

АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ

УДК 621.38 : 519.87

С. А. Фролов

(Новосибирск)

АЛГОРИТМЫ ГЛОБАЛЬНОЙ ТРАССИРОВКИ
МЕЖСОЕДИНЕНИЙ СБИС

Представлены алгоритмы, предназначенные для глобальной трассировки соединений при проектировании СБИС. Предложена процедура определения минимального пути, основанная на методе одновременного распространения волн из нескольких вершин графа. Приведены сведения о программной реализации алгоритмов.

Введение. Глобальная трассировка соединений — важный этап автоматизированного проектирования СБИС: на этой стадии проводится планирование всех межсоединений до определения их конкретной топологии. Входной информацией для программ глобальной трассировки являются сведения о расположении топологических ячеек в некоторой области, размерах этой области, а также информация об именах входов (портов) ячеек. При глобальной трассировке пространство, в котором допускается проведение межсоединений, делится на прямоугольные области — каналы. Каждое соединение может проходить через некоторую последовательность таких каналов, и задачей глобального трассировщика является определение конкретной последовательности каналов для всех заданных соединений.

Большинство программ глобальной трассировки основано на алгоритмах Ли [1] или Дейкстры [2]. Оригинальный волновой алгоритм Ли позволяет находить минимально возможное соединение между двумя вершинами графа. Алгоритмы нахождения минимального расстояния между тремя и более вершинами имеют эвристические обоснования. Особенности таких алгоритмов будут обсуждаться в разд. 2.

В данной статье представлены алгоритмы определения каналов и собственно глобальной трассировки, причем последний основан на методе, предусматривающем одновременное распространение волн из нескольких вершин графа.

1. Создание каналов. В этом разделе описан алгоритм разбивания всей области, отводимой под трассировку, на прямоугольные каналы.

1.1. *Структура данных.* Для представления геометрической информации, с которой работает алгоритм создания каналов, была выбрана предложенная в [3] списковая структура со сшиванием углов. В такой структуре (которую в дальнейшем будем называть тайловой) запись, соответствующая прямоугольной области (тайлу), содержит координаты левого нижнего угла тайла; четыре указателя на записи тайлов-соседей в нижнем левом и верхнем правом углу; тип прямоугольника, к которому относится этот тайл (канал или ячейка), и указатель на список портов. Для ячейки этот список полностью определен, так как порты ячейки известны заранее. Для тайлов типа «канал» этот список указывает на занятость портов только на сторонах, граничащих с ячейками. В процессе глобальной трассировки списки портов каналов заполняются в соот-

ветствии с тем, какое соединение проведено через канал. Каждый тайл типа «канал» имеет также «вес» — величину, которая характеризует геометрические размеры канала. На тайловых структурах эффективно реализуются различные поисковые операции: нахождение тайла, содержащего данную точку; обнаружение всех тайлов в области; нахождение всех тайлов, граничащих с данным тайлом и т. п., являющихся ключевыми в задачах трассировки.

1.2. Алгоритм создания каналов. Исходя из особенностей дальнейшей трассировки соединений, можно выделить требования, которым, по возможности, должен удовлетворять результат разбиения области на каналы.

Первое условие: поскольку сторона ячейки может граничить более чем с одним каналом, а порт имеет ненулевую ширину, то возможна ситуация, когда два канала *A* и *B* граничат с одним портом ячейки. При глобальной трассировке мы будем вынуждены учесть оба канала *A* и *B* как каналы, содержащие порты с одинаковым именем, между которыми должно быть проложено соединение. Чтобы избежать этого, каналы следует построить так, чтобы каждый порт ячейки граничил только с одним каналом.

Второе (необязательное, но желательное) условие: границы между каналами должны быть как можно более короткими. Выполнение такого условия позволяет уже на стадии глобальной трассировки определять ситуации, когда места для проведения трассировки не хватает, и выделять «узкое горло», через которое все требуемые соединения не могут быть проведены. В примере на рис. 1, *a* соединения могут «проникнуть» в канал *A* через протяженные границы и глобальная трассировка может быть закончена успешно, однако на стадии канальной трассировки возможно переполнение канала; в случае, приведенном на рис. 1, *b*, ситуация с «переполнением» канала будет обнаружена на этапе глобальной трассировки (*I—IV* — ячейки, *A, B, C, D, E* — каналы).

Третье желательное условие: необходимо избегать образования узких и длинных каналов. Появление таких каналов приводит к тому, что могут возникать искусственные ситуации «узкого горла». Подобные эффекты упоминаются в [4].

Процедура определения каналов, направленная на удовлетворение перечисленных выше требований, осуществляется следующим образом. Вся свободная от ячеек площадь делится на прямоугольники путем проведения линий вдоль границ ячеек до пересечения этими линиями либо другой ячейки, либо границы отводимой под трассировку области (см. рис. 2).

Далее для каждого порта ячейки проверяется, граничит ли он более чем с одним из образованных прямоугольников. Если граничит, то производится «слияние» этих прямоугольников. В примере на рис. 2 слиты каналы *A, B* и *C*. Таким образом удовлетворяется первое из требований (*I, II* — ячейки, *1, 2* — порты).

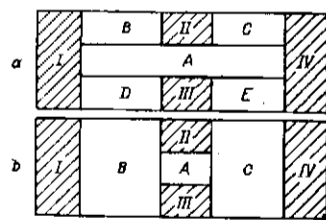


Рис. 1

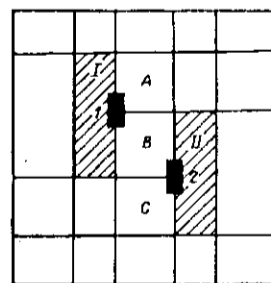


Рис. 2

Затем анализируются прямоугольники, которые могут быть слиты между собой с образованием прямоугольного канала. Для этого выбираются прямоугольники, границы между которыми имеют максимальную длину (стремление удовлетворить второму требованию: если слияние производится по возможно большей границе, то оставшиеся границы будут иметь меньшую длину). Если таких возможностей несколько, то сливается с другим тот прямоугольник, у которого самое большое отношение длины большей стороны к меньшей, и слияние будет производиться вдоль большей стороны. (Выполнение третьего условия: такое слияние ведет к уменьшению отношения большей стороны к меньшей, т. е. форма каналов приближается к квадратной.)

Все эти действия проводятся до тех пор, пока дальнейшее слияние становится невозможным.

2. Глобальная трассировка. Задача глобальной трассировки часто формулируется как задача нахождения минимального пути в графе, в котором, например, множество вершин соответствует ячейкам, а множество граней — каналам [5]. Каждой грани приписывается динамический вес, который зависит от размеров канала и занятости его предыдущей трассировкой. В описываемом алгоритме аналогом такого графа является тайловая структура, полученная после завершения работы программы создания каналов.

В основе алгоритма глобальной трассировки лежит модифицированный волновой алгоритм Ли [1]. Алгоритм Ли используется во многих программах [5] и позволяет находить минимальное расстояние между двумя вершинами графа. При трассировке часто возникает потребность в нахождении минимального соединения между тремя и более вершинами графа. Если последовательно находить минимальные расстояния между парами вершин, то полученное соединение не обязательно будет минимальным.

Некоторые известные процедуры после выполнения трассировки предусматривают проведение оптимизации полученных соединений [5]. В алгоритме, описанном в [6], перед выполнением глобальной трассировки для каждой из соединяемых вершин графа определяется приоритетное распространение фронта волны; затем получают все возможные минимальные соединения между всеми парами вершин и из нескольких таких соединений выбирается то, которое имеет максимально большое пересечение (общий участок пути) с другими полученными соединениями. Ниже рассматривается алгоритм нахождения кратчайшего пути между вершинами графа, когда волны из этих вершин запускаются и распространяются одновременно. Таким образом отпадает потребность предобработки информации о расположении вершин в графе или в корректировке полученных соединений.

2.1. Структура данных для распространения фронта волны. В описываемом алгоритме распространение волны происходит на тайловой структуре, полученной после завершения работы программы создания каналов. Каждый тайл типа «канал» (далее просто «тайл»), граничащий с портом ячейки, становится источником волны, если имя этого порта соответствует имени трассируемого в это время соединения. Назовем такой тайл исходным.

Последовательность трассировки соединений хранится в списке, каждый элемент которого содержит следующую информацию:

- имя соединения;
- предполагаемая площадь, которую займет соединение;
- число каналов, имеющих порты с именем этого соединения.

При распространении волны тайл, которого достиг фронт волны, становится источником новой волны и помещается в линейный список, называемый очередью. Для каждого исходного тайла заводится своя очередь. Элемент такого списка-очереди хранит следующие данные:

- указатель на тайл, соответствующий этому элементу очереди;

указатель на начало другой очереди (указывает на непустую структуру, если тайл, соответствующий этому элементу очереди, уже включен в другую очередь);

целое число, которое характеризует «расстояние» между соответствующим элементом тайлом и исходным.

2.2. Последовательность трассировки соединений. Для каждого соединения между портами ячеек с одинаковым именем определяются число каналов, которые граничат с этими портами, координаты минимального прямоугольника, покрывающего указанные каналы, и его площадь. Все эти сведения заносятся в список, организация которого описана выше. Полученный список сортируется по возрастанию числа каналов, причем элементы списка с одинаковым числом каналов располагаются в порядке возрастания площадей, занимаемых соответствующими соединениями. При проведении трассировки в таком порядке сначала будут разведены более простые и короткие соединения, затем более сложные. Аргументом в пользу подобной последовательности служит то, что для более простых соединений количество вариантов минимальных соединений меньше, и если ограничить их проведением более сложных соединений, то мы рискуем полностью лишиться возможности оптимальной реализации «коротких» трасс.

2.3. Алгоритм глобальной трассировки с распространением одной волны. Сначала рассмотрим, как находится кратчайшее соединение между двумя портами. Для этого осуществляется распространение волны из одного источника. На каждом этапе этого процесса определяется, какой из каналов становится принадлежащим фронту волны. Последовательность действий отражается при составлении очереди, структура которой приведена в п. 2.1. Создание каждого элемента очереди означает, что очередной тайл T включается во фронт волны; соответствующий элемент Q очереди содержит указатель на этот тайл T . (Для обозначения этого будем использовать запись $Q \rightarrow T$.) Если фронт волны не достиг цели, то для тайла T находится тайл T_1 (их может быть несколько) такой, что волна может распространиться из тайла T в тайл T_1 . Условия для возможности распространения:

1. T_1 не является тайлом типа «ячейка».
2. T и T_1 имеют общий отрезок границы Γ .
3. T_1 не был ранее включен в очередь.
4. На отрезке Γ существует не занятый другими соединениями порт.

Если все эти условия выполнены, то создается элемент очереди Q_1 , соответствующий тайлу T_1 . В поле «расстояние» элемента Q_1 ($Q_1 \rightarrow s$) записывается сумма веса тайла T_1 и целого числа, содержащегося в поле ($Q \rightarrow s$).

Если для тайла T существует несколько тайлов T_1, T_2, \dots , удовлетворяющих условиям 1—4, то сначала проводится создание всех новых элементов очереди Q_1, Q_2, \dots в отдельном списке L , список L сортируется по возрастанию величины $Q_1 \rightarrow s$, и лишь затем элементы списка L последовательно включаются в общую очередь. Это делается для того, чтобы получить всю очередь отсортированной по возрастанию величины s . По этой же причине, если возникает ситуация, когда перед включением списка L в очередь s первого элемента L больше, чем s последнего элемента очереди, необходимо провести сортировку всей очереди после включения списка L в очередь.

Поясним работу процедуры распространения фронта волны из одного источника на примере рис. 3. Здесь тайлы I и II — ячейки, A и B — порты ячеек, тайлы 1.1 ... 4.1 — каналы. Свободные порты каналов обозначены \leftrightarrow . Пусть тайл 1.1 — исходный, тайл 4.1 — целевой. В скобках записан вес каналов (рис. 3, а).

Для тайла 1.1 создается первый элемент очереди Q_1 , в поле s которого записывается 0. Распространяем фронт волны. Условиям 1—4 удовлетво-

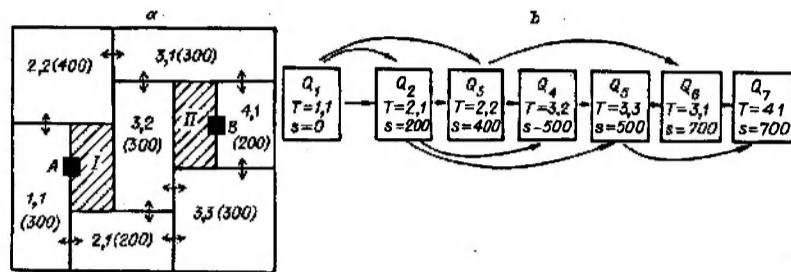


Рис. 3

ряют тайлы 2.1 и 2.2. Для тайла 2.1 $s = 0 + \text{«вес тайла 2.1»} = 200$, для тайла 2.2 получаем $s = 400$. В очередь последовательно ставятся элементы Q_1, Q_2 ($Q_2 \rightarrow 2.1$), Q_3 ($Q_3 \rightarrow 2.2$) (рис. 3, б).

Для тайла $Q_1 \rightarrow 1.1$ все соседние тайлы включены в очередь. Следующим элементом очереди является Q_2 , поэтому распространяем волну дальше от тайла $Q_2 \rightarrow 2.1$, создавая элементы очереди Q_4 ($Q_4 \rightarrow 3.2$; $Q_4 \rightarrow s = Q_2 \rightarrow s + 300 = 500$) и Q_5 ($Q_5 \rightarrow 3.3$; $Q_5 \rightarrow s = 500$).

При распространении волны из тайла $Q_3 \rightarrow 2.2$ получаем элемент Q_6 . Затем выбираем из очереди элемент Q_4 . Все тайлы, граничащие с тайлом $Q_4 \rightarrow 3.2$, уже включены во фронт волны, поэтому образования новых элементов очереди не происходит. Выбираем элемент Q_5 . Для тайла $Q_5 \rightarrow 3.3$ и тайла 4.1 выполняются все условия 1—4; создается элемент Q_7 . Тайл $Q_7 \rightarrow 4.1$ целевой, поэтому распространение волны прекращается. Путь из тайла 4.1 в тайл 1.1 определяется движением по списку Q_1, \dots, Q_7 в обратном порядке. Для этого из величины $Q_7 \rightarrow s$ вычитаем вес тайла $Q_7 \rightarrow 4.1$ и получаем 500. В очереди находим Q_5 , у которого $Q_5 \rightarrow s = 500$ ($Q_4 \rightarrow s$ также равно 500, но тайлы $Q_4 \rightarrow 3.2$ и $Q_7 \rightarrow 4.1$ не удовлетворяют условиям 1—4). Из величины $Q_5 \rightarrow s$ вычитаем вес тайла $Q_5 \rightarrow 3.3$, получаем 200. Находим Q_2 , у которого $Q_2 \rightarrow s = 200$. Из $Q_2 \rightarrow s$ вычитаем вес тайла $Q_2 \rightarrow 2.1$, получаем 0 и находим элемент очереди Q_1 ; $Q_1 \rightarrow 1.1$ — исходный тайл. Искомая последовательность каналов такова: $Q_7 \rightarrow 4.1$; $Q_5 \rightarrow 3.3$; $Q_2 \rightarrow 2.1$; $Q_1 \rightarrow 1.1$.

2.4. Алгоритм глобальной трассировки: одновременное распространение нескольких волн. Идею метода одновременного распространения волн можно проиллюстрировать следующим образом. Запустим из двух источников волны. До тех пор, пока волны не встретились, их фронты можно распространять независимо. Пусть в какой-то момент времени фронты этих волн встретятся в точке P . Новые фронты образуются испусканием волн из всех точек этих фронтов, которые можно рассматривать либо как общий фронт, либо как два отдельных фронта, причем точка P считается принадлежащей только одному из этих фронтов. Другими словами, распространение волн происходит независимо, если общие точки фронтов этих волн считаются принадлежащими только одному из фронтов.

При распространении волн одновременно из нескольких исходных тайлов для каждой волны заводится своя очередь. Каждая такая очередь строится подобно тому, как это описано в предыдущем разделе. Добавляются лишь некоторые новые условия распространения (дополнительно к условиям 1—4 из разд. 2.3) и учитывается появление общих тайлов для элементов из разных очередей. Для этого в элементах очередей используется «указатель на начало другой очереди» (см. разд. 2.1) (назовем это поле "pnt"). Поле "pnt" элемента Q i -й очереди указывает на начало j -й очереди, если элемент Q указывает на тайл, уже включенный в j -ю очередь.

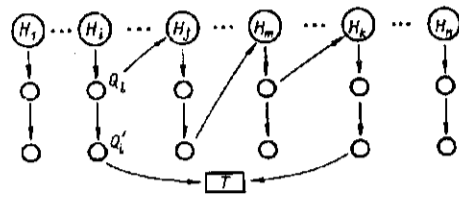


Рис. 4

Перечислим условия, дополнительные к условиям 1—4, при выполнении которых разрешается распространять фронт волны из тайла T в тайл T_1 . Пусть начало очереди, в которую мы собираемся вставить элемент Q_i ($Q_i \rightarrow T$), есть H_i (начало всех очередей $H_1, \dots, H_i, \dots, H_n$). Тогда пятое условие:

5. Если на тайл T уже существует указатель из элемента Q_k очереди H_k , но для ранее созданного элемента Q_i не существует цепочки вида:

- "*pnt*" элемента Q_i указывает на начало очереди Q_j ;
- "*pnt*" какого-либо элемента очереди Q_j указывает на начало очереди Q_m ;
- "*pnt*" какого-либо элемента Q_m указывает на ... начало очереди Q_k (рис. 4).

Тогда элемент Q_k очереди H_k помечается как «элемент, принадлежащий нескольким волнам, с разрешением распространения волны» (указатель "*pnt*" элемента Q_k будет указывать на H_i); разрешается создание элемента Q_i очереди H_i , который будет помечен как «элемент, принадлежащий нескольким волнам, с запрещением распространения волны». (Указатель "*pnt*" элемента Q_i будет указывать на H_k .)

Если же такая цепочка существует, то это означает, что волна H_i ранее достигла волны H_j , волна H_j уже достигла волны H_m , волна H_m ... достигла волны H_k , т. е. существует путь между источниками волн H_i и H_k более короткий, чем путь, включающий тайл T . Итак, условие 5 проверяется, если тайл T уже включен в другую очередь.

6. Распространение волны из тайла T запрещено, если в элементе очереди, указывающем на этот тайл, содержится информация о том, что этот «элемент принадлежит нескольким очередям с запрещением распространения волны» (см. условие 5).

7. Проверка: должно ли продолжаться далее распространение волн. Распространение волн прекращается, если для каждой очереди H_i , любой другой очереди H_k и какого-либо элемента Q_i очереди H_i существует цепочка, описанная в условии 5.

Распространение нескольких волн происходит следующим образом:

- а) для каждого исходного тайла создаются отдельные очереди H_1, \dots, H_n ;
- б) выбираются первые элементы очередей, и для соответствующих тайлов находят граничащие с ними тайлы так, чтобы выполнялись условия 1—6;

в) если выполняется условие 7, то следует закончить распространение волн, в противном случае выбираем следующие элементы очередей, возвращаемся на шаг «б» и т. д.

После завершения этого процесса находим путь, соединяющий исходные тайлы. Этот процесс проводится для каждой из очередей по отдельности точно так же, как и в предыдущем разделе, с тем отличием, что в качестве «целевых», соединяемых с исходным тайлом, выступают все те тайлы, для которых в соответствующих элементах очереди есть информация о том, что они принадлежат нескольким разным очередям.

Рассмотрим пример на рис. 5, где A, B и C — ячейки; $1-9$ — каналы; a, b, c — порты. Тогда исходные тайлы — 1, 2, 3. Строим три очереди: Q_a, Q_b и Q_c . Первые элементы очередей определяются последовательно в порядке возрастания весов соответствующих каналов (1, 2, 3). Для всех первых элементов очередей $s = 0$. Начинаем распространение

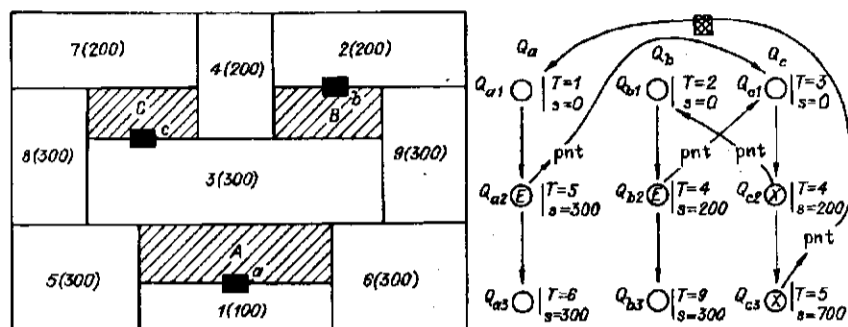


Рис. 5

волн для каждого исходного тайла. Из тайла 1 волна распространяется в тайлы 5 и 6; создаются элементы очереди Q_{a2} , Q_{a3} . Из тайла 2 — в тайлы 4 и 9; получаем Q_{b2} , Q_{b3} . Из тайла 3 — в тайлы 4 (это общий тайл для очередей Q_b и Q_c ; для этих очередей нет общих тайлов с меньшими величинами веса и s ; тогда тайл 4 считается «соединяющим» очереди Q_b и Q_c , причем из элемента очереди Q_{c2} (конечный элемент) запрещается распространение волны, а из Q_{b2} (также конечный элемент) — разрешается. Точно так же находится и тайл 5 — общий для очередей Q_a и Q_c соответственно, элементы очередей Q_{a2} (конечный с разрешением распространения волны) и Q_{c3} (конечный — с запрещением). При проверке условия 7 устанавливается, что все волны теперь соединены между собой, и их распространение прекращается. В каждой из очередей находятся конечные элементы (в очереди Q_c таких элементов два — Q_{c2} и Q_{c3}), и для соответствующих этим элементам тайлов проводится соединение их с исходными тайлами (точно так же, как и в предыдущем разделе). Получаем соединение между портами a , b и c , которое проходит через каналы 1, 5, 3, 4, 2.

3. Примеры результатов трассировки. Типичные характеристики работы программ, созданных на основе описанных в разд. 1 и 2 алгоритмов, показаны в таблице. Временные параметры приведены к производительности компьютера VAX 11/780. В качестве примера для «простого случая» трассировки (тест 1) выбрана схема на рис. 6; для более сложного случая (тест 2) трассировки 100 разноименных соединений (по шесть выводов у каждого) — схема на рис. 7, где также показана последовательность каналов для одного из соединений. На рис. 6 приведены также места пересечения соединения границ каналов ($I-VI$ — ячейки; a, b, c — соединения; $1-9$ — каналы; темные прямоугольники — порты ячеек; темные квадраты — пересечения соединениями границ каналов).

Заключение. В данной работе описаны алгоритмы создания каналов и глобальной трассировки при проектировании ИС. Использована «тайловая» структура данных, позволяющая эффективно проводить различные поисковые операции и естественным образом содержащая

Схема	Количество колонок	Количество каналов	t_1, c	t_2, c	t_3, c
Тест 1	6	10	0,11	0,07	0,20
Тест 2	67	132	5,22	16,10	22,54

Примечание. t_1 — время работы программы создания каналов; t_2 — время работы программы глобальной трассировки; t_3 — общее время работы (включая чтение данных и запись результата в файлы).

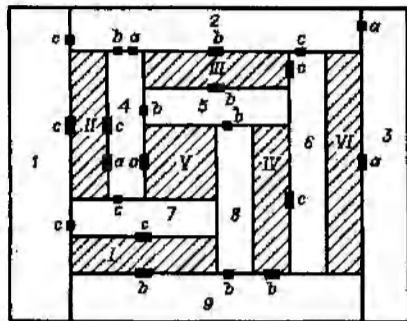


Рис. 6

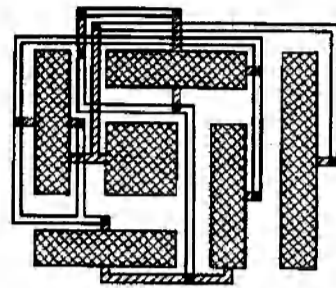


Рис. 8

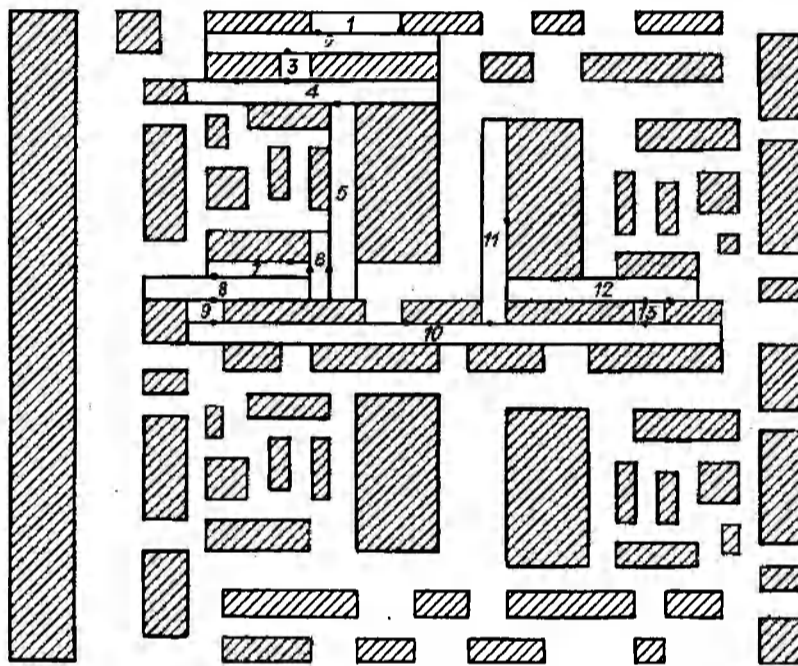


Рис. 7

необходимую геометрическую информацию. Уменьшение сложности алгоритма глобальной трассировки достигается за счет отсутствия действий по оптимизации уже проведенных соединений.

При описании алгоритма глобальной трассировки показано, как, проводя последовательные операции, можно получить эффект одновременного распространения волн от нескольких вершин в графе при поиске кратчайшего расстояния между этими вершинами. Используемый при этом метод может применяться и для решения других подобных задач на графах. В частности, алгоритм был опробован для бессеточной канальной трассировки; на рис. 8 приведен простой пример результата работы такой программы.

СПИСОК ЛИТЕРАТУРЫ

1. Lee C. An algorithm for path connection and its application // IRE Trans. on EC.—1961.— N 9.
2. Дал У., Дейкстра Э., Хоор К. Структурное программирование.—М.: Мир, 1975.
3. Ousterhout J. K. Corner-stitching: a data-structuring technique for VLSI layout tools // IEEE Trans. on CAD.—1984.—CAD-3, N 1.—P. 87.

4. Ousterhout J., Hamachi G., Mayo R. et al. MAGIC: a VLSI layout system // Proc. 21th Design Automation Conference, 1984.—P. 152.
5. Advanced in CAD for VLSI.—V. 4: Layout Design and Verification /Ed. by Ohtsuki.—Elsevier Science Publishers B.V., 1986.
6. Xiong J. G. Algorithms for global routing // Proc. 23rd Design Automation Conference, 1986.—P. 824.

Поступила в редакцию 8 апреля 1991 г.

УДК 621.38 : 519.87

А. Г. Рябченко

(Новосибирск)

МІСЕ2: ПРОГРАММА ЭКСТРАКЦИИ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ ИЗ ОПИСАНИЯ ТОПОЛОГИИ МОП СВІС

Рассмотрены принципы организации программы, обеспечивающей выделение списка соединений между транзисторами и электрических параметров схемы из описания ее топологии. Благодаря использованию тайловой структуры представления данных о геометрии масок, а также учету иерархии, возникающей при проектировании, были достигнуты высокие скоростные характеристики работы программы и обеспечена линейная зависимость времени экстракции от сложности схемы.

Введение. Определение параметров электрической схемы, которая реализуется заданной топологией БИС (экстракция электрической схемы), является необходимым этапом современных методологий проектирования, нацеленных на обеспечение отсутствия ошибок в разрабатываемых устройствах.

Программы экстракции (экстракторы) различаются между собой по следующим признакам: тип топологии, с которой они могут работать (с произвольными углами, углами, кратными 45 или 90°); типы транзисторов, распознаваемые ими (ТТЛ, МОП и др.); степень детализации выдаваемых данных (информация о списке соединений транзисторов, их размерах, величинах паразитных емкостей, сопротивлениях проводников и т. д.) [1—5].

Рассматриваемая в данной статье программа экстракции МІСЕ2 является частью разрабатываемой в Институте автоматики и электрометрии СО АН СССР интегрированной системы топологического проектирования, использующей единую базу геометрических данных на основе структур тайлового типа [6, 7]. Экстрактор предназначен для работы с «манхэттенскими» топологиями МОП-схем. Уровень подробности информации, получаемой с помощью МІСЕ2, определяется характером ее дальнейшего использования при верификации схемы: в настоящее время это в основном логическое моделирование на переключательном уровне с применением программы SimSim [8], входным для которой является формат esim [9]. Для этих целей достаточно следующих данных о схеме: а) список соединений транзисторов; б) размеры транзисторов; в) величины емкостей узлов относительно подложки.

Одной из задач, которая ставилась при разработке данной программы, было максимальное использование возможностей тайловых структур (простота определения смежных объектов, наличие «символических» слоев и т. д.) для достижения высоких показателей скорости экстракции.

1. МІСЕ2 — структура и алгоритмы. Рассматриваемая программа работает с иерархически представленной топологией, т. е. данные о каждой ячейке могут содержать как собственно информацию о геометрии масок, так и вызовы подъячеек. Важно подчеркнуть, что в МІСЕ2 процесс