

УДК 681.32

**О.Е. Смирнов, В.Н. Струнилин**  
Донецкий национальный технический университет  
кафедра компьютерных наук и технологий

## **РАЗРАБОТКА ПРОГРАММЫ-ТРАССИРОВЩИКА ПЕЧАТНЫХ ПЛАТ**

### *Аннотация*

*Смирнов О.Е., Струнилин В.Н. Разработка программы-трассировщика печатных плат. Выполнен обзор алгоритмов трассировки печатных плат. Разработана программа для трассировки печатных соединений различными способами.*

*Ключевые слова:* трассировка печатного монтажа, волновой алгоритм, дискретное рабочее поле (ДРП).

### **Постановка проблемы.**

Современные САПР сквозного проектирования средств вычислительной техники имеют многомодульную структуру и выполняют разработку принципиальных и монтажных схем, печатных плат (ПП), компоновку и авторазмещение элементов изделий, автотрассировку. Наиболее сложной задачей топологического проектирования является трассировка монтажных соединений между конструктивами на всех уровнях (соединение микросхем на плате, монтажные и межблочные соединения). Трассировкой ПП является определение оптимального пути между контактными площадками ПП. Критерии оптимальности могут быть различными и зависят от технологического процесса производства ПП, ее функционального назначения и прочих параметров. В настоящее время существует множество программ для трассировки, такие как P-CAD, DipTrace, Ansoft и другие. В этих программах рассмотрен один или несколько волновых алгоритмов, которые требуют больших временных затрат и требуют большого объема памяти. В то же время в ряде случаев возникает необходимость трассировки платы несколькими методами, которые наиболее оптимальны по тем, или иным параметрам.

**Анализ литературы.** Проведен анализ методов трассировки печатных соединений [1]. Это такие методы, как волновой алгоритм Ли и его модификации (метод Акерса, метод встречной волны и другие), лучевые алгоритмы, а так же их оптимизация по двум параметрам – минимальному числу пересечений и минимальной длине. Учитывая вышеизложенное, задача разработки единой универсальной программы для трассировки печатного монтажа различными методами и их исследование является актуальной. Ограничимся только определением оптимального пути (построение проводника) между контактными площадками.

**Цель статьи** — разработка программы-трассировщика печатных плат различными методами, оптимальными по времени выполнения, объему используемой памяти, минимальному числу пересечений и минимальной длине проводников.

**Решение задачи и результаты исследований.** Каноническим вариантом определения кратчайшего пути между двумя точками на матрице является волновой алгоритм. Для реализации данного алгоритма следует ввести ограничение: размер дискретного коммутационного поля, на котором будет производиться трассировка, равен  $n \times m$ . Исходя из данного ограничения, реализация волнового алгоритма представляет собой тривиальную задачу.

Рассмотрим принцип волнового алгоритма [1]. Основой волнового алгоритма является процедура построения оптимального по заданным критериям пути между парой ячеек дискретного рабочего поля, что соответствует соединению пары контактов. Процедура состоит из 2х этапов: поиск пути и проведение пути.

Реализация волнового алгоритма относительно проста и показана на рисунке 1.

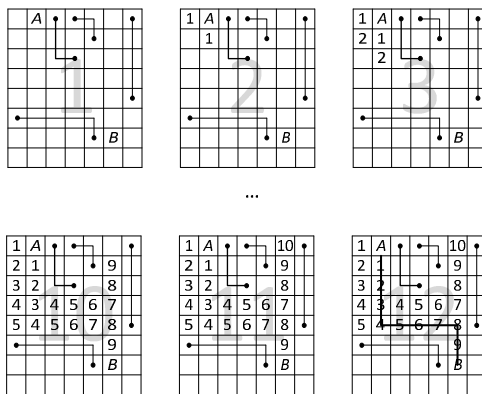


Рисунок 1 – Пример волнового алгоритма трассировки

На первом этапе из одной заранее свободных ячеек А (источник) моделируется распространение числовой волны до тех пор, пока её фронт не достигнет заданной второй ячейки В (приёмник), либо это никогда не произойдёт. Образование очередного фронта волны  $\Phi_k$  ( $k = 1, 2, \dots$ ) начинается с присвоения всем свободным ячейкам, соседним с ячейками фронта  $\Phi_{k-1}$ , веса  $P_k = P_{k-1} + \Psi(f_1, \dots, f_2)$ , где  $\Psi$  — весовая функция, которая является показателем качества проведения пути. Каждый параметр  $f_i$  характеризует путь с точки зрения одного из критериев качества (длина пути,

число пересечений и т.д.). При этом должно выполняться следующее ограничение:  $P_{k-1} \leq P_k$ .

Соседними считаются ячейки, которые имеют с ячейками предыдущего фронта либо общую сторону, либо хотя бы одну общую точку. Для простоты будем считать соседними те ячейки, которые имеют общую сторону (ребро) и  $P_k = P_{k-1} + 1$ . Итак, в результате распространения числовой волны последовательно получим фронты  $\Phi_1(A)$ ,  $\Phi_2(A)$ , ...,  $\Phi_k(A)$ . Если ячейки фронта  $\Phi_k$  содержат ячейку  $B$ , то кратчайший путь между  $A$  и  $B$  существует. После этого осуществляем построение пути, двигаясь в обратном направлении от  $B$  к  $A$  и переходя по соседним ячейкам так, чтобы вес  $P$  монотонно уменьшался на 1. При реализации волнового алгоритма важная проблема — сокращение объема памяти, необходимой для запоминания весов ячеек.

В отличие от волновых алгоритмов в лучевых алгоритмах рассматривается не вся площадь дискретного рабочего поля, а только те ячейки, которые лежат на заранее заданных направлениях — лучах, что, естественно, сокращает число просматриваемых ячеек дискретного рабочего поля, однако приводит к тому, что пути сложной конфигурации могут быть не обнаружены [1].

Для программной реализации процесса трассировки используем объектно-ориентированную модель. Согласно данной модели следует выделить все основные объекты, которые будут использоваться при реализации процесса трассировки:

- контактная площадка, которая должна располагаться на поле трассировки;
- проводник, соединяющий две дискретного рабочего поля трассировки;
- поле трассировки размером  $n$  на  $m$ , на котором будут располагаться контактные площадки и проводники;
- связь, которая содержит в себе информацию о соединении двух контактных площадок и является комбинацией контактных площадок и наборов проводников.

Результат работы программы выводится в графическом виде. Из этого следует, что при проектировании интерфейса следует учесть необходимость организации удобного и наглядного отображение графических данных.

Наибольшую область главного окна программы занимает поле трассировки. Это необходимо для более наглядного отображения разведенных связей.

Управление исходными данными трассировки (списком контактных площадок и списком элементов схемы) следует организовать в отдельных окнах. Элементы управления программой следует расположить на панели управления.

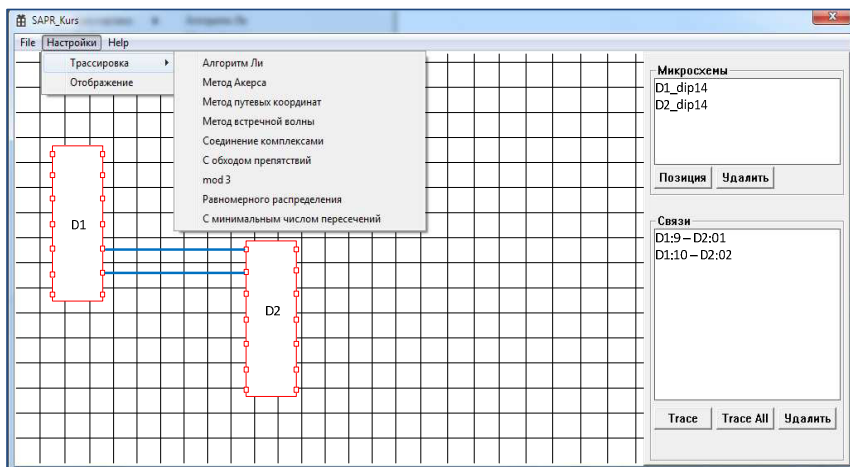


Рисунок 2 – Интерфейс программы менеджера трассировки

Непосредственное управление процессом трассировки обеспечивает менеджер трассировки, который показан на рисунке 2. Пользователь может выбирать один из доступных методов трассировки, а также вывести ДРП с элементами и оттрассированными проводниками на экран.

Программа разрабатывалась в среде Microsoft Visual Studio 2012 на языке C++ [2].

С использованием разработанной программы были выполнены исследования зависимости разрядности для кодирования ячеек ДРП от размера ПП (рис. 3).

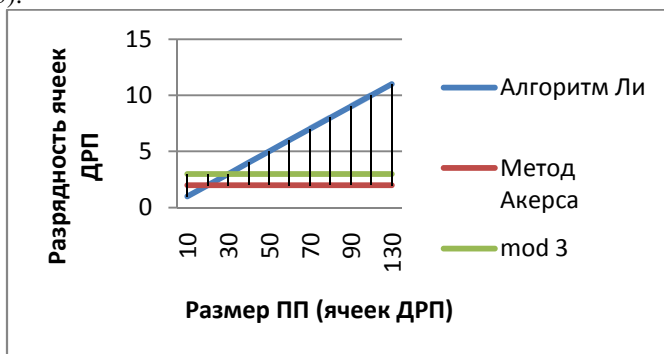


Рисунок 3 – Зависимость разрядности ячеек ДРП от размера ПП

**Выводы.** Преимущество разработанной программы по отношению к существующим программам заключается в том, в данной программе реализуются большинство основных волновых и лучевых алгоритмов трассировки печатного монтажа. Разработанная программа позволяет выполнять трассировку на дискретном рабочем поле, которое задается пользователем. Проведенные исследования позволяют сделать вывод, что используемые методы уменьшают объем памяти для кодирования ячеек дискретного рабочего поля в 2-10 раз (чем больше степень интеграции ПП, тем больше коэффициент эффективности реализованных методов). В реальных схемах этот показатель может значительно увеличиться. Кроме того, реализованные методы позволяют минимизировать суммарную длину соединений и количество пересечений проводников, а следовательно, и число слоев многослойных печатных плат.

### Список литературы

1. Автоматизированное проектирование узлов и блоков РЭС средствами современных САПР: Учеб. пособие вузов/Под ред. И. Г. Мироненко. – М., 2002. – 391 с.
2. Прата, С. Язык программирования C++. Лекции и упражнения / С. Прата. – Вильямс, 2011. – 1244 с.