

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ

УДК 621.38

Д. Г. Титов

(Новосибирск)

АЛГОРИТМЫ ИЕРАРХИЧЕСКОЙ ВЕРИФИКАЦИИ
ГЕОМЕТРИЧЕСКИХ ОГРАНИЧЕНИЙ НА ТОПОЛОГИЮ СБИС

Описаны базовый и иерархический алгоритмы верификации геометрических ограничений на топологию ИС, разработанные в рамках создания системы верификации проектируемых ИС в Институте автоматки и электротрии СО РАН.

Введение. При проектировании заказных и серийных интегральных схем (далее ИС) принято использовать как методы полного проектирования, так и методы, основанные на некоторой степени компиляции ранее спроектированных схемных и топологических блоков. Вторая группа методов проектирования, к которой относятся, например, трассировка вентиляльных матриц и метод стандартных элементов, имеет своей целью максимально упростить процесс проектирования ИС как с точки зрения вычислительных ресурсов, так и с точки зрения самой методологии проектирования и требуемой квалификации разработчиков. При этом подобные преимущества «оплачиваются» неэффективным использованием ресурсов кремниевой пластины (площадь и быстродействие), что делает весьма актуальным применение метода полного проектирования, при котором высококвалифицированным коллективом разработчиков создаются оптимизированные схмотехнические и топологические блоки, поддающиеся плотной упаковке в рамках архитектуры проектируемой ИС.

Однако подобный метод имеет большую трудоемкость и несет с собой повышенную вероятность появления ошибок в проекте ИС. Избавиться от этих недостатков позволяет высокоэффективный комплекс программных средств, осуществляющий «механизацию» ручного труда разработчиков и, главное, глубокую и всестороннюю верификацию проектных решений ИС. В Институте автоматки и электротрии СО РАН в течение 6 лет ведутся работы по созданию подобного комплекса. Настоящая статья описывает одну из его компонент — систему верификации геометрических ограничений на топологию ИС, представленную в иерархическом виде.

Топология ИС представляет собой набор топологических масок, каждая из которых используется в процессе производства ИС как фотомаска для нанесения на поверхность полупроводниковой пластины рисунка какого-либо материала. Типичное количество масок (или топологических слоев) в таком наборе 10—20 шт. при количестве элементов (прямоугольников), формирующих каждую из масок, около 1 млн и при сложности кристалла около 100 тыс. транзисторов. Нанесенные с помощью топологических масок взаимодействующие слои образуют транзисторы и прочие элементы и соединения между ними. Как правило, при разработке топологии стремятся обеспечить максимально допустимую в рамках используемой технологии плотность компоновки элементов кристалла, что вызывает необходимость размещения геометрических

ких фигур масок на предельно допустимых расстояниях друг от друга. Довольно обширный список соответствующих геометрических ограничений составляется фирмой-изготовителем на основе детальных экспериментов с технологическим оборудованием. Задача разработчика — обеспечить максимальную плотность компоновки ИС при нарушении ни одного из ограничений. Проверка топологии ИС на предмет нарушения подобных ограничений и является задачей описываемой системы.

Необходимо отметить основную трудность подобной задачи — большую вычислительную сложность проверки различных вариантов взаиморасположения геометрических фигур при их числе порядка десятков миллионов. В самом грубом приближении эта задача формулируется как анализ отношений взаимного расположения «каждый с каждым», причем число таких отношений на реальных топологиях достигает 10^{14} . Очевидно, что для решения такой задачи необходимо использовать алгоритмы, существенно сокращающие перебор и отсекающие заведомо ненужные проверки при небольших вычислительных затратах на подобные оптимизации.

Далее описаны методы и основные алгоритмы, разработанные и примененные в системе контроля (верификации) геометрических ограничений, являющейся частью программного комплекса разработки ИС Института автоматики и электрометрии СО РАН. В разд. 1 описываются основные алгоритмы, использующие тайловое представление данных [1], в разд. 2 — базовый алгоритм верификации геометрических ограничений в рамках отдельной тайловой плоскости, в разд. 3 обсуждается вопрос построения и оптимизации таблицы обобщенных правил. В разд. 4 содержится описание иерархического расширения базового алгоритма, предназначенного для работы с топологией реальных ИС, представление которых возможно только в иерархическом виде, разд. 5 посвящен средствам настройки системы на любые наборы геометрических ограничений (и вместе с тем на любую технологию производства ИС), в разд. 6 содержатся основные данные о реализации системы и измеренных параметрах производительности и ресурсопотребления.

1. Алгоритмы на тайловой плоскости. В основу системы верификации геометрических ограничений положено тайловое представление топологических данных, подробно описанное в [1, 2]. В целях иллюстрации напомним читателю основные положения тайлового метода представления данных.

В рамках тайлового представления геометрические фигуры располагаются в так называемых «тайловых плоскостях», которые являются по существу четырехсвязными списками, непосредственно отражающими взаиморасположение соседствующих геометрических элементов (рис. 1). При этом базовый элемент списков — прямоугольный четырехугольник, что предопределяет использование тайловых алгоритмов исключительно для так называемой «манхеттенской» топологии, т. е. топологии, в которой все углы образующих фигур кратны 90° .

В качестве лирического отступления можно заметить, что тайловый подход не ограничен лишь прямоугольным представлением и подобное ограниче-

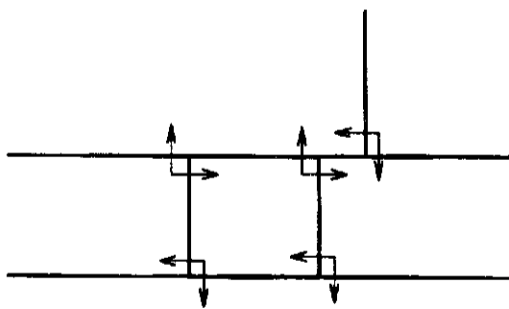


Рис. 1. Четырехсвязная структура тайловой плоскости

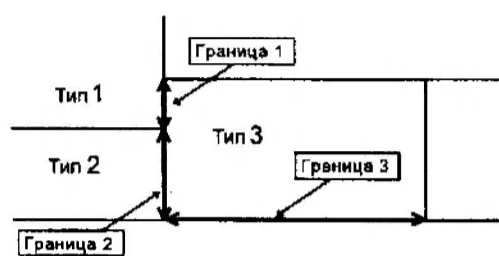


Рис. 2. Образование проверяемых границ

ние выбрано автором сознательно как средство упростить базовые алгоритмы. В ходе разработки были реализованы тайловые алгоритмы на 45°-ных фигурах (базовый элемент — трапезоид) и показана возможность реализации подобных структур для фигур с произволь-

ными углами. При этом время выполнения основных операций в случае 45°-ной реализации вдвое превышает аналогичное время для «манхэттенского» варианта, что является неприемлемой платой за возможность экономии 10—15 % площади кристалла.

Основными использованными автором тайловыми алгоритмами являются алгоритмы перечисления (итерирования) тайлов в прямоугольной области и, естественно, алгоритм построения тайловой плоскости.

Алгоритм итерирования тайлов в заданной прямоугольной области имеет вычислительную сложность cn , где n — количество тайлов, попавших в область. Вообще говоря, из [1] следует, что вычислительная сложность этого алгоритма может изменяться от $c1\sqrt{N} + c2n$ в случае случайной последовательности задаваемых областей итерирования (из-за накладных расходов на операцию InWhich) до cn в случае, когда область итерирования многократно сдвигается на расстояние, сравнимое с размером тайла, постепенно покрывая при этом всю тайловую плоскость. Именно такой вариант использования этого алгоритма и применяется в системе верификации геометрических ограничений.

Напомним, что тайловая плоскость представляет собой полностью покрытое тайлами различного типа пространство, причем незанятое, «пустые» области также покрыты тайлами типа «пусто» и каждая точка плоскости принадлежит ровно одному тайлу.

2. Верификация ограничений в плоскости. Базовый алгоритм проверки геометрических ограничений работает в одной тайловой плоскости. В процессе работы алгоритм итерирует все тайлы в этой плоскости (включая тайлы типа «пусто») и для каждого тайла строит множество проверяемых границ. Границей является отрезок левой или нижней стороны тайла, с каждой стороны которого находится лишь один тайл (рис. 2). Каждая граница проверяется в двух направлениях — внутрь тайла и вовне тайла (рис. 3). При этом из таблицы так называемых универсальных правил извлекаются все правила, соответствующие упорядоченной (в соответствии с направлением проверки) паре типов тайлов, разделенных данной границей. Формат правила показан на рис. 4. Правило состоит из двухэлементного вектора, идентифицирующего типы тайлов, разделенные границей, и трех полей, содержащих размер области проверки, множество типов тайлов, запрещенных в области проверки, и признак необходимости проверки угловой области. Каждое извлеченное из таблицы правило вызывает проверку области, отложенной от границы в направлении проверки на расстояние, указанное в правиле (рис. 5). При итери-

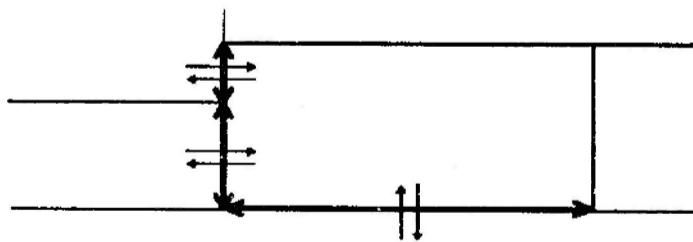


Рис. 3. Направления проверки границ (каждая граница проверяется в двух направлениях)

Тип 1	Тип 2	Расстояние (размер проверяемой области)	Множество запрещенных типов	Флаг: проверять ли угловую область?
-------	-------	---	-----------------------------------	--

Рис. 4. Формат универсального правила

ровании области проверки контролируется отсутствие в ней тайлов с типами, запрещенными соответствующим множеством текущего правила. Если при этом встречается тайл запрещенного типа, это вызывает построение в специальной тайловой плоскости (e-гог-плоскость) прямоугольника ошибки, совпадающего по размерам и местоположению с областью проверки. Если в правиле указано, что нужно проверить еще и угловую область, то аналогично проверяется еще и угловая область, расширяющая в одну сторону основную область проверки. Угловая область имеет квадратную форму с размером ребра, указанным в текущем правиле. Для тайлов запрещенного типа, обнаруженных в угловой области, проверяется угловое расстояние (см. рис. 5), вычисляемое по теореме Пифагора. Следует отметить, что это единственный случай, когда в системе верификации используются числа с плавающей точкой. Если вычисленное расстояние меньше размера, указанного в текущем правиле, фиксируется нарушение правил проектирования.

Каким же образом с помощью вышеописанных универсальных правил можно обеспечить проверку реальных геометрических ограничений? На рис. 6 показано несколько основных типов ограничений на топологию ИС. В таблице отражено соответствие между этими высокоуровневыми правилами и множествами универсальных правил, которые из них генерируются. Легко убедиться, что сгенерированные таким образом универсальные правила, действительно, обеспечивают проверку основных геометрических ограничений. Дополнительную гибкость в задании ограничений даст реализованный в системе механизм преобразования топологии ИС, подробно описанный в разд. 5 настоящей статьи.

3. Оптимизация набора правил. Описывая набор ограничений для конкретного технологического процесса производства ИС, разработчик не должен вручную составлять таблицу универсальных правил. Вместо этого составляется набор высокоуровневых правил, которые преобразуются в универсальные, как описано в предыдущем разделе. При этом из каждого высокоуровневого правила порождается несколько универсальных (в таблице множественное порождение универсальных правил заменено использованием символа "*", обозначающего «все типы тайлов, использованные в топологии»). Обычно в результате такого порождения образуется много универсальных правил, частично или полностью эквивалентных либо перекрывающих друг друга. Действительно, если имеются два правила, одинаковые во всем, кроме размера области проверки, то правило с меньшим размером можно просто исключить. Аналогично два или более правил с одним размером области проверки, но с разными множествами запрещенных типов можно заменить одним правилом, в котором объединены эти множества.

Процесс анализа таблицы универсальных правил с последующим уменьшением их числа за счет исключения повторяющихся и избыточных или слияния нескольких

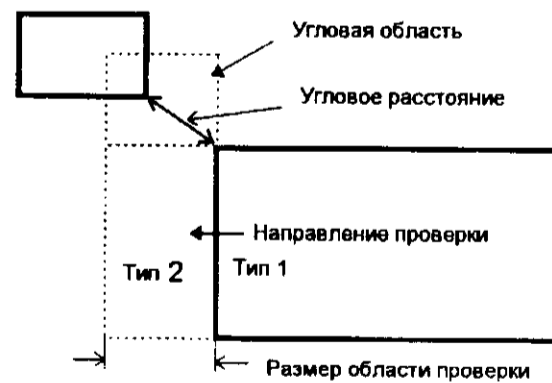


Рис. 5. Угловая область и угловое расстояние

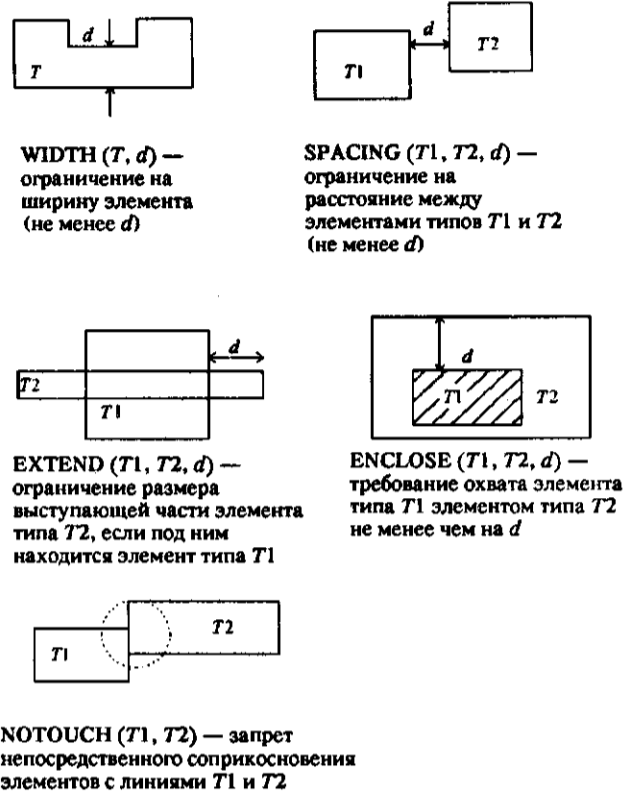


Рис. 6. Основные типы ограничений на топологию ИС

правил в одно (оптимизация таблицы универсальных правил) весьма полезен, так как позволяет в несколько раз сократить время проверки ограничений с помощью базового алгоритма проверки ограничений в плоскости. Так, в случае обычного набора геометрических ограничений на топологию КМОП разработчик имеет дело с 50—60 высокоуровневыми правилами, из которых порождается около 1000 универсальных правил, а после оптимизации остается примерно 350. Так как перечисление правил для каждой проверяемой границы является одним из самых глубоко вложенных (а следовательно, выполняемых максимальное число раз) действий, эффект от оптимизации набора универсальных правил проявляется как примерно трехкратное повышение производительности всей системы.

Правило	Тип 1	Тип 2	Расстояние	Запрещенные типы	Флаг проверки угловой области
WIDTH ($T1, d$)	* - $T1$	$T1$	d	* - $T1$	Требует проверки
SPACING ($T1, T2, d$)	$T1$	* - $T1$	d	$T2$	То же
EXTEND ($T1, T2, d$)	$T1$	$T2$	d	* - $T2$	—
ENCLOSE ($T1, T2, d$)	$T1$	*	d	* - $T2$	Требует проверки
NOTOUCH ($T1, T2$)	$T1$	$T2$	1	$T2$	—

4. Расширение алгоритма на иерархическую модель топологии. Топология ИС обычно бывает представлена в иерархическом виде с целью минимизировать объем оперативной памяти, используемой для ее представления. В иерархическом виде топология ИС представляет собой дерево вызовов ячеек. Ячейка является самостоятельным топологическим фрагментом, который может содержать как элементы топологии (геометрические фигуры, составленные из прямоугольников различных типов), так и вызовы других ячеек. При вызове ячейка может поворачиваться, зеркально отражаться относительно координатных осей, а также мультипликативно размножаться с образованием одно- или двумерного массива. При этом в памяти всегда хранится только одно представление каждой ячейки, сколько бы раз она не использовалась в топологии ИС. Вызов ячейки представляет собой своеобразный топологический примитив, в котором указывается имя вызываемой ячейки, матрица преобразования координат и, если нужно, информация для образования массива ячеек.

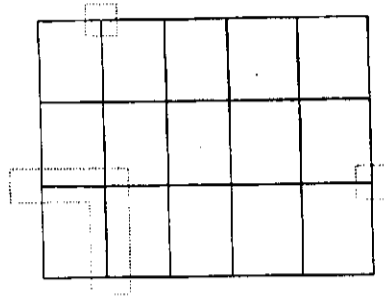
Очевидно, что базовый алгоритм проверки геометрических ограничений, описанный в разд. 2, работает только с «плоским» представлением топологии ИС, т. е. для его работы необходимо преобразовать вызовы ячеек в плоскую топологию путем замены вызовов реальной топологической информацией из всех нижележащих по иерархии ячеек («раскрыть» вызовы ячеек). Прodelать такую операцию на всей топологии ИС невозможно, так как это вызовет множественную проверку повторяющихся топологических фрагментов, образующих регулярные структуры (многоуровневые АЛУ, массивы памяти, ПЛМ и другие), что потребует огромных затрат памяти (на ИС сложностью около 100 тыс. транзисторов примерно 100 Мбайт) и заведомо сильно снизит производительность всей системы. Вместо этого применен метод восхождения по иерархическому дереву снизу вверх с проверкой лишь тех областей топологии, которые еще не проверялись.

На первом шаге алгоритма иерархической верификации с помощью базового алгоритма проверки ограничений в плоскости проверяется топология ячеек, лежащих в самом основании иерархии (т. е. не содержащих вызовов других ячеек). После этого для каждой вышележащей по степени вложенности ячейки осуществляется следующая последовательность действий:

- а) создается пустая временная ячейка;
- б) во временную ячейку копируются топологические элементы проверяемой ячейки («собственная» топология);
- в) для каждого топологического элемента из собственной топологии строится область, являющаяся результатом расширения этого элемента на максимальный размер проверяемой области из всех универсальных правил;
- г) итерированы все вызовы в проверяемой ячейке, и если вызываемая ячейка перекрывает какой-либо своей частью расширенную вокруг топологического элемента область, то в зоне перекрытия топология вызываемой ячейки (и если в зоне перекрытия в вызываемой ячейке есть вызовы еще более нижележащей ячейки, то и топология этих ячеек и т. д.) преобразуется в плоское представление;
- д) для каждого вызова в проверяемой ячейке строится область, расширяющая этот вызов на максимальный размер проверяемой области; если расширенные области вызовов где-либо пересекаются, то в области пересечения топология этих вызовов раскрывается на полную глубину;
- е) после раскрытия топологии во всех необходимых областях к полученной плоской топологии применяется базовый алгоритм проверки геометрических ограничений в плоскости.

Из приведенного описания алгоритма иерархической проверки следует, что он фактически раскрывает топологию вызовов в областях, где возможно взаимодействие (и нарушение ограничений) между элементами перекрывающихся или близко расположенных вызовов ячеек и/или собственной топологией проверяемой ячейки. Таким образом, раскрывается и проверяется не вся топология проверяемой ячейки, а только те ее участки, где возможны нарушения геометрических ограничений. Топология же нижележащих ячеек уже была проверена на предыдущих шагах восхождения по дереву иерархии.

Рис. 7. Области проверки двумерного массива ячеек



Описанный метод позволяет достичь двух основных результатов: сократить объем используемой оперативной памяти за счет лишь частичного раскрытия топологии ячейки и существенно сократить время проверки за счет практически полного исключения повторяющихся проверок.

При использовании в топологии ИС массива ячеек (т. е. вызовов одной и той же ячейки, расположенных в виде сетки с некоторым шагом) может быть применен дополнительный метод оптимизации: частичное раскрытие топологии массива. В этом случае расширенные области для каждого вызова ячейки в массиве не строятся, так как все вызовы в массиве пересекаются одним и тем же способом и нет нужды проверять все пересечения вызовов в массиве. Области раскрытия топологии в этом случае определяются, как показано на рис. 7. Как легко убедиться из рисунка, указанные области покрывают все возможные пересечения ячеек в массиве.

5. Настройка на произвольный набор ограничений. Как правило, набор геометрических ограничений на топологию ИС не удастся отобразить на модель единственной тайловой плоскости и таблицы универсальных правил. Вместо этого в системе используется несколько тайловых плоскостей на ячейку, в каждой из которых содержатся тайлы взаимодействующих между собой типов. Эти плоскости порождаются из исходной топологии ИС с помощью операций типа AND, OR, XOR, SIZING и т. д., последовательность которых образует программу преобразования топологии. Для каждой тайловой плоскости строится своя таблица универсальных правил, порождаемых, в свою очередь, из высокоуровневых правил типа WIDTH, SPACING и т. д. В совокупности высокоуровневые правила и программа преобразования топологии составляют так называемый технологический файл, который необходимо составлять для каждой используемой технологии производства ИС. Этим обеспечивается независимость программ системы верификации геометрических ограничений от используемой разработчиками технологии производства ИС.

6. Полученные результаты. Система верификации геометрических ограничений реализована на языке Си и содержит около 15 000 строк исходного текста. Текст системы переносим на любой 32-разрядный компьютер. Рабочая версия системы неоднократно тестировалась при разработке заказных ИС в Институте автоматизации и электротехники СО РАН. Результаты тестирования показали полную работоспособность разработанных алгоритмов и высокую производительность системы. Например, при использовании компьютера VAX-8530 система проверяет топологию ИС сложностью около 60 000 транзисторов за 50 мин процессорного времени, тогда как коммерческий пакет MASKAP затрачивает на ту же работу около 10 ч.

СПИСОК ЛИТЕРАТУРЫ

1. Лившиц З. А., Титов Д. Г. Алгоритмы работы с тайловыми представлениями топологии СБИС // Автометрия.—1991.—№ 3.
2. Ousterhout J. K. Corner stitching: a data structuring technique for VLSI layout tools // IEEE Trans. on CAD/ICAS.—1984.—CAD-3, N 1.—P. 87.

Поступила в редакцию 26 мая 1994 г.