользователя предоставлен редактор топологии.

III. Планирование цепей

Традиционный подход к решению проблемы трассировки состоит в разделении задачи на несколько стадий [1]. Разделение производится для снижения размерности и упрощения оптимизации противоречивых критериев качества трассировки. Современные технологии производства нанометровых СБИС характерны большим числом и сложностью правил проектирования [5], что требует учёта правил проектирования на всех этапах синтеза топологии.

A. Граф планировки

В рамках системы планирование внутри ячейки производится на основе регионов. Разбиение ячейки на регионы начинается с создания сетки. Сетка образуется вертикальными и горизонтальными прямыми, проведёнными через углы полигонов транзисторов. Она определяет самое мелкое разбиение области трассировки на регионы. При этом некоторые из них могут иметь недостаточный размер для размещения в них проводов и/или межслойных переходов. Для преодоления этого ограничения сеточное разбиение огрубляется. Огрубление производится путём объединения смежных регионов слоя в соответствии с технологическими правилами. Построение сетки и её огрубление осуществляется для каждого трассировочного слоя отдельно.

С целью учёта дополнительных ограничений на глобальную трассировку, сохраняется информация об участках ячейки, над которыми лежит каждый регион. Данная информация важна, так как её анализ позволяет учитывать соединения, порождающие паразитные транзисторы; постановку контактов над затворами и другие технологические ограничения.

Регионы используются для создания графа планировки. Каждому региону ставится в соответствие вершина графа. Рёбра графа показывают смежность регионов, как на отдельном слое, так и на соседних слоях. По построению граф имеет решетчатую структуру, т. е. является разреженным. У каждого ребра имеется трассировочная ёмкость. Трассировочная ёмкость определяется геометрией пересечения регионов, моделируемых инцидентными с ребром вершинами [1]. Рёбра хранят информацию о распределении трассировочного ресурса между цепями, которые спланированы через них.

B. Маршрут

Планирование цепей на графе осуществляется путём последовательного поиска решения для очередной цепи (Рис. 2). Решение для каждой цепи моделируется деревом Штейнера. Планировка всех цепей ячейки – лесом деревьев Штейнера.

Пусть $G = (V, E)$ – взвешенный граф трассировки, где V– множество вершин графа, E – множество рёбер графа. $c: E \to \Re_+$ – положительная весовая функция, задающая вес каждого ребра графа. $W \subseteq V$ – множество вершин, моделирующих терминалы некоторой цепи. Задача построения плана некоторой цепи состоит в нахождении подграфа $G' = (V', E') \subseteq G : W \subseteq V', c(G') \to min$, где $c(G') = \sum_{e \in V'} c(e)$.

```
1:  Построить граф планировки
2:  Упорядочить цепи
3:  Рассчитать кратчайшие пути

4:  пока следует продолжать итерации
    {
5:    для каждой цепи
      {
6:      Построить дерево
7:      Обновить веса рёбер
8:      Обновить кратчайшие пути
      }

9:    Оценить лес
10:   если лес не удовлетворяет ограничениям
      {
11:     Обновить веса рёбер
12:     Обновить кратчайшие пути
      }
13:   иначе
      {
14:     Прекратить итерации
      }
    }
```

Рис. 2 Маршрут глобальной трассировки

Для построения деревьев Штейнера в маршруте используется модификация алгоритма MPH [6]. В силу того, что время построения леса зависит от времени построения дерева, для оптимизаций был выбран именно алгоритм MPH, обладающий небольшой вычислительной сложностью, которая была снижена до $O(|V|)$.

Предлагаемая модификация алгоритма построения дерева Штейнера требует хранения кратчайших путей между всеми парами вершин, поэтому в начальный момент эти пути рассчитываются. Так как граф является разреженным, то для расчёта используется алгоритм Джонсона, обладающий сложностью $O(|V|^2 lg|V|)$ [3].

В связи с необходимостью хранить кратчайшие пути между всеми парами вершин возникает дополнительная задача по поддержанию этой информации в актуальном состоянии, так как веса ребёр изменяются в процессе построения леса. Для решения этой задачи используется новый алгоритм, обновляющий кратчайшие пути только между теми вершинами, кратчайшие

пути между которыми до обновления содержали изменившиеся рёбра. Вычислительная сложность этого алгоритма для разреженных графов составляет $O(|V|^{4/3})$.

Таким образом, если обозначить через k – число цепей, I – число итераций для построения леса деревьев Штейнера, то оценка сложности маршрута составит $O\left(|V|^2 lg|V| + k \cdot I \cdot \left(|V| + |V|^{4/3}\right)\right)$.

На этапе обновления весов рёбер в цикле построения леса (строка 7 на Рис. 2) анализируется состояние трассировочного ресурса каждого ребра, входящего в план только что построенного дерева. Вес ребра увеличивается линейно, пока суммарный ресурс назначенных цепей не превышает ёмкости ребра, при назначении цепей сверх ёмкости рост становится экспоненциальным.

Оценка леса (строка 9 на Рис. 2) состоит в выявлении перегруженных рёбер (тех, на которые назначены цепи сверх трассировочной ёмкости) и в упорядочивании цепей на каждом ребре. Множество цепей, проходящих через вершину, определяются рёбрами, смежными с этой вершиной. Упорядочивание заключается в минимизации пересечений цепей внутри региона. Алгоритм упорядочивания цепей основан на идее топологической трассировки М. В. Щемелинина [7]. Устранение пересечений во всех регионах позволяет определить порядок детальной трассировки цепей в каждом регионе.

При обновлении весов рёбер (строка 11 на Рис. 2) назначаются дополнительные весовые штрафы на ребра, ассоциированные с регионами, в которых не удалось устранить пересечения.

C. Упорядочивание цепей

Практика показывает, что выбор порядка обработки цепей является ключевой подзадачей планирования. Нам не известны подходы к выбору порядка цепей, которые бы давали оптимальный результат во всех случаях, однако анализ библиотек единичной высоты, спроектированных ручным способом, показал эффективность подхода, согласно которому цепи, имеющие меньшее количество вариантов трассировки, следует обрабатывать в первую очередь [4]. Для того, чтобы провести ранжирование по этому признаку, предлагается следующая классификация цепей:

- P цепь – это цепь, соединяющая только стоки или истоки P транзисторов. К P цепям относится цепь питания;
- N цепь – это цепь, соединяющая только стоки или истоки N транзисторов. К N цепям относится цепь земли;
- PN цепь – это цепь, соединяющая только стоки или истоки транзисторов. PN цепи являются выходными цепями ячейки;
- G цепь – это цепь, соединяющая только затворы транзисторов. Цепи этого типа являются входными цепями ячейки;
- PNG цепями называются все остальные цепи ячейки.

Самыми простыми являются цепи земли и питания, поэтому они трассируются в первую очередь. Вторыми по простоте являются остальные цепи P и N типа.

Сравнимыми по количеству вариантов трассировки являются цепи PN и G типа. Цепи этих групп являются выходными и входными цепями ячейки, поэтому на их трассировку накладываются дополнительные ограничения, связанные с постановкой портов.

Самым широким набором конфигураций (вариантов трассировки) обладают цепи PNG типа. Основная трудность состоит в выборе правильного порядка для цепей этой группы. Для упрощения выбора множество PNG цепей разбивается на подмножества согласно их взаимным пересечениям. Если на некоторой итерации невозможно закончить планировку, то осуществляется независимое изменение порядков цепей внутри подмножеств.

D. Инкрементальный алгоритм поиска кратчайших путей в графе

Алгоритм основан на утверждении о том, что если имеется матрица кратчайших расстояний между всеми парами вершин графа, то при изменении веса одного ребра достаточно пересчитать кратчайшие пути между всеми парами вершин, кратчайшие пути между которыми содержали изменившееся ребро. Данное утверждение доказано в [2], там же показано, что вычислительная сложность алгоритма для разреженных графов составляет $O(|V|^{4/3})$.

```
1: Построить множества I и O

2: для каждой вершины i ∈ I
   {
3:   Рассчитать кратчайшие пути из i в O

4:   для каждой вершины o ∈ O
     {
5:     Обновить кратчайший путь
     }
   }
```

Рис. 3 Инкрементальный алгоритм поиска кратчайших путей в графе

Пусть в графе изменился вес только одного ребра (u, v), причем он увеличился, u – входящая вершина этого ребра, v – исходящая. Рассмотрим кратчайший путь P между вершинами u' и v', проходящий через ребро (u, v) и имеющий вид $u'P_1(u, v)P_2v'$, где P_1, P_2 – некоторые пути. Множества входящих вершин ребра $(u, v) - I$, исходящих – O. Процедура обновления кратчайших путей содержит следующие этапы.

Из вершины u поиском в ширину находятся все вершины, принадлежащие множеству I. Аналогично, из вершины v находятся все вершины, принадлежащие множеству O.

Для каждой вершины из множества I с помощью алгоритма Дейкстры вычисляются значения кратчайших путей до вершин множества O. Каждой вершине из множества O приписывается значение длины кратчайшего пути до каждой вершины из множества I, полученное на предыдущем шаге.

E. Алгоритм построения дерева Штейнера

На каждом шаге алгоритма МРН к уже построенной части дерева добавляется ближайшая, из ещё не присоединённых терминальных вершин цепи, вершина и кратчайший путь, ведущий из части дерева в эту вершину.

Если обозначить через $PATH(F, v)$ путь минимальной стоимости среди всех кратчайших путей из вершин множества $F \subseteq V$ в вершину $v \notin F$ и через $\hat{c}(F, v)$ стоимость $PATH(F, v)$, тогда формальное описание алгоритма нахождения дерева G'_k, имеющего k терминальных вершин [6], выглядит следующим образом:

- добавить в дерево произвольно выбранную вершину $w_1 \in W$, при этом $G'_1 = (V'_1, E'_1)$, $V'_1 = \{w_1\}$ и $E'_1 = \emptyset$;
- для каждого $i \in 2, 3, \dots, k$ найти вершину $w_i \in W \backslash V_{i-1}$, такую, что $\hat{c}(V_{i-1}, w_i) = min\left\{\left\{\hat{c}(V_{i-1}, w_j) \middle| w_j \in W \backslash V_{i-1}\right\}\right\}$. Построить дерево $G'_i = (V'_i, E'_i)$ путём добавления пути $PATH(V'_{i-1}, w_i)$ к дереву G'_{i-1}, т. е. $V'_i = V'_{i-1} \cup \{w | w \in PATH(V'_{i-1}, w_i)\}$ и $E'_i = E'_{i-1} \cup \{e | e \in PATH(V'_{i-1}, w_i)\}$.

В силу того, что в системе хранится актуальная информация о кратчайших путях между всеми парами вершин, расчёт пути $PATH(V'_{i-1}, w_i)$ может быть осуществлён за $O(|V|)$, а построение дерева потребует $O(k|V|)$ операций просмотра информации о кратчайших путях.

IV. Экспериментальные результаты

Предложенная подсистема планирования цепей используется в рамках системы синтеза библиотек Nangate Library Creator. Для демонстрации характеристик работы подсистемы была подготовленная тестовая библиотека, содержащая несколько сотен ячеек следующих групп:

- Группа 1 (3% от общего числа ячеек) – простейшие ячейки типа INV и BUF;
- Группа 2 (92%)– комбинационные ячейки типа NAND, NOR, AND, OR, AOI, OAI, AO, OA и MUX;
- Группа 3 (5%) – ячейки типа XOR, XNOR, HA и FA.

Каждая группа содержала варианты ячеек, рассчитанных на несколько входных сигналов (например, NAND2, NAND3, NAND4), и нескольких нагрузочных способностей (например, D1, D2, D4 и т. д.).

Шаблон библиотеки позволял использовать для трассировки два слоя металлизация в обоих направлениях.

Таблица 1

Выход топологий

Группа	% оттрассированных	% с портами
Группа 1	100	100
Группа 2	98	98
Группа 3	60	60

Таблица 2

Время выполнения

Группа	Среднее время трассировки, сек
Группа 1	311
Группа 2	1387
Группа 3	1564

В табл. 1 приведён процент ячеек, успешно прошедших этап трассировки (планирования и детальной трассировки), а также процент ячеек со всеми требуемыми портами. Во всех оттрасированных ячейках порты были выставлены автоматически, что подтверждает высокое качество планировки этих ячеек.

В табл. 2 представлено среднее время трассировки ячейки для каждой из групп.

Литература

[1] MARS—A Multilevel Full-Chip Gridless Routing System / Cong J., Fang J., Xie M., Zhang Y. // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. - 2005. - V. 24. - № 3. - P. 382 – 394.

[2] Кривошеин Д. Ю., Марченко А. М. Инкрементальный алгоритм поиска кратчайших путей в графе // Информационные технологии. - 2012. – № 7.

[3] Introduction to Algorithms / Cormen H., Leiserson C., Rivest R., Stein C. // The MIT Press. - 2009. - P. 700 – 704.

[4] De V.K., Kozminski K., Kedem G. A Heuristic Global Router for Polycell Layout // Custom Integrated Circuits Conference. - 1988. - P. 11.4/1 – 11.4/4.

[5] Volkov A. Impact of Manufacturing on Routing Methodology at 32/22 nm // ISPD'11 (Santa Barbara, California March 27-30, 2011). - 2011. - P. 139.

[6] Takahashi H., Matsuyama A., – An Approximate Solution for Steiner Problem in Graphs // Math. Japonica 24. - 1980. - № 6. - P. 573 – 577

[7] Казённов Г.Г., Щемелинин М.В. – Топологическое проектирование нерегулярных БИС // Высшая школа. - 1990. - С. 51 – 56.