**РЕФЕРАТ  
на тему: "Алгоритмы трассировки "**

**Введение**

В настоящее время используются различные варианты волнового алгоритма, в частности, лучевой и маршрутные.

Простейшим видом волнового алгоритма является волновой алгоритм нахождения кратчайшего пути без пересечения множества занятых и запрещенных элементов (участков печатной платы). Его целесообразно использовать при трассировке соединений в одной плоскости, когда недопустимо выходить из пределов этой плоскости. Определяются начальная и конечная точки и моделируется распространение волны от конечной точки к начальной в направлении волны. Недостатком этого алгоритма является то, что он мало пригоден для трассировки многослойных печатных плат, проводники прокладываются по краям платы, значительное число длинных параллельных проводников являются причиной большой взаимоиндуктивности.

Более совершенным волновым алгоритмом является волновой алгоритм прокладки пути с минимальным числом пересечения. В этом случае число пересечений ранее проложенных трасс должно быть минимальным. Для преодоления недостатка этого алгоритма, при котором трассы стремятся к одной из границ платы и прижимаются друг к другу, был предложен алгоритм для проведения пути, минимально приближающихся к другим трассам. Основой алгоритма является условие, при котором элементы данного соединения должны иметь минимум соседних элементов, принадлежащих ранее проложенным трассам.

Если одним из условий является требование регулярности соединений (один слой горизонтальные, другой – вертикальные и т.п.), то удобнее использовать волновой алгоритм прокладки пути с минимальным числом изменений направления, который позволяет минимизировать количество межслойных соединений.

В отличие от волновых и лучевых алгоритмов, в которых на начальной стадии перебираются все возможные варианты трассы, в маршрутных алгоритмах прокладка трассы ведется сразу и по кратчайшему маршруту.

# Маршрутный алгоритм трассировки

Каждый слой платы представлен в памяти ЭВМ булевой матрицей, элементы которой имеют значение 0, если соответствующий элемент свободен для прокладки пути, и имеют значение 1, если соответствующий элемент занят. Все элементы матрицы, которые принадлежат исходным препятствиям, задаются единичным значением.

Алгоритм реализует следующие последовательно выполняемые этапы:

1. построение пути до встречи с препятствием;
2. обход препятствий;
3. минимизация построенного пути.

**Этап 1.** Пусть требуется проложить путь между элементами da, булевой матрицы, описывающей модель платы. При отсутствии препятствий между элементами можно проложить конечное множество путей, имеющих минимальную длину в выбранной геометрии. Процесс построения Р-пути (Н-пути) сводится к тому, чтобы определить такую последовательность элементов L=<da, da+1,…, dk,…, db>, что любой элемент dk принадлежит Р-окрестности (Н-окрестности) элемента dk-1.

Если будем рассматривать Н-окрестность, то вектор перехода Zk от элемента dk к элементу dк+1 возможен только в направлениях, параллельных координатным осям. Для случая Р-окрестности вектор перехода может иметь диагональные направления.

На каждом шаге построения пути направление вектора перехода Zk от элемента dk к элементу dк+1 определяется функциями sgn(xb-xk), sgn(yb-yk), где xb, yb - координаты элемента db пути, xk, yk - координаты элемента dk.

Правило выбора направления построения пути до встречи с препятствием в наилучшем направлении приведено в таблице 1.

Таблица 1.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Функция | Zk | | | | | | | |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| sgn(xb-xk) | 1 | 1 | 0 | -1 | -1 | -1 | 0 | 1 |
| sgn(yb-yk) | 0 | 1 | 1 | 1 | 1 | -1 | -1 | -1 |

Наименование направлений приведено на рисунке 1.

|  |  |  |
| --- | --- | --- |
| 3 | 2 | 1 |
| 4 | A | 0 |
| 5 | 6 | 7 |

Рисунок 1. Наименование направлений вектора перемещения Zk.

После каждого шага построения пути необходимо по вектору перехода Zk определить состояние очередного элемента dk (свободный или занятый) булево матрицы. Рассмотрим построение Н-пути.

Для описания дискретного пространства, в котором строим путь, используем булеву матрицу С размером m×n. Кроме того, для сокращения вычислений введем усеченную матрицу А размером m×l. Число строк в матрице А определяется шириной прокладываемого проводника в дискретах. При прокладке проводников шириной в один дискрет матрица А будет матрицой-строкой, только один элемент которой принимает единичное значение. Номер этого элемента определяется координатой xk анализируемого элемента dk.

Состояние элементов описывается через булеву функцию

,

где ci,j – элемент матрицы С; ai - элемент матрицы-сторки А.

Здесь через индекс j обозначается номер строки матрицы С, который определяется координатой yk элемента dk.

Если V=1, то элемент dk занят, и построение пути прекращается. Дальнейшее построение осуществляется путем обхода препятствий, начиная с элемента dk-1, который будем называть элементом встречи с препятствием.

При построении Р-пути распознавание состояния элемента выполняется в два этапа. На первом этапе определяем, принадлежит ли элемент dk какому-либо объекту, записанному в матрице С. Если элемент dk не принадлежит никакому объекту, то переходим к выполнению второго этапа, суть которого сводится к следующему: определяем состояние элементов, которые принадлежат одновременно Н-окрестностям элементов dk, dk-1. Таких элементов может быть только два, причем они расположены диагонально. Если оба элемента заняты, то построение пути из элемента dk-1 в dk запрещено.

При построении пути в диагональных направлениях состояния элементов описывается булевой функцией

, i=1, 3, 5, 7. (1)

Булевы функции Vi, Vi-1, Vi+1 определяются при просмотре   
Р-окрестности элемента dk. Если функция (1) равна нулю, то выбранный элемент свободный; в противном случае – занятый.

Если очередной элемент dk булевой матрицы С, через который должен пройти путь занят, то из элемента dk-1, который назовем элементом встречи с препятствием (на рисунке 2 это элемент 1), начинается обход препятствия.

**Этап 2.** Переход от элемента встречи с препятствием к следующему свободному элементу пути выполняются согласно правилу первого шага.

*Правило первого шага.* Этап обхода препятствия начинается с элемента dk встречи с препятствием в направлении Zk, двоичный код которого определяется путем сложения кода предшествующего направления (Z’)k-1 с кодом 001 по модулю 8 при отрицательном направлении обхода препятствий, а при положительном обходе – с кодом 111.

Если выбранное направление запрещено, то принимаем первое возможное направление.

При построении пути выполняется отрицательный (правый) и положительный (левый) обход всей группы препятствий, лежащих между конечными элементами пути. В этом случае у первого элемента встречи с препятствием путь разветвляется на два. По одному пути осуществляется обход препятствий справа, а по другому – слева.

При построении Н-пути для обхода препятствий используется алгоритм Н-слежения, а при построении Р-пути – Р-слежение.

При отрицательном направлении Р-слежения двоичный код приоритетного направления опреднляется соотношением

,

а при положительном

.

Если направление с высшим приоритетом запрещено, то выбирается первое возможное направление с низшим приоритетом. Определяемое соотношением

,

где n – двоичный код чисел из последовательности 1, 2, …,8.

Суммирование по модулю 8 выполняется при отрицательном направлении слежения, вычитание – при положительном.

Важным моментом является определение элемента, в котором заканчивается обход препятствий и начинается построение пути в оптимальном направлении (по прямой к элементу db). Если в нужный момент не прекратить обход препятствий, то неизбежно зацикливание пути вокруг препятствий. Элемент пути, в котором прекращается обход препятствий, назовем элементом спуска. На рисунке 2 элементом спуска является элемент 19. Здесь приведен путь в лабиринте, построенный согласно этой методике от элемента da к элементу db. От элемента da до элемента 1, который является элементом встречи, выполняется построение пути согласно этапу 1. Обход препятствий начинается от элемента встречи 1 в отрицательном направлении (этап 2) и заканчивается элементом спуска 19. От элемента спуска 19 до конечного элемента пути выполняется   
этап 1.

Для определения элемента спуска пути предлагается следующий алгоритм:

1. определяем двоичный код угла поворота вектора перехода относительно вектора Z’ из соотношения

;

причем суммирование выполняется при отрицательном направлении обхода препятствий, вычитание – при положительном.

1. В каждом элементе излома проверяем значение двоичного кода ak и направление построения пути в наилучшем направлении. Если ak≥0 и направление обхода препятствий совпадает с наилучшим направлением построения пути, то элемент dk будет элементом спуска. В противном случае dk не является элементом спуска.

**Этап 3.** Минимизация длинны пути сводится к построению выпуклого контура, описанного вокруг первоначального пути. Если нет возможности получить выпуклый контур из-за наличия препятствий, то строится сглаженный контур, т.е. контур, имеющий меньшую длину, чем первоначальный.

Находим все элементы спуска первоначального пути и разбиваем его на отдельные участки, разграниченные элементами спуска. Последовательно минимизируем все участки пути.

1. Находим все элементы излома соответствующего участка пути, и если имеется не более одного излома, то он не подлежит минимизации если элементов излома два и более, то минимизация заключается в том, что строится новый путь Lн(da, dj) пути L(da, dj), где dj - элемент излома пути, самый близкий к конечному элементу.
2. Построенный вновь подпуть Lн(da, dj) сравнивается по длине с путем L(da, dj), и если новый путь меньше, то L(da, dj) заменяется на Lн(da, dj).
3. Минимизация повторяется для следующего элемента излома, самого близкого к dj, и до тех пор, пока на Lн(da, dj) или L(da, dj) останется один элемент излома.

Осуществляется минимизация обоих первоначально построенных путей, полученных при положительном (левом) и отрицательном (правом) обходе группы препятствий, из которых выбирается минимальный   
(рисунок 3).

# Волновой алгоритм трассировки.

Дискретное поле платы разбивают на три множества, описываемых с помощью булевых матриц:

С – множество элементов поля, требующих соединения между собой (на рисунке 4 множество , где i=0, 1, 2, 3);

Р – множество элементов поля, запрещенных для трассировки (на рисунке 4,а(б) это множество закрашено черным);

S – множество свободных элементов поля платы.

Требуется, используя элементы множества S, соединить элементы множества С в одну цепь, не пересекающую Р.

Процесс нахождения минимального пути состоит из двух этапов:

* Распространение волны от источника до встречи с одним из приемников;
* Определение пути от источника к приемнику.

В качестве источника выбирается один из элементов множества С, все остальные элементы являются приемниками. Обозначим через Rk множество элементов волны на шаге k и назовем его k-фронтом волны, тогда Rk+1 принадлежит Н-окрестности Rk.

На каждом шаге расширения делается проверка пересечения фонта волны с приемником. Как только какой-либо элемент приемника будет включен в волну, процесс распространения волны завершается, и от ближайшего к источнику элемента приемника начинается построение пути.

Для построения волны используются матрицы распространения волн в горизонтальном (Rk’), и вертикальном (Rk) направлении и матрицы точек поворота с вертикального направления на горизонтальное (Mk) и с горизонтального направления на вертикальное (Mk’), где

;

;

.

На рисунке 5, а - г приведены соответственно матрицы Rk, Rk’, Mk, Mk’, построенные для k=12. Источником является фрагмент С0. Для наглядности в клетках, занятых волной, указывается номер шага, на котором достигнута эта точка.

На этапе построения пути определяется, каким фронтом достигнут приемник: горизонтальным или вертикальным. От элемента пересечения волны с приемником в направлении, соответствующему последнему фронту волны, определяем элементы, не принадлежащие множеству Р. При встрече с первым же элементом множества точек поворота рассмотренного направления движение прекращается. Все пройденные элементы включаются в искомый путь. Элемент встречи принимается за исходный для движения по другому фронту волны.

Направление чередуется таким образом до тех пор, пока не будет достигнут элемент источника.

На рисунке 5, д показан пример построения пути от приемника С1 к источнику С0. Стрелками показано направление движения. В дальнейшем процесс распространения волны повторяется с учетом построенного пути. Такой выбор источника обеспечивает ветвление цепи в любой ее точке.

Процесс заканчивается, когда все элементы С будут связаны в единую цепь (рисунок 5, е), либо когда оставшиеся элементы этого множества уже не могут быть присоединены к источнику.