

Г. А. Петухов
Г. Г. Смолич
Б. И. Юлин

Алгоритмические
методы
конструкторского
проектирования
узлов
с печатным
монтажом



Москва
«Радио и связь»
1987

ББК 32.844
П 31
УДК 621.396.6

Петухов Г. А. и др.

П 31 Алгоритмические методы конструкторского проектирования узлов с печатным монтажом / Г. А. Петухов, Г. Г. Смолич, Б. И. Юлин. — М.: Радио и связь, 1987. — 152 с.: ил.

Рассмотрены вопросы разработки согласованных критериев и методов, на основании которых реализованы алгоритмы основных этапов проектирования печатных плат. Кратко изложен математический аппарат, используемый авторами для оптимизации алгоритмов конструкторского проектирования. Отражены технико-экономические стороны разработки и внедрения САПР.

Для инженерно-технических работников, занимающихся разработкой, внедрением и эксплуатацией САПР. Может быть полезна аспирантам и студентам старших курсов специализации «Конструирование и производство РЭА».

П 2401000000-002
046(01)-87 12-87

ББК 32.844

Рецензенты: д-р техн. наук Ю. Х. Вермишев и д-р техн. наук М. И. Песков

Редакция литературы по конструированию и технологии производства РЭА

Производственное издание

Годар Анатолевич Петухов
Григорий Григорьевич Смолич
Борис Илларионович Юлин

**АЛГОРИТМИЧЕСКИЕ МЕТОДЫ КОНСТРУКТОРСКОГО
ПРОЕКТИРОВАНИЯ УЗЛОВ С ПЕЧАТНЫМ МОНТАЖОМ**

Заведующий редакцией П. И. Никонов
Редакторы Н. К. Калинина, Л. Г. Селиванова
Художественный редактор Т. В. Бусарова
Обложка художника М. В. Бакурина
Технический редактор Э. Н. Ратникова
Корректор Т. В. Дземидович

ИБ № 1478

Сдано в набор 14.03.86. Подписано в печать 16.07.86. Т-07585 Формат 60×90^{1/16}
Бумага типогр. № 2 Гарнитура литературная Печать высокая
Усл. печ. л. 9,5 Усл. кр.-отт. 9,75 Уч.-изд. л. 10,90
Тираж 15000 экз. Изд. № 21766 Заказ 5166 Цена 55 к.

Издательство «Радио и связь». 101000 Москва, Почтамт, а/я 693

Ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО «Первая Образцовая типография имени А. А. Жданова» Союзполиграфпрома при Государственном комитете СССР по делам издательства, полиграфии и книжной торговли
118054, Москва, Валуевая, 28.

© Издательство «Радио и связь», 1987

ПРЕДИСЛОВИЕ

В центре практической деятельности нашей партии и народа в настоящее время находятся вопросы ускорения социально-экономического развития страны на основе научно-технического прогресса. Задача особой важности сегодня — наладить массовое изготовление техники новых поколений, способной дать многократное повышение производительности труда, открыть путь к автоматизации всех стадий производственного процесса.

Реализация мероприятий по созданию и использованию систем автоматизированного проектирования в народном хозяйстве дает возможность сократить сроки разработки, повысить качество и технологический уровень проектных и проектно-конструкторских работ, а также обеспечить значительную экономию используемых для этого материалов, энергетических и трудовых ресурсов, ускорить работы в области эффективного использования вычислительной техники и автоматизированных систем. Усложнение проектируемой электронной аппаратуры требует применения средств, позволяющих быстро перерабатывать огромное количество информации как исходной, так и получаемой в процессе проектирования.

Автоматизация процесса проектирования РЭА и создание систем автоматизированного проектирования (САПР) позволяют не только сократить сроки создания аппаратуры и облегчить условия труда, но и повысить качество производимой продукции. Это достигается тем, что при производстве изделий для управления прецизионными станками, автоматами и поточными линиями используются перфоленты и магнитные ленты, получаемые на ЭВМ в качестве выходной информации.

При производстве современного электронного оборудования постоянно возрастает процент применения печатных плат (ПП). При их конструировании необходимо решать задачи компоновки, размещения и трассировки.

Компоновка — распределение по корпусам ИС базовых элементов (БЭ) при реализации логики, представленной функциональной схемой. К компоновке относится также выделение из больших схем подсхем.

Размещение — определение места для каждой ИС и дискретного элемента на данном типоразмере платы с учетом выдвигаемых критериев.

Трассировка — определение расположения печатных проводников, обеспечивающих физическую реализацию данной схемы на определенном числе слоев, на основании имеющихся координат установки корпусов на плате.

Все эти три задачи тесно связаны между собой, и в процессе конструирования опытные конструкторы нередко решают их совместно, т. е. одновременно располагают элементы на плате и прокладывают проводники, меняя при этом местами уже установленные корпуса, делая перекомпоновку модулей. В связи с большими трудностями алгоритмизации процесса одновременного решения этих трех задач в большинстве известных САПР эти задачи решаются последовательно.

Актуальность продолжения исследований в данной области подчеркивается в работах многих ведущих ученых нашей страны. Одним из самых сложных вопросов в области алгоритмизации остается трассировка, но необходимо также разрабатывать новые методы и алгоритмы компоновки и размещения и усовершенствовать имеющиеся. Необходимо искать и новые связи между этими тремя основными этапами монтажно-коммутационного проектирования. В существующих САПР слабо используются соотношения эквивалентности.

Применение все более быстродействующих ЭВМ позволяет строить и применять более сложные алгоритмы проектирования, совершенствовать имеющиеся методы. Одной из первоочередных задач является повышение плотности печатного монтажа, что особенно важно при проектировании двусторонних плат, имеющих ограниченные ресурсы для прокладки трасс [19].

В книге сделана попытка отразить личный опыт авторов в решении указанных выше проблем. При описании математического аппарата авторы сочли нужным привести только те математические теории, которые были отражены в литературе недостаточно.

Читателю, хорошо знакомому с теорией проектирования с помощью ЭВМ, с первой главой достаточно ознакомиться бегло. Чтение второй главы необходимо большинству читателей, так как здесь освещены вопросы, выходящие за пределы общих курсов технических вузов. Для дальнейшего понимания содержания книги предполагаются хорошие знания дискретной математики в объеме обязательных курсов технических вузов. Третья глава содержит новые результаты авторов, поэтому знание ее для дальнейшего понимания необходимо. В последующих трех главах рассмотрены три основных этапа автоматизированного проектирования ПП, поэтому желательно читать их последовательно, хотя эрудированный в этой области читатель может знакомиться с содержанием глав выборочно.

После прочтения книги у читателя может сложиться мнение, что разработанные методы, модели и алгоритмы применимы только при проектировании ПП. Авторами сделан акцент на автоматизированное проектирование ПП, так как предлагаемые алгоритмы программно реализованы и проверены именно на проектировании ПП. Тем не менее большинство предлагаемых методов и алгоритмов могут быть успешно применены при конструировании электронной аппаратуры на различных уровнях, например при проектировании матричных БИС.

Глава 1

КРАТКИЙ ОБЗОР СОСТОЯНИЯ АВТОМАТИЗАЦИИ КОНСТРУИРОВАНИЯ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ

1.1. ОСНОВНЫЕ ТЕНДЕНЦИИ ПРИМЕНЕНИЯ ЭВМ ПРИ КОНСТРУИРОВАНИИ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ

В настоящее время ЭВМ широко и успешно внедряется во все этапы разработки радиоэлектронной аппаратуры (РЭА), включая и ее конструирование.

По степени общения с ЭВМ и по соотношению выполняемых задач принято разделять проектирование на автоматизированное и автоматическое.

Простейший вариант автоматизированного проектирования подразумевает выполнение основных этапов разработки аппаратуры ручным способом. Так, при конструировании печатных плат (ПП) размещение элементов на плате и трассировка проводятся конструкторами без применения ЭВМ. В отдельных таких системах ЭВМ используется для получения предварительных результатов на некоторых этапах. В дальнейшем результаты неавтоматизированного проектирования вводятся в ЭВМ, где подлежат дальнейшей обработке. На ЭВМ при этом возлагаются задачи контроля результатов на соответствие технологическим требованиям, получения конструкторско-технологической документации и ряд других функций. Недостатком такого подхода является большой объем рутинных работ, выполняемых конструктором.

Наибольшее распространение получили такие системы автоматизированного проектирования, в которых на высоком уровне организован активный диалог человека и ЭВМ. В процессе проектирования таким способом человек может вносить коррективы как в промежуточные, так и в конечные результаты.

Автоматическое проектирование не предусматривает вмешательства человека в процесс конструирования. При таком подходе после получения положительных результатов контроля исходной информации ЭВМ самостоятельно выполняет все этапы, оставляя пользователю только транспортировку носителей информации с управляющими программами для получения конструкторско-тех-

нологической документации и программ для станков с числовым программным управлением (ЧПУ).

Большинство работающих САПР реализовано на крупных ЭВМ (в качестве базовых); очевидно, эта тенденция сохранится и в будущем [1].

История развития систем автоматизированного проектирования ПП насчитывает уже более двадцати лет. За это время накоплен большой опыт разработки и эксплуатации этих систем. Данные о многих системах можно найти в монографиях [2, 3], а также в сообщениях [4—9]. В основу большинства этих систем положена автоматизация двух основных этапов конструкторского проектирования ПП: размещения и трассировки. Гораздо реже можно встретить в системах решение задачи компоновки.

Создаваемые в настоящее время САПР следует отнести к вычислительным системам, где оптимальным образом должны сочетаться широкое использование различного оборудования и программного обеспечения [10]. При создании таких вычислительных систем выдвигается ряд требований, одно из которых в литературе известно как интеграция по глубине [11], под которой понимается охват всех этапов проектирования — от ввода описания проектируемого объекта до получения проектной документации. Если раньше САПР охватывали в своем большинстве только этапы размещения и трассировки как наиболее трудоемкие, то в настоящее время обеспечивается получение фотошаблонов аппаратурой, непосредственно состыкованной с ЭВМ либо выдается управляющая перфолента (магнитная лента) для координатографов, различных станков с ЧПУ, тестовых автоматов, частично автоматизируется выпуск конструкторской документации [12]. Для расширения функций САПР в нее включаются различные задачи компоновки, автоматизация межблочного монтажа и т. п.

С точки зрения интеграции по глубине рассмотрим вопрос включения в САПР подсистемы контроля используемых радиоизделий [13].

Для увеличения номенклатуры конструкторской документации, выпускаемой автоматически с помощью САПР, выдаются спецификации, ведомости покупных изделий, перечни элементов. Анализ разработок показывает, что в этих документах содержится до 30 % ошибочных записей: синтаксические ошибки, указание на использование устаревших изделий или изделий несуществующих номиналов, использование в разработке неразрешенных изделий. Более 90 % таких ошибок либо не влияют на изготовление аппаратуры, либо устраняются нормоконтролерами или конструкторами. Однако даже тот небольшой процент ошибок, достигающий до производства, значительно замедляет изготовление аппаратуры, увеличивает ее стоимость.

В упоминаемой подсистеме контроля выделяются два рода ошибок: одни могут быть исправлены без вмешательства человека, например некоторые ошибки в написании ГОСТа или ТУ; для исправления других привлекается пользователь. В этом случае подсисте-

ма указывает местонахождение ошибки, ее вид, в отдельных случаях рекомендуемые исправления, например по использованию наиболее близкого номинала. Включение такой подсистемы контроля позволяет как получить более достоверную документацию, так и ускорить цикл производства аппаратуры за счет сокращения дорогостоящих переделок, требующих большой затраты времени.

Другим требованием, предъявляемым к современным системам, является интеграция по ширине [11], которая заключается в наличии на отдельных этапах проектирования альтернативных алгоритмов и программ с оценочными критериями, позволяющими выбрать комплексы программ применительно к конкретному проектируемому объекту или к их классам.

Примером здесь может служить этап трассировки, когда с целью экономии машинного времени для цепей, содержащих малое количество контактов, иногда применяют эвристические алгоритмы, например алгоритм малоповоротного пути [14], а для цепей с большим числом контактов — одну из модификаций алгоритма Ли.

На этапе компоновки, в зависимости от класса проектируемого изделия (макет, образец, серийное производство), число используемых корпусов может быть различным. При макетировании аппаратуры допускается оставление незадействованных базовых элементов (БЭ) в корпусе, так как в процессе настройки и доработки аппаратуры возможны изменения в схеме. К серийно выпускаемой аппаратуре предъявляется требование минимизации стоимости, и, как следствие, максимального заполнения корпусов, т. е. использование всех БЭ в корпусе.

Начало интеграции по ширине было положено в первой очереди АСП-5 [6]. В этой очереди имеются три подсистемы, используемые для различных классов проектируемых плат: АСП-51 — пакетная полуавтоматическая система проектирования и контроля топологии, изготовления КД, фотошаблонов для любых изделий, эскизное проектирование которых получено вручную; АСП-2 — автоматическая система проектирования топологии изделий, имеющих двуслойную или попарно многослойную коммутацию; АСП-53 — автоматическая система проектирования топологии изделий, имеющих многослойную коммутацию со сквозными металлизированными отверстиями для переходов [104]. Указанные системы при проектировании используют единую базу данных, в которой находятся информационные и архивные данные для всех систем.

Все более возрастающий процент периферийного оборудования в общей стоимости ЭВМ, расширение функций этого оборудования, а также достижения в области искусственного интеллекта непрерывно повышают роль диалога человек-машина, который позволяет сочетать опыт и интуицию пользователя с быстродействием ЭВМ [15]. Диалог начинается с ввода первичной информации о проектируемом изделии, что пока остается слабым местом в существующих САПР. Имеющиеся средства ввода связаны с кодированием схемы оператором, что приводит к ошибкам кодирования из-за утомляемости оператора и удлиняет цикл проектирования.

Применение различных кодировщиков с одновременной подсветкой на дисплей результатов ввода позволяет оперативно вносить исправления, но не устраняет полностью ошибок ввода. Попытка автоматизации ввода электрической схемы с листа хотя и дает некоторые положительные результаты [16], однако пока представляет чисто академический интерес, так как необходимо для ввода стилизовать схему с последующим распознаванием ее на ЭВМ. На возможность автоматического ввода электрических схем указывается также в работе [17].

На следующих этапах проектирования интерактивный режим позволяет пользователю вмешиваться в процесс проектирования, используя свой опыт и интуицию, сочетая их с быстродействием ЭВМ, с достоверностью получаемых результатов. Однако при этом получают дополнительные трудности, связанные с необходимостью приобретения соответствующего оборудования. Так, из-за его отсутствия в ряде систем до сих пор для ввода корректирующей информации используются перфокарты, как это было в системах, реализованных на ЭВМ 2-го поколения. Такое положение не обеспечивает оперативного принятия решения и, следовательно, неприемлемо. С другой стороны, применение дефицитной или узкоспециализированной аппаратуры не позволяет тиражировать разрабатываемые системы, как это получилось с системой ИСАП [7], реализованной на ЭВМ М-4030, которая в настоящее время снята с производства. В ближайшее время следует считать перспективным совместное использование больших ЭВМ для решения трудоемких задач с комплексами АРМ.

Заслуживают внимания такие направления, как совершенствование имеющихся методов, алгоритмов и подходов всех этапов проектирования ПП, а также разработка новых [18, 19]. Одной из сложнейших задач при этом является нахождение методов совместного решения основных этапов конструкторского проектирования, что отличает работу опытного конструктора от работы существующих САПР. Совершенствование имеющихся методов и алгоритмов следует вести в направлении разработки согласованных критериев основных этапов, разработки локальных оптимальных алгоритмов и критериев. Большие возможности повышения эффективности существующих систем лежат в использовании соотношений эквивалентности (инвариантности) как отдельных БЭ, так и их входов [18], однако в решении этого вопроса разработчик сталкивается с большими трудностями алгоритмизации. Следует обратить внимание и на развитие параллельных методов трассировки, например [20], ибо это также один из характерных приемов работы опытного конструктора.

Характерной чертой существующих и эксплуатируемых современных САПР является наличие базы данных. Хорошо организованная база данных позволяет сократить затраты на подготовку и контроль исходной информации, сокращая, таким образом, и время проектирования изделия. Отличительной особенностью САПР следующего поколения будет база знаний (БЗ). На рис. 1.1 приве-

дена структурная схема такой системы. Рассмотрим функции отдельных блоков.

Блок прогнозирования определяет возможность получения полжительного решения по имеющейся информации о проектируемом изделии, экономические показатели проектируемого изделия, необходимый набор изделий более низкого уровня, используемый для проектирования аппаратуры. В диалоговом режиме определяется наиболее рациональная форма проектирования (например, двух- или многослойная ПП), метод проектирования (например, автоматический, автоматизированный), учитываются пожелания о форме и видах выдаваемой впоследствии сопровождающей документации. Решаются также другие подобные задачи.



Рис. 1.1

Блок обоснования принимаемых автоматически решений сообщает о точках ветвления процесса проектирования и о принятых бескомпромиссно решениях; о параметрах, повлиявших на выбор того или иного решения. Такая информация необходима пользователю для анализа хода проектирования, для введения возможных изменений при принятии решений.

Отличительной чертой диалога будущих систем должна быть взаимная активность. При этом подразумевается, что блок взаимно активного диалога «помогает» пользователю при возникновении сложной ситуации, когда необходимо принять решение и ЭВМ выступает в роли консультанта и эксперта одновременно, предлагая ему альтернативные варианты вместе с их оценкой. При необходимости пользователь может запросить или ввести дополнительную информацию, необходимую для выработки стратегии принятия решения. Может возникнуть необходимость повторения предыдущих этапов проектирования с учетом вновь полученных факторов или с учетом дополнительного введенной информации.

Блок пополнения базы знаний на основании полученного спроектированного изделия проводит анализ состояния базы знаний, оценивает правильность работы блока прогнозирования и вносит соответствующие коррективы. База знаний может пополняться и

автономно, например при начале проектирования новых типов изделий, не имеющих ранее аналогов.

Приведенная выше схема — укрупненная, отдельные этапы не нашли в ней отражения. Очевидно, что такой вопрос, как постоянный контроль обрабатываемой информации, не исключается из рассмотрения, как и многие другие традиционные вопросы.

Заметим также, что расположение блоков на схеме последовательно-условное. Так, блок диалога работает непрерывно от начала процесса проектирования и до окончания. Аналогично и блок обоснования принятого решения (см. приложения 1 и 2).

1.2. РАЗВИТИЕ АЛГОРИТМОВ КОМПОНОВКИ

Задача компоновки может решаться как «сверху вниз»; так и «снизу вверх». Первый вариант принято называть разрезанием, второй — покрытием. В некоторых работах, например [21], из задачи разрезания выделяют отдельно типизацию.

Одним из основных критериев при решении *задачи разрезания* функциональной или принципиальной схемы устройства на части является минимум числа связей между частями. Точное ее решение для большого числа элементов в схеме получается за практически неприемлемое время [22], поэтому разрабатываются различные эвристические алгоритмы. В задаче типизации ставится дополнительно вопрос выделения изоморфных подграфов [23]. Необходимость его решения диктуется требованием сокращения номенклатуры проектируемых изделий более низкого ранга, а следовательно, и удешевления стоимости разработки. Трудности решения данной задачи заключаются в том, что необходимо выделить не просто изоморфные подграфы, а учесть при этом и «качество» вершин, т. е. соблюдать строгое соответствие вершин их функциональной значимости, учитывая при этом не просто связи между вершинами, а связи между конкретными номерами выводов интерпретируемого вершинной элемента.

Основным критерием при решении *задачи покрытия* являются минимум числа модулей, необходимых для покрытия исходной схемы; минимум числа межмодульных связей; использование строго допустимого набора типов ИС и др. Сначала определяется оптимальный набор типовых модулей [24]. Для получения точного решения в задаче покрытия, как и в задаче разрезания, необходим большой объем вычислений [25, 26]. Автор этих работ предлагает точный алгоритм решения задачи компоновки по критерию максимальной связности для схем из одинаковых БЭ, но здесь же указывает, что для решения уже при $n=16$ требуется проводить большое количество вычислений, несмотря на сокращение перебора за счет применения метода ветвей и границ.

В [27] приводится разработанный авторами алгоритм, дающий неплохие результаты, однако при этом надо заметить, что он ориентирован на конкретные серии ИС, что требует коррекции и настройки системы компоновки при применении других серий.

Большое значение при решении задачи компоновки имеют математические модели описания схемы устройства. Ранее в большинстве работ для решения задачи компоновки электрическая схема устройства описывалась графом, мультиграфом или псевдографом [24, 28, 28]. Такое описание имеет ряд недостатков. При описании схемы устройства простым графом не возникает трудностей в процессе решения только в тех случаях, когда цепь соединяет между собой только два контакта. Если два элемента могут быть соединены более чем одной связью, то удобнее использовать при описании мультиграф. Более полно из семейства простых графов представляет схему устройства псевдограф, отдельные ребра которого начинаются и кончаются в одной вершине, т. е. образуют петли. Представление схемы устройства псевдографом удобно при решении задачи компоновки, когда одним из требований выдвигается получение при отображении схемы максимального числа связей в пределах одного корпуса или в пределах очередной компоновочной единицы.

Недостатком описания схемы простыми графовыми моделями являются трудности, возникающие при отображении цепей, связывающих между собой более двух контактов, т. е. при отображении многоэлементной (многоконечной) цепи. Для выхода из этих ситуаций каждую цепь, содержащую более двух контактов, описывают полным графом. При переходе от схемы к графу за счет развязки узлов (узел соответствует соединению всех элементов между собой) в графе имеются лишние ребра, т. е. соединения, фактически не существующие на электрической схеме. Это вносит избыточность в информацию и в дальнейшем в ряде случаев приводит к худшим результатам [30].

Другой метод представления многоэлементной цепи основан на представлении в виде вершин графа как самих элементов, так и их ребер. В результате такого представления получается двудольный граф (биграф), так как его вершины можно разделить на два непересекающиеся и несоединяющиеся в своих пределах подмножества вершин: подмножество, соответствующее элементам, и подмножество, соответствующее связям.

Преимущество представления схемы устройства простыми графовыми моделями заключается в том, что большинство теоретических результатов, полученных в области графов, приложимо только к регулярным графам, т. е. к графам, в которых ребро соединяет между собой только два узла, но при этом модификации отображения многоэлементной цепи графом позволяют в большинстве случаев строить только процедуры приближенного решения [31].

Более точно электрическая схема устройства описывается гиперграфом [30, 32]. Коренным отличием гиперграфа от графов, описанных выше, является то, что его ребрами могут служить произвольные, а не только двухвершинные и одновершинные подмножества заданного множества. Это отличие избавляет модели от указанных выше недостатков.

Решение задачи покрытия реализовано далеко не во всех САПР,

однако разработчики, автоматизировавшие этот этап, отмечают эффективность подобных систем, возможность создания преемственных критериев основных этапов проектирования ПП и таких алгоритмов, предшествующих трассировке, когда последняя из центральной задачи переходит просто в завершающий конструкторский этап [33].

Действительно, включение задачи компоновки в САПР позволяет не только расширить функции системы и увеличить объем получаемой с помощью ЭВМ конструкторской документации, но и открывает новые возможности для дальнейшей работы по улучшению качества автоматизированного проектирования. Дело в том, что необходимая на этапе компоновки исходная информация достаточна и для использования эквивалентности БЭ и их входов, что позволяет значительно улучшить качество проектирования. При этом следует заметить, что указанная эквивалентность широко используется опытными конструкторами, так как человек в данной ситуации использует свой опыт и интуицию, позволяющие ему на каждом шагу выбрать различные стратегии дальнейшего решения задачи, корректировать их и т. д., что трудно поддается формальному описанию [34].

Отдельного внимания заслуживают работы [35, 36], где задачу компоновки решают параллельно с задачей размещения. Основная идея таких подходов заключается в размещении пустых корпусов на плате, т. е. в выделении для них посадочных мест с дальнейшим распределением БЭ по закрепленным корпусам с учетом критериев, выдвигаемых как на этапе компоновки, так и на этапе размещения.

Разнообразие и противоречивость отдельных требований, возникающих при решении различных задач компоновки, вызывают некоторые затруднения в создании единой методики компоновки, хотя уже есть соображения о включении в единую систему задачи разбиения функциональной схемы на типовые элементы замены и покрытия их заданным набором модулей с одновременной установкой на плате по критерию линеаризации [37].

1.3 НЕКОТОРЫЕ ВОЗМОЖНОСТИ ОПТИМИЗАЦИИ АЛГОРИТМОВ РАЗМЕЩЕНИЯ

Задача размещения элементов на плате следует непосредственно за задачей компоновки. В ряде упоминаемых выше систем процесс автоматизированного проектирования начинается сразу с размещения. Основной целью при решении задачи размещения ставится предоставление «лучших» условий для трассировки, соблюдая при этом технологические ограничения.

Полный перебор, как единственный известный в настоящее время метод получения точного решения, здесь неприемлем, так как при размещении, например, 50 одинаковых элементов на 50 равнозначных позициях требуется около 3×10^{64} перестановок [38], откуда следует, что необходимо разрабатывать эвристические алго-

ритмы. Существуют различные методы решения задачи размещения, большинство из которых проанализировано в различных монографиях, например [2, 39], поэтому остановимся только на некоторых методах размещения. При решении задачи размещения целесообразно рассматривать отдельно размещение разногабаритных элементов и элементов одинаковых установочных или одинаковых приведенных размеров.

Первоначально задача размещения разногабаритных элементов была связана только с применением дискретных электро-радиоизделий (ЭРИ), а в настоящее время она приобрела еще и новый интерес в связи с применением на одной плате ИС, сильно отличающихся по размерам друг от друга, с применением при проектировании микропроцессоров на плате, а также с проектированием БИС. Продолжают вестись разработки, где наряду с применением ИС в электрических схемах встречается много дискретных ЭРИ, например в широкополосной аппаратуре связи.

Некоторые авторы считают, что для установки разногабаритных элементов наиболее перспективными являются алгоритмы последовательного размещения [40]. В этих алгоритмах сначала определяется порядок установки элементов на плату. При определении порядка установки элементов с малым числом выводов — до четырех — целесообразно учитывать, кроме связей с уже установленными элементами, в первую очередь связи с последними размещенными элементами, как это сделано в работе [41]; тогда при дальнейшем размещении с помощью ЭВМ получаются результаты, близкие к результатам конструктора [42]. В некоторых алгоритмах на определение очередного претендента на установку влияет не только количество размещенных связей, но и расположение свободных позиций для размещения элемента [43]. Представляет интерес работа [44], в которой предлагается при последовательном размещении сначала определять перспективную позицию для установки очередного элемента, а затем выбирать конкретного претендента на эту позицию.

Известен подход, когда из схемы выделяют группы из нескольких связанных между собой элементов, а затем группами же и устанавливают их на плату. Площадь установки при этом разрезают на части [45]. Ввиду оригинальности такого подхода мы еще остановимся на нем в одной из последующих глав.

Улучшение результатов при плотном размещении разногабаритных элементов путем перестановок малоэффективно и затруднительно, поэтому для улучшения размещения методом перестановок на конфигурацию элементов дополнительно накладывают ограничения. Так, в работе [46] рассматриваются элементы, имеющие равную ширину, а на высоту ограничения практически не накладываются.

При размещении дискретных ЭРИ за счет аппроксимирующих прямоугольников отношение плотности установки элементов к плотности печатного монтажа при автоматизированном проектировании ниже, чем при неавтоматизированном. Для повышения

плотности установки автоматическими методами иногда предлагается отойти от принципа последовательного наращивания, т. е. от последовательного расширения области установки, а вести размещение одновременно с четырех углов. Теоретически повышение плотности установки при таком подходе не обосновано. Рассмотрим некоторые алгоритмы размещения элементов равных размеров.

В работе [47] предлагается достаточно простой критерий оптимизации размещения. На плате с регулярным размещением модулей между рядами и колонками корпусов проводят воображаемые линии. При определении места модуля проводят прямолинейные отрезки между контактами согласно списку цепей. В зависимости от конфигурации платы минимизируется число пересечений вертикальных или горизонтальных воображаемых линий с этими прямолинейными отрезками. Автор указывает, что получаемые по такому критерию результаты не являются оптимальными, но тем не менее достаточно приемлемыми для практических целей. Критерий линеаризации для сети соединений можно найти и в работе [48].

В работе [49] ставится задача получения минимального числа внешних связей по рядам модулей. Автор указывает, что лучшие результаты предлагаемый алгоритм дает при небольшом числе связей, выходящих на соединитель.

Представляет интерес метод последовательного деления платы на части при установке модулей для получения минимума связей между частями [50]. Алгоритм дает неплохие результаты, однако его недостатком является определение частей платы до определения связности элементов, что может привести к разбиению сильно связанных между собой элементов и расположению их в разных частях платы.

Продолжают разрабатываться алгоритмы последовательного размещения и для установки элементов равных размеров. В работе [51] после определения последовательности установки модулей предлагается при оценке очередного места установки модуля использовать метод ветвей и границ, сокращая тем самым число просматриваемых посадочных мест.

Недостатками имеющихся алгоритмов последовательного размещения являются: стягивание первых устанавливаемых элементов к одному месту на плате, что ведет к неравномерному распределению прокладываемых трасс, и малая гибкость процесса установки: последующие устанавливаемые элементы получают существенно худшие позиции для установки из-за жесткой закрепленности ранее размещенных элементов, поэтому необходимо разрабатывать математические модели посадочных мест, позволяющие избежать столь жесткого закрепления.

При проектировании плат с элементами равных размеров находят широкое применение алгоритмы, основанные на перестановках элементов. Итерационные алгоритмы хорошо проанализированы в указанных выше работах, поэтому заметим только, что хорошие результаты дают алгоритмы групповых перестановок

[52]. В этой же работе показано, что алгоритмы перестановок связанных множеств значительно быстрее и качественнее обеспечивают окончательное размещение, чем алгоритмы перестановок несвязанных множеств. Отсюда следует, что одним из направлений разработки алгоритмов размещения должно быть выделение максимально связанных элементов в группы с последующей установкой группами. Учитывая, что на качество размещения итерационных алгоритмов влияет также и начальное размещение, необходимо совершенствовать и последовательные алгоритмы установки элементов.

При реализации электрических схем в виде печатных плат иногда сталкиваются с проблемой установки на плату и ИС, и дискретных элементов. В некоторых системах установка дискретных ЭРИ делается вручную после установки ИС, в других — для этой цели используют фиктивный соединитель либо вводят псевдомодули. В работе [53] описан алгоритм автоматической установки навесных элементов после размещения ИС; ЭРИ устанавливаются в упоминаемом алгоритме на любые свободные места по выбранному критерию. Основным критерий установки — минимальная длина связывающих цепей.

Общим недостатком большинства известных критериев установки (минимальная суммарная длина связей, минимум пересечений, линеаризация соединений, минимизация полупериметра описывающего прямоугольника и т. п.) является то, что эти критерии при размещении не учитывают вообще либо слабо учитывают удлинение трасс, которое впоследствии получается при прокладке. В оптимизируемых функциях размещения вообще не учитывается число каналов, которые, возможно, потребуются для реализации связей, что снижает эффективность этих критериев.

1.4. ТЕНДЕНЦИИ РАЗВИТИЯ АЛГОРИТМОВ ТРАССИРОВКИ

Основой большинства разрабатываемых и эксплуатируемых алгоритмов трассировки является алгоритм Ли. Он работает по волновому принципу и соединяет две точки одним из кратчайших путей, обходя имеющиеся препятствия. По этому алгоритму всегда находится путь, если он только существует. Для сокращения времени счета, объема используемой памяти и повышения качества печатного монтажа разрабатывались и разрабатываются различные модификации волнового алгоритма, например [54, 55]. В связи с большой универсальностью этот алгоритм изучается и совершенствуется и в настоящее время.

Интересная модификация волнового алгоритма описана в работе [56], где волна распространяется сразу от всех контактов цепи. Данный подход позволяет существенно сократить время счета, так как много времени обычно затрачивается на проверку возможности проведения нереализуемого пути ввиду того, что в зависимости от назначения источника и цели волной может про-

веряться практически вся свободная площадь платы. При использовании предлагаемого подхода определение изолированных областей происходит быстрее, что сокращает время поиска возможности проведения. Дальнейшее развитие идей и результаты реализации нашли отражение в работе [57].

Результаты работы волнового алгоритма сильно зависят от последовательности проведения соединений. Наиболее распространенный порядок проведения — по возрастанию длины соединений. В работе [59] описан метод определения очередности проведения трасс в процессе трассировки, в котором выдвигается расплывчатое требование: взять ту трассу, которая «меньше всего мешает остальным». В критерии используется кратчайшая длина соединений и число пересечений с другими трассами. Известны принципы определения порядка трассировки цепей в зависимости от плотности распределения контактов трассируемой цепи на плате и от соотношения длин сторон минимального описывающего прямоугольника.

Общим недостатком волновых алгоритмов является пошаговая локальная оптимизация. Проводимые по таким алгоритмам трассы не учитывают или слабо учитывают потребности последующих прокладываемых трасс. Для устранения этого недостатка разрабатываются методики предварительного распределения трасс в каналах с последующей итерационной трассировкой в каналах [60]. В этой же работе для корректировки рисунка используется прием отодвижения проложенных ранее трасс, названный автором методом «скользящей горки». Здесь же указывается на использование при трассировке эквивалентности контактов платы.

Использование топологической модели платы при реализации волнового алгоритма позволило авторам работы [61] существенно сократить число неразведенных фрагментов. Окончательное расположение трасс в предлагаемом алгоритме производится после предварительного проведения всех цепей.

Топологические методы позволяют повысить плотность печатного монтажа. В этом плане следует отметить результаты «гибкой» трассировки [20]. Пары точек соединяются кратчайшим путем, как и в классическом алгоритме Ли. В процессе прокладки проводимому проводнику разрешается отодвигать в возможных случаях ранее проложенные трассы, поэтому увеличивается число автоматически проведенных соединений и повышается плотность печатного монтажа. Чтобы повысить плотность монтажа, в настоящее время в алгоритмах применяются приемы неавтоматизированного конструирования: смещение ранее проложенных трасс, перебор их через контакты [58, 62].

На основе идей алгоритма Ли построен и алгоритм канальной трассировки [63]. Метод предполагает трассировку на двух слоях, существенно сокращает время счета и объем используемой памяти.

Развитие идей параллельной трассировки находит свое отражение в предварительном распределении проводников в различ-

ных частях канала [64, 65]. Если в первой работе авторы выделяют середину канала и края, выталкивая наиболее конфликтующие трассы в середину, то во второй связи делятся на две группы, одна из которых должна трассироваться в верхней части канала, а другая — в нижней. Этими приемами авторы делают попытку предопределить прокладку последующих трасс.

Представляет интерес оптимизация расположения трасс в канале методом сечений, описанным в работе одного из основателей параллельной трассировки в нашей стране [58]. В основу метода сечений положена идея полного перебора всех возможных вариантов взаимного положения проводников в канале. При этом в рамках заданных ограничений (ортогональная форма проводников, пропускная способность канала ограничена числом 8) достигается нахождение точного решения задачи трассировки в канале. В [93] эта же идея распространена на области пересечения каналов, точная трассировка в которых, как известно, представляет наибольшую сложность.

Примеры предварительного распределения трасс по каналам можно найти в работе [66]. Недостатком данного подхода можно считать то, что цепи делятся перед трассировкой на пары контактов, т. е. многоконечные цепи при этом искусственно разделяются, что не позволяет вести общую оптимизацию связывающей сети. Для повышения плотности печатного монтажа применяются методы частичной перетрассировки.

В работе [67] описан один из лучевых алгоритмов, который применим при разводке непроведенных соединений методом перетрассировки. Алгоритм предназначен для трассировки многослойных печатных плат и соединяет только пары контактов.

Для увеличения процента реализованных автоматически трасс разрабатываются методы автоматической коррекции соединений. В работе [68] такой метод основан на удалении «мешающих» соединений, проведении не разведенной ранее цепи и попытки проведения удаленного соединения. Для проведения трасс с числом конфликтов более одного предлагается деформировать ранее проведенные трассы.

Более сложной проблемой является разработка приемственных критериев компоновки, размещения и трассировки [18]. В упоминаемой монографии авторы предлагают алгоритм совместного решения задач размещения и трассировки, что является значительным шагом вперед. Суть подхода заключается в размещении элементов в линейке с одновременным проведением трасс, последующим установлением связей между элементами и перетрассировкой цепей, содержащих более двух контактов. Эффективность данного подхода зависит от решения задачи компоновки элементов в линейке, где применяется последовательный алгоритм размещения, что несколько снижает эффективность подхода, так как к недостаткам алгоритмов последовательного размещения добавляется требование устанавливать выбранные элементы в один ряд. Из сказанного можно сделать вывод, что предлагае-

мый подход может дать хорошие результаты на определенном классе схем.

На разработку единого подхода к решению задачи размещения и трассировки указано и в работе [69], однако сжатое изложение материала не позволяет сделать выводов об эффективности и применимости предлагаемого метода.

1.5. НЕКОТОРЫЕ ВОЗМОЖНОСТИ РАСШИРЕНИЯ ФУНКЦИЙ САПР

Тенденции к обучению ЭВМ видеть и слышать ставят перед разработчиками САПР задачи по расширению возможностей систем проектирования. Одним из возможных направлений решений данной проблемы является ввод графической информации в ЭВМ. Переход к проектированию на ЭВМ 3-го поколения способствовал дальнейшему развитию периферийного оборудования и математического обеспечения для развития новых методов ввода данных в ЭВМ. Решение задачи автоматического ввода информации без предварительного кодирования позволяет существенно сократить цикл проектирования, так как в отдельных вычислительных системах время подготовки данных соизмеримо со временем обработки их на ЭВМ. Так, в САПР на ЭВМ 2-го поколения время подготовки исходных данных превосходит время проектирования платы на машине, в связи с чем возрастает интерес к проблеме подготовки исходной информации и ввода ее в ЭВМ.

Следует отметить, что среди ошибок входной информации, выявляемых синтаксическим контролем, ошибки разработчика, появляющиеся в электрических схемах, составляют около 20 % (частично они выявляются при кодировании схемы). Приблизительно 15 % ошибок приходится на этап перфорации, 5 % — на кодирование технической и конструкторской информации о плате, а остальные — на кодирование электрической схемы. Часть ошибок происходит из-за того, что данные готовят работники различных специальностей и из-за их утомляемости в связи с однообразием труда.

Существенная часть времени проектирования уходит на взаимодействие различных подразделений, занятых в процессе проектирования. Чтобы сократить время подготовки информации для ввода электрической схемы в ЭВМ, применяют различные полуавтоматические устройства планшетного типа. В этом случае распознавание условных образов делается специально обученными операторами. Это ускоряет процесс подготовки, позволяет избежать ряда ошибок, а также исключает этап перфорации и частичного контроля.

Для ввода с помощью дисплея электрическая схема предварительно вычерчивается на специальном бланке, который разбивается на несколько частей ввиду ограниченности информационного поля дисплея. Информация с частей бланка заносится на

экран с помощью светового пера с последующим вводом нанесенного изображения в ЭВМ. Время ввода электрической схемы объемом 20—25 микросхем — около одного часа [70].

Оба описанных способа ввода позволяют сократить время подготовки данных и сократить число ошибок, но обладают одним общим недостатком: время ввода электрической схемы с помощью как полуавтоматического устройства, так и дисплея ограничено скоростью работы оператора. В первом случае оператор должен либо передвинуть движок по всем общим контактам цепи, либо наводить следящее перекрестие; во втором случае он должен указать все общие точки световым пером. И то, и другое остается нетворческой и утомительной частью труда в процессе автоматизированного проектирования.

Исследования в области распознавания образов с помощью ЭВМ достигли значительных практических успехов. Однако к начертанию распознаваемых символов предъявляются жесткие требования. Сложность проблемы распознавания легко пояснить примером: первоклассник в конце года легко может узнавать различные начертания какой-либо буквы, допустим «а», а для машины все начертания на плоскости буквы «а», не совпадающие при простом переносе, т. е. без сжатия и растяжения по осям и без поворота, являются различными образами. Эти соображения заставляют создавать стилизованные модели электрических схем со всеми ее элементами. Естественно, что модель должна нести только необходимый минимум информации.

В предлагаемой ниже схеме ввода черно-белого изображения в ЭВМ уровень черного заносится в память в виде «1», а уровень белого — в виде «0». Задача теперь ставится так: по взаимному расположению нулей и единиц определить элементы схемы и связи между ними. Связность элементов схемы и отсутствие изолированных элементов являются особенностями, на которых строится распознавание.

К вводу информации предъявляются следующие требования: простота подготовки данных, высокая скорость ввода, надежность. Кроме этого система ввода должна предусматривать синтаксический контроль вводимой информации и возможность оперативного исправления ошибок.

Наиболее удачно, с точки зрения авторов, перечисленным выше требованиям удовлетворяет система, показанная на рис. 1.2. На рис. 1.3 приведена часть схемы, подготовленной к вводу в ЭВМ, а на рис. 1.4 — укрупненная блок-схема алгоритма распознавания. Контакты соединителя предлагается изображать прямоугольниками у лицевой линии (лицевая линия ограничивает соединитель схемы и является одной из четырех линий, ограничивающих электрическую схему, что необходимо, чтобы машина «знала» свои пределы действия).

Если контакт закреплен, т. е. к нему может быть подсоединена только определенная цепь, то он полностью затушевывается, в противном случае внутри остается белое пространство. Число

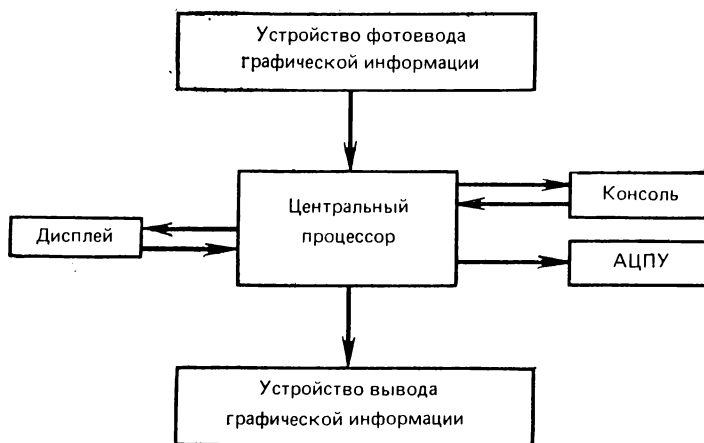


Рис. 1.2

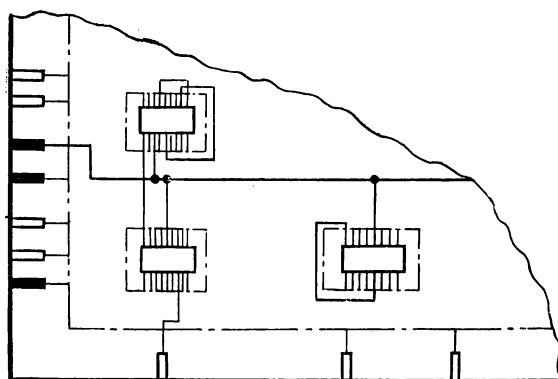


Рис. 1.3

изображаемых контактов соединителя $n-2$, так как два контакта отводятся под цепи питания. Дискретные элементы представляются как контакты верхнего и нижнего соединителей. Около соединителей и модулей оставляются запрещенные зоны, в которых не разрешается пересечение проводников. Такая зона необходима для облегчения определения выводов, их номеров и номеров модулей. Остановимся подробнее на некоторых блоках схемы, представленной на рис. 1.4.

При определении i -го задействованного контакта соединителя программа определяет задействованный контакт соединителя, запоминает координаты и номер контакта перед началом работы программы ЛИНИЯ. Это необходимо для продолжения счета после окончания распознавания очередной цепи. Номер контакта необходимо запоминать, так как в дальнейшем могут встретиться как незакрепленные, так и закрепленные контакты соединителя.

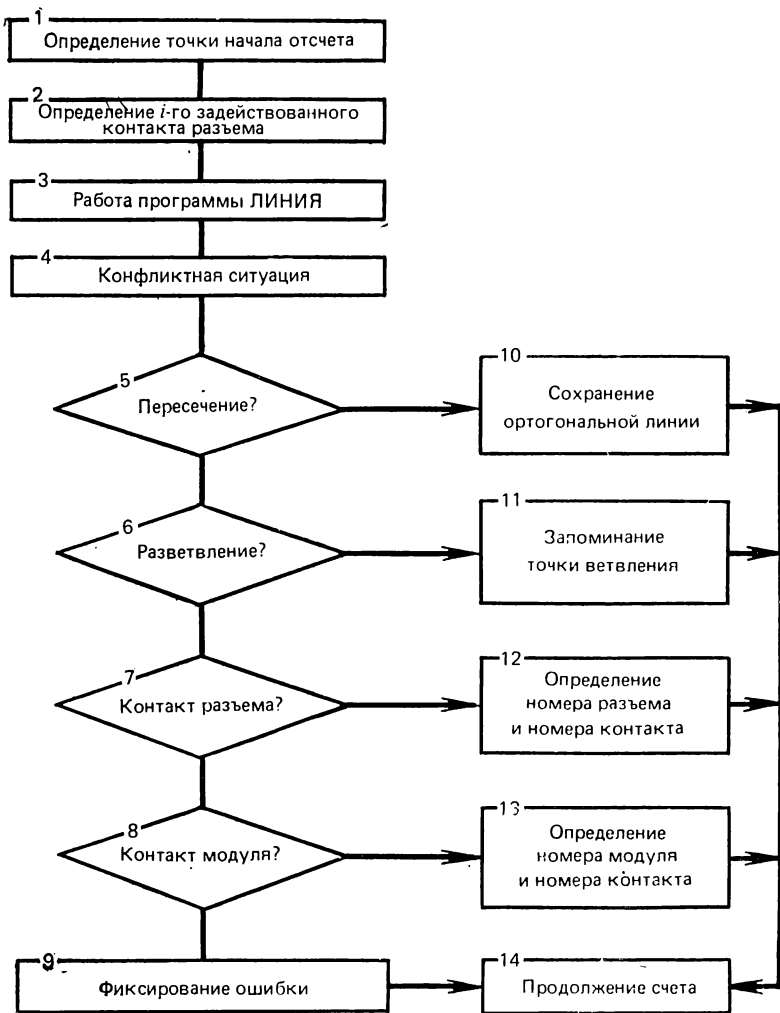


Рис. 1.4

Программа ЛИНИЯ предназначена для определения связей между элементами. Поиск ведется модификацией волнового алгоритма. Волна направленно распространяется по проводнику от задействованного контакта соединителя. При считании линии шириной 0,5 мм на рецепторном поле фронт волны получается равным 3 единицам (при разрешающей способности устройства 2,5 линии на миллиметр). Учитывая случайные утолщения линий, перекосы при вводе и кривизну линий допускаем фронт волны до 6 единиц.

Конфликтная ситуация характеризуется расширением либо исчезновением фронта волны. При расширении фронта методом зондирования определяется либо характер ветвления, либо номер

соединителя и его контакта, либо номер контакта модуля. Последние характеризуются конфликтной ситуацией в одной из запрещенных зон (модули на специальных бланках располагаются на закрепленных местах, таким образом, каждой позиции поставлен в соответствие определенный модуль). При считывании возможны разрывы линий, что также определяется зондированием, только на меньшей глубине. Чтобы эффективно использовать указанный метод, необходимо, чтобы расстояние между линиями было не менее чем в 5—6 раз больше ширины линии, т. е. около 3 мм. Волновой принцип распознавания допускает при вычерчивании любую кривизну линий, т. е. можно вводить эскизы, нарисованные от руки.

Для увеличения скорости распознавания в программе ЛИНИЯ имеется программа ЛИСА, в обязанности которой входит затирание пройденных линий (заметание следа), что сокращает число конфликтных ситуаций. При этом важно не затереть общие точки с пересекаемой линией (блок 10), а точки ветвления запомнить (блок 11), чтобы в дальнейшем продолжать счет (блок 14).

Для определения номера контакта модуля от всех выводов должны отойти маркирующие линии до границы запрещенной зоны. Пересчет числа маркирующих линий вправо или влево даст ответ на вопрос о номере контакта. Расстояние от границы запрещенной зоны до модуля должно быть не менее чем вдвое больше расстояния между маркирующими линиями, что необходимо для отличия вывода от проходящего мимо модуля проводника.

Из-за отдельных ошибок не следует снова вводить схему: результаты записываются на магнитную ленту, а ошибки подробно распечатываются на АЦПУ. Ошибки анализируются на рабочем столе, и подготавливается информация для коррекции. Коррекцию можно проводить с помощью перфокарт или дисплея, после чего задача передается на счет. Примером ошибки может быть повисание проводника.

Кроме ранее оговоренного продолжения счета от точки ветвления сюда же относится перебор контактов соединителя и контактов модуля. Этот перебор необходим для того, чтобы решение было замкнутым, т. е. чтобы не осталось нераспознанных цепей.

При вводе графической информации в ЭВМ с «листа» встает проблема сжатия информации. Этот вопрос заслуживает отдельных исследований. Можно предложить метод вычеркивания строк, не несущих новой информации: при вводе сравниваются предыдущая и настоящая строки (строка — поперечная линия по всей ширине вводимой схемы), и если они равны, то перенос настоящей строки в память не происходит. Этот метод позволяет сжимать информацию только по одной оси, однако можно после ввода провести сканирование в ортогональном направлении по такой же методике, что ускоряет процесс распознавания.

Опробование фотоввода проводилось с помощью фототелеграфной аппаратуры ФАК-ДМ. Использование аппаратуры с

большой скоростью ввода, например «Газета-2», позволяет сократить время ввода формата А1 до 2—3 минут, т. е. в 15—20 раз. Недостатком такого скоростного ввода является большой требуемый объем оперативной памяти.

К интеграции по глубине следует отнести и включение в САПР автоматизированной информационно-справочной системы по электрорадиоизделиям, включающей в себя и библиотеку элементов. Использование указанной подсистемы в САПР позволяет не только проверять комплектующие ЭРИ на соответствие ГОСТу или ТУ, но и полностью отказаться в задании на проектирование от указания ГОСТа или ТУ, получая эти данные из библиотеки.

Несмотря на отсутствие явно выраженного экономического эффекта таких подсистем, интерес к данному вопросу возрастает с возрастанием номенклатуры используемых при проектировании радиоизделий, так как при этом увеличивается и число ошибок разработчика при составлении подобной документации.

Трудности при создании и эксплуатации таких систем связаны с большим объемом обрабатываемой и хранимой информации, поэтому здесь остро стоит вопрос со сжатием последней. Другая проблема — постоянное обновление и пополнение базы данных большим количеством новых изделий.

Одной из актуальных проблем расширения функций САПР является вопрос выпуска конструкторской документации. При решении на ЭВМ только задачи размещения и трассировки на плечи конструктора ложится наиболее рутинная и нетворческая работа: вычерчивание сопровождающих чертежей, составление различных таблиц. Такие таблицы, как ведомости покупных изделий, спецификации, перечень элементов, могут быть получены на печатающем устройстве ЭВМ (хотя качество получаемых документов зачастую невысоко, но все же проблема успешно решается). Сложнее дело обстоит с чертежами. Сборочный чертеж и чертеж деталей платы на ряде предприятий уже получается с помощью различных графических устройств. Вычерчивание принципиальных электрических схем пока широкого промышленного распространения не получило. В решении этих вопросов намечаются два направления.

Первое основано на сколке эскиза схемы на кодировщике графической информации с последующей выдачей на графопостроитель. При этом изображение элементов схемы (ИС, резисторов, конденсаторов и т. п.) находится в базе данных, т. е. при сколке указываются только координаты местонахождения элемента на чертеже и его шифр. Таким образом, сокращается время кодирования схемы. При таком подходе элементы практически остаются на тех же местах, что и на эскизе, т. е. фактически происходит только аккуратное перечерчивание схемы, а также контроль правильности начертания.

Второе направление предполагает кодирование связей схемы и введение информации об используемых элементах, т. е. как и в системах проектирования ПП. При таком подходе проводится

автоматическое размещение и прокладка связей. Для сохранения большей наглядности и читаемости схемы в подобного рода подсистемах должно предусматриваться директивное размещение некоторых или всех элементов. Некоторые алгоритмы таких подсистем найдут отражение в соответствующих главах книги. Недостатком данных подсистем является ввод большого объема информации. Для первого подхода характерна большая наглядность кодируемой схемы и получаемых результатов, однако вводимая информация менее пригодна для последующего проектирования с помощью ЭВМ, т. е. такие системы целесообразно использовать автономно.

Другой подход позволяет использовать вводимую информацию для дальнейшего автоматизированного проектирования. Его преимуществом следует считать также возможность более оперативной корректировки схемы после доработки в результате возможности более простого хранения информации, что исключено в первом подходе.

Учитывая, что перечерчивание схемы средней сложности с эскиза ручным способом занимает от 5 до 10 дней, разработка и внедрение подсистем вычерчивания является актуальной проблемой для сокращения объема рутинных работ при конструировании аппаратуры. Вместе с тем, учитывая стоимость проектирования на ЭВМ, экономически обоснованным является автоматизированное вычерчивание только тех схем, которые проходят полный цикл проектирования с помощью САПР. Только при этом окупаются затраты на кодирование исходной информации и на затрачиваемое машинное время, а также на прочие расходы.

В заключение отметим, что расширение функций САПР в указанном выше плане позволяет сократить время разработки аппаратуры, уменьшить число ошибок проектирования, избавить проектировщика от большей части нетворческого труда.

Глава 2

МАТЕМАТИЧЕСКИЙ АППАРАТ ОПТИМИЗАЦИИ АЛГОРИТМОВ КОНСТРУИРОВАНИЯ

2.1. ОСНОВНЫЕ ПОНЯТИЯ ТЕОРИИ РАСПЛЫВЧАТЫХ МНОЖЕСТВ

Впервые понятие расплывчатых множеств было введено Лофти Заде в 1965 г. [71]. Учитывая, что в отечественной литературе, предназначенной для широкого круга читателей, данная теория не нашла достаточного отражения, приведем основные понятия

и результаты, которые либо будут нами использованы в дальнейшем, либо, по мнению авторов, могут найти применение при разработке освещаемых в данной работе вопросов.

В отечественной литературе терминология описываемой теории пока еще окончательно не утвердилась, поэтому в литературе можно встретить следующие термины: «размытые», «расплывчатые», «неопределенные», «нечеткие», являющиеся переводом английского слова *fuzzy*. Суть этих понятий одна.

Расплывчатые множества и построенные на их основе алгоритмы удобно использовать при предварительном исследовании проблем и задач для последующего более обоснованного принятия конечного решения, построения более эффективных алгоритмов и процедур. Теория расплывчатых множеств уже нашла широкое применение при распознавании рукописных знаков, при решении задач планирования и т. п.

В процессе выработки требований к разрабатываемому алгоритму мы часто выдвигаем нечеткие промежуточные требования. Так, при прокладке очередного проводника мы стараемся как можно меньше помешать последующим прокладываемым трассам, стараемся выбрать позицию для установки очередного элемента или выбрать элемент для установки на позицию с желанием как можно меньше «испортить» будущее для других, еще не установленных элементов. Такое нечеткое условие принятия решения приводит нас к теории расплывчатых множеств.

В чем же суть теории расплывчатости? Попробуем сначала объяснить ее на примере. Рассуждая о возрасте человека, мы говорим, что Иванов средних лет, Петров — молодой и т. п., однако строгих границ возраста определить не можем, а поэтому из-за отсутствия четких границ между возрастными определениями «молодой» ... «старый» мы часто затрудняемся отнести рассматриваемого индивидуума к той или иной возрастной группе.

Даже отойдя от большинства таких слов, как «большой», «значительно», «существенно», «лучший» и т. п., мы порой оказываемся во власти расплывчатых определений множества. Кажущееся точное определение не дает нам точных границ между мягким и твердым металлом, твердым и сверхтвердым металлом. Итак, интуитивное понятие расплывчатого множества — это множество, которое не имеет точных границ.

Дадим точное определение. Пусть у нас имеется множество Ω . Тогда подмножество A множества Ω называется расплывчатым, если для каждого $a \in A$ определена мера принадлежности $\mu(a)$, определенная на множестве всех элементов Ω . Мера принадлежности $\mu(\omega)$ принимает значение на отрезке $[0,1]$ при $\forall \omega$ для $\forall A$. Если $\mu(\omega)$ принимает только два значения: 0 и 1, то мы имеем дело с хорошо известной характеристической функцией $\chi(\omega)$ и границы множества A уже не размыты, т. е. мы имеем дело с обычным множеством.

Вернемся к одному из приведенных выше примеров. При определении меры принадлежности по отношению к выражению «мо-

лодой» ... «старый» в расплывчатом подмножестве «молодой» (M) начиная с некоторого момента кривая, определяющая меру принадлежности к подмножеству M , сначала будет возрастать, достигнет 1, затем будет убывать, а с некоторого момента будет снова равна 0. График функции представляет собой некий «горбик», ибо когда-то человек бывает юным, а следовательно, мера принадлежности в этот момент равна 0, когда-то $\mu(m)=1$, а в старости снова $\mu(m)=0$.

Приведем некоторые определения, которые в дальнейшем могут быть полезными. *Расплывчатое множество A пустое* тогда и только тогда, если $\mu_A(\omega)=0$ для всех $\omega \in \Omega$. *Расплывчатое множество A нормальное* тогда и только тогда, если $\sup_{\omega \in A} \mu(\omega) = 1$.

Если $\sup_{\omega} \mu_A(\omega) < 1$, то множество называется *субнормальным*.

Любое непустое субнормальное множество нормализуется делением меры принадлежности для каждого элемента на супремум. Понятие нормальности имеет важное значение при выборе и принятии решений. Если расплывчатое множество субнормально, то при $\mu(\omega) < 1/2$ трудно определить приоритетную принадлежность рассматриваемого элемента к одному из альтернативных множеств.

Множество A' называется дополнением расплывчатого множества A тогда и только тогда, если $\mu_{A'}(\omega) = 1 - \mu_A(\omega)$. Например, пусть имеется множество пар точек на одной прямой, которые необходимо соединить либо выше, либо ниже этой прямой. Любую из рассматриваемых пар разрешается соединять и ниже, и выше прямой. Если возможности соединения ограничены, то сначала желательно получить предварительное распределение предположительных соединений. По тем или иным критериям в этом случае каждой паре удобно поставить в соответствие меру принадлежности к множеству соединений выше прямой $\mu_B(x_i, x_j)$ и ниже прямой $\mu_H(x_i, x_j)$. Тогда, обозначив множество возможных соединений над прямой через A , получим множество возможных соединений ниже прямой A' ; в этом случае имеем: $\mu_B(x_i, x_j) = 1 - \mu_H(x_i, x_j)$. Очевидно, что дополнение расплывчатого множества всегда есть расплывчатое множество.

Определим операцию *выключения*. Будем говорить, что A содержится в B и обозначать $A \subset B$ (или A является подмножеством B , иногда говорят, что A меньше или равно B) тогда и только тогда, если $\mu_A(\omega) \leq \mu_B(\omega)$.

Под *пересечением* расплывчатых множеств A и B понимают наибольшее расплывчатое множество $C = A \cap B$, содержащееся и в A и в B одновременно и имеющее меру принадлежности $\mu_C(\omega) = \min \{ \mu_A(\omega), \mu_B(\omega) \}$, где $\omega \in \Omega$.

Например, если Иванов с мерой $\mu = 3/4$ принадлежит к множеству светловолосых людей и с мерой $\mu = 1/2$ принадлежит к множеству высоких людей, то мера принадлежности его к множеству высоких светловолосых людей будет равна $\mu = 1/2$. На

рис. 2.1, показан пример определения меры принадлежности пересечения двух множеств.

Под *объединением* расплывчатых множеств A и B понимается наибольшее расплывчатое множество $C=A \cup B$, содержащее как A , так и B одновременно; мера принадлежности определяется по формуле $\mu_C(\omega) = \max\{\mu_A(\omega), \mu_B(\omega)\}$, где $\omega \in \Omega$. В приведенном выше примере Иванов «получает» меру $\mu=3/4$ при определении его принадлежности к множеству, объединяющему и высоких и светловолосых людей.

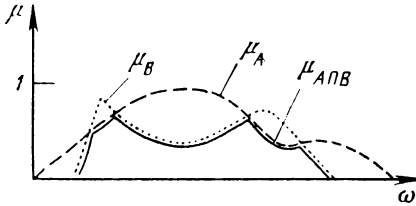


Рис. 2.1

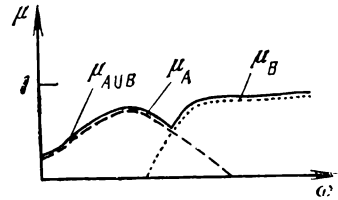


Рис. 2.2

Объединение расплывчатого множества с обычным множеством может быть как расплывчатым (рис. 2.2), так и обычным множеством. Обозначив через $\chi(\omega)$ характеристическую функцию обычного множества, получим правило: если мера принадлежности $\mu_A(\omega)$ пересечения расплывчатого множества с обычным совпадает с характеристической функцией обычного множества, то объединение будет обычным множеством: $\mu_{A \cup B}(\omega) = \max\{\mu_A(\omega), \chi_B(\omega)\} = \chi_{A \cup B}(\omega)$ (рис. 2.3), если пересечение не пусто и такого совпадения нет, то получается расплывчатое множество. Более того, объединение двух (возможно, и более) расплывчатых множеств может быть обычным множеством, если функция принадлежности объединения вырождается в характеристическую функцию:

$$\mu_{A \cup B}(\omega) = \max\{\mu_A(\omega), \mu_B(\omega)\} = \chi_{A \cup B}(\omega) \quad (\text{рис. 2.4}).$$

При поверхностном знакомстве с теорией расплывчатых множеств у читателя может возникнуть подозрение, что мера принадлежности является функцией распределения, если Ω счетное, или плотностью вероятности, если Ω — континуум. Внимательное рассмотрение приведенных выше определений теории расплывчатых множеств показывают коренные отличия меры принадлежности от вероятности. Действительно, понятие расплывчатого множества не является случайным по своей природе.

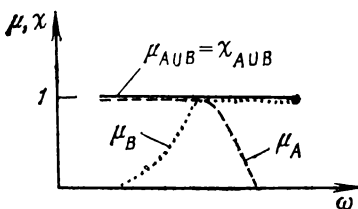


Рис. 2.3

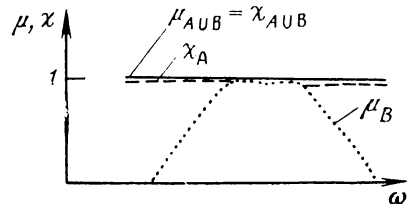


Рис. 2.4

Сходством, на основании которого, вероятно, появляются эти подозрения, является то, что как мера принадлежности, так и вероятность события изменяются в интервале $[0,1]$. И то, и другое понятия сообщают нам о причастности некоторого события к какому-нибудь множеству. Но вероятность говорит нам о возможности появления или свершения рассматриваемого события, в то время как мера принадлежности говорит нам о том, что данное событие имеет место, и сообщает нам о множестве, в котором это событие имеет место. Кроме того, понятие вероятности требует четких границ множества объектов, что является коренным отличием от понятия расплывчатости и меры принадлежности.

На основании теории расплывчатых множеств строятся расплывчатые алгоритмы (размытые, нечеткие), которые относятся к квазиалгоритмам (впрочем, как и вероятностные алгоритмы). Примерами расплывчатых инструкций в таких алгоритмах могут служить: «Солить по вкусу!», «Варить до готовности!» Реализовать на ЭВМ подобные размытые утверждения невозможно, поэтому для принятия и выбора применяют меру принадлежности. В этом случае для меры принадлежности вводят пороговые значения, позволяющие детерминировать реализуемый процесс.

Желающим более глубоко изучить теорию расплывчатости можно порекомендовать работу [72].

2.2. ОПРЕДЕЛЕНИЯ И НЕКОТОРЫЕ ПОНЯТИЯ ИЗ ТЕОРИИ ГИПЕРГРАФОВ

Прежде чем переходить к описанию гиперграфов попробуем коротко изложить смысл их использования для создания моделей, показать, в чем отличие этих моделей от графовых моделей.

Пусть имеется множество объектов A . Если при работе с математической моделью рассматриваемого объекта достаточно иметь дело только со связями между парами элементов этого множества, то предпочтительней иметь дело с графовыми моделями (под графовыми моделями подразумевается использование обычных графов, мультиграфов и псевдографов). Если необходимо рассматривать подмножества множества A и связи между этими подмножествами, отдельные из которых содержат более двух элементов, то лучше использовать при создании модели гиперграф.

Заметим, что описание гиперграфом охватывает как частный случай графовую модель, однако теория простых графов более развита, чем теория гиперграфов, поэтому не следует избегать возможности опираться на более мощный фундамент. С другой стороны, описание объекта проектирования гиперграфом может быть заменено описанием двудольным графом, но и здесь есть свои недостатки. Например, доказательство теоремы при выделении минимальных групп в терминах двудольных графов имеет более сложный и запутанный характер [22].

Отличительной чертой гиперграфа от графа является то, что ребра гиперграфа могут объединять произвольное число вершин. Отсюда следует, что гиперграф является обобщением понятия

неориентированного графа. Удобство описания электрической схемы гиперграфом обратило на себя внимание разработчиков САПР, и в настоящее время теория гиперграфов находит все более широкое применение при разработке алгоритмов автоматизированного проектирования.

Ряд определений и понятий будут нам полезны при дальнейшем изложении [83]. Гиперграфом $H=(X, U; R)$ называется пара множеств X и U , определенных следующим образом: $X = \{x\}$ — множество объектов, называемых вершинами; $U = \{u\}$ — множество подмножеств X , элементы которого называются ребрами. Заметим, что допускается существование двух ребер, содержащих одни и те же элементы, т. е. одно и то же подмножество множества X может являться n -кратным ребром; кроме того допускается существование ребер, не содержащих вершин вообще, т. е. включающих пустое подмножество; R — двухместный предикат $R(x, u)$, определенный для всех $x \in X, u \in U$, называемый инцидентом. Если ограничить число элементов в подмножествах двумя и исключить пустые ребра, то получим обыкновенный псевдограф; накладывая условия равенства числа элементов в подмножествах строго двум, получаем мультиграф; ограничивая повторяемость ребер, получаем обыкновенный граф. (Заметим, что одни авторы допускают петли в графе, другие считают это нецелесообразным и относят такие графы к псевдографам.)

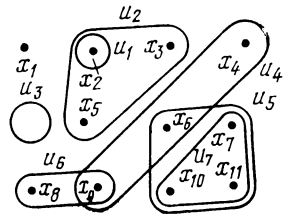


Рис. 2.5

Пример наглядного изображения гиперграфа приведен на рис. 2.5. Вершины гиперграфа назовем смежными, если существует хотя бы одно ребро, которому они обе инцидентны. Ребра гиперграфа назовем смежными, если существует хотя бы одна вершина, которой они инцидентны. Определим степень элемента гиперграфа (элементом может быть как ребро, так и вершина гиперграфа). Число ребер u , инцидентных вершине x , назовем степенью вершины x и обозначим $|U(x)|$. Число вершин x , инцидентных ребру u , назовем степенью ребра u и обозначим $|X(u)|$. Множество всех подмножеств в множестве X обозначим $S(X)$ и назовем классом. Элементы класса $S(X)$ обозначим рукописны-

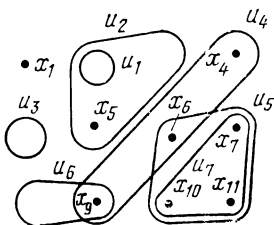


Рис. 2.6

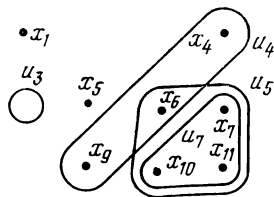


Рис. 2.7

ми буквами \mathcal{X}_i . Отвлекаясь от природы элементов, заметим, что $U \subseteq S(X)$.

Степенью элемента \mathcal{X}_i класса $S(X)$ назовем число ребер u , инцидентных вершинам $x \in \mathcal{X}_i$, и будем обозначать $|U(\mathcal{X}_i)|$.

Любой конечный гиперграф может быть однозначно задан матрицей инцидентностей. Матрица, соответствующая гиперграфу, изображенному на рис. 2.5, приведена ниже.

	u_1	u_2	u_3	u_4	u_5	u_6	u_7
x_1	0	0	0	0	0	0	0
x_2	1	1	0	0	0	0	0
x_3	0	1	0	0	0	0	0
x_4	0	0	0	1	0	0	0
x_5	0	1	0	0	0	0	0
x_6	0	0	0	1	1	0	0
x_7	0	0	0	0	1	0	1
x_8	0	0	0	0	0	1	0
x_9	0	0	0	1	0	1	0
x_{10}	0	0	0	0	1	0	1
x_{11}	0	0	0	0	1	0	1

Два гиперграфа H и H' называются изоморфными, если между множествами вершин X и X' и множествами ребер U и U' существует взаимно однозначное соответствие, сохраняющее инцидентность.

Гомоморфизмом ψ называется отображение гиперграфа $H(X, U; R)$ в гиперграф $H'(X', U'; R')$, если каждой вершине $x \in X$ и каждому ребру $u \in U$ гиперграфа H однозначно относятся их образы $\psi x \in X'$ и $\psi u \in U'$ в гиперграфе H' с сохранением инцидентности.

При решении задач проектирования приходится иметь дело с частью или частями схемы. Поэтому будут полезны следующие определения.

$H' = (X', U'; R')$ называется *полнореберным подграфом* гиперграфа $H = (X, U; R)$, если $X' \subseteq X$, $U' = U$ и сохраняется инцидентность. (Здесь и далее мы говорим подграф, суграф, относя эти понятия к гиперграфам, считая, что из контекста понятно, к каким графам это относится, в противном случае будут сделаны оговорки.) Пример полнореберного подграфа, полученного из гиперграфа на рис. 2.5, приведен на рис. 2.6.

Если вместе с вершинами удаляются и инцидентные им ребра, то получается *усеченный подграф*. Соответствующее изображение приведено на рис. 2.7. В полнореберном подграфе сохраняется смежность вершин, а в усеченном — нет.

$H'=(X', U'; R)$ называется *полновершинным суграфом* гиперграфа $H=(X, U; R)$, если $U' \subseteq U$, $X'=X$ и сохраняется инцидентность. Пример, соответствующий удалению ребер из гиперграфа, показанного на рис. 2.5, приведен на рис. 2.8. Если вместе с ребрами удаляются и инцидентные им вершины, то получается *усеченный суграф*. Соответствующий пример приведен на рис. 2.9. Как и для подграфов, в полновершинном суграфе сохраняется смежность ребер, а в усеченном — нет.

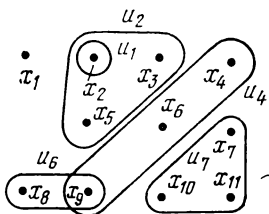


Рис. 2.8

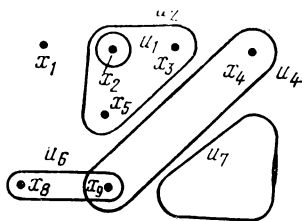


Рис. 2.9

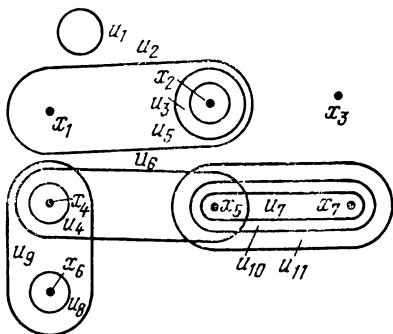


Рис. 2.10

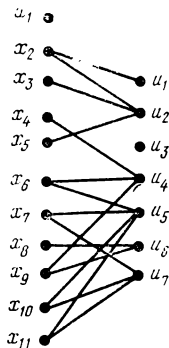


Рис. 2.11

$H^*=(X^*, U^*; R^*)$ называется *двойственным гиперграфом* гиперграфу $H=(X, U; R)$, если $X^*=U$; $U^*=X$; $R^*(u, x)=R(x, u)$. На рис. 2.10 показан гиперграф, двойственный к изображенному на рис. 2.5.

Матрица инцидентий двойственного гиперграфа может быть получена транспонированием матрицы инцидентий исходного гиперграфа. Ниже приведена матрица, соответствующая двойственному гиперграфу, показанному на рис. 2.10.

Нетрудно заметить, что число компонент связности в двойственном гиперграфе не изменяется. Суграфу (подграфу) в исходном гиперграфе соответствует подграф (суграф) в двойственном гиперграфе.

$$\begin{array}{c}
 u_1^* u_2^* u_3^* u_4^* u_5^* u_6^* u_7^* u_8^* u_9^* u_{10}^* u_{11}^* \\
 x_1^* \left| \begin{array}{cccccccccccc}
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0
 \end{array} \right.
 \end{array}$$

Любой гиперграф представляется двудольным графом; в этом случае элементы гиперграфа представляются вершинами, а ребра двудольного графа соединяют эти вершины, если в гиперграфе выполняется соответствующее равенство $R(x, u) = 1$. На рис. 2.11 показан двудольный граф, соответствующий изображенному на рис. 2.5 гиперграфу. С другой стороны, всякий двудольный граф может быть представлен гиперграфом. При записи двудольного графа в виде $G = (Z', Z'', U)$ одно из множеств вершин Z' или Z'' представляется вершинами образуемого гиперграфа, а другое — ребрами. Множество U гиперграфа показывает, какие вершины входят в какие ребра. Таким образом, отображение получается с точностью до двойственности. Такое взаимное представление позволяет переносить результаты, полученные для гиперграфов, на двудольные графы, и наоборот.

2.3. НЕКОТОРЫЕ ВОПРОСЫ ТЕОРИИ ПРИНЯТИЯ РЕШЕНИЙ

Возрастающая роль интерактивных систем проектирования повышает интерес к теории принятия решений. Имеется много примеров, когда из множества альтернатив, имеющих равные или приблизительно равные количественные оценки, лучшее решение может выбрать только опытный проектировщик, оставляя ЭВМ работу по подготовке альтернатив и принятию решений в бесспорных ситуациях. Знакомство с теорией принятия решений разработчику САПР будет полезно для построения с большим пониманием обобщенных схем и алгоритмов как отдельных этапов, так и всего процесса проектирования, облегчит разработку более эффективных методов сокращения перебора и в ряде других случаев. Непосредственному принятию решений предшествует анализ ситуации, позволяющий взвесить все «за» и «против». При принятии решения человеком влияние на результат оказывают не только количественные показатели, но и психологические факторы, которые влияют на появление таких расплывчатых качественных категорий, как «лучшее», «прекрасное», «более приятное» и т. п.

Принципы выбора решений применялись уже на первых стадиях автоматизации и моделирования действий человека. Так,

в классическом алгоритме Ли заложен динамический принцип принятия решений.

Для реализации процесса принятия решений необходимы следующие факторы: цель (цели), информация к начальному моменту принятия решений, наличие альтернатив, критерии, влияющие на выбор альтернатив.

Цель, которая ставится для достижения, может быть в начальный момент сформулирована глобально. Например, выполнить работу в срок, разработать передающую аппаратуру, купить продукты к обеду и т. п. При реализации процесса принятия решений цель может конкретизироваться, как-то: выполнить работу в срок может означать написание и отладку программ, а также разработку сопровождающей документации; под разработкой аппаратуры понимается изготовление опытного образца и подготовка документации к серийному производству и т. д.

Цель может быть расплывчатой. Например, выбрать несколько лучших студентов факультета, провести максимально возможное число проводников при трассировке. В последнем примере для плат с невысокой плотностью монтажа данная цель является нерасплывчатой, так как возможна реализация всех заданных связей; при высокой плотности — точное значение числа возможно реализуемых связей указать трудно, поэтому цель является расплывчатой. Заметим, что в этом примере граница между расплывчатой и нерасплывчатой целями сама размыта.

Для принятия решений может возникнуть необходимость конкретизации цели, т. е. подразделения (разбиения) ее на более подробные цели более низкого уровня, что способствует разъяснению более общей цели. Конкретизация общей (глобальной) цели, т. е. разбиению ее на более мелкие, подробные цели, соответствует построению иерархии «сверху вниз». Одновременно цели нижележащего уровня можно считать средствами (носителями информации) для достижения целей вышележащего уровня. Таким образом, отправляясь от цели (целей) нижнего уровня и рассматривая их как необходимое средство достижения вышеразположенного уровня, можно построить иерархию целей «снизу вверх».

Выдвигаемые цели могут противоречить друг другу в том смысле, что одна может быть достигнута только за счет другой.

Информация для принятия решений имеет особо важную роль при многошаговых процедурах, когда для очередного шага информацией служат достигнутые цели предыдущего шага и информация, привлекаемая только для данного момента. Если по имеющейся информации можно получить количественные оценки для однозначного определения пути решения, то в данной ситуации нет необходимости привлекать лицо, принимающее решение (ЛПР) (здесь и далее под ЛПР мы понимаем не только человека, но и программные средства принятия решений). Эту аббревиатуру мы сохраним только в данном параграфе, чтобы не отходить от классических определений; если под ЛПР будет в дальнейшем

пониматься только человек, то будем употреблять термин «пользователь».

Такая ситуация, как правило, складывается при принятии решений в условиях определенности, когда по имеющейся информации можно сделать точные количественные оценки. В ряде случаев, когда количественные оценки для предполагаемых альтернатив совпадают, необходимо либо построение автоматизированных процедур, либо принятие решения произвольного с равной функцией полезности.

Как уже отмечалось выше, при многошаговых процедурах принятия решений информацией для i -го шага служит и цель $(i-1)$ -го шага (шагов). Так, для прокладки трасс необходимо иметь информацию о связях, которая является исходной до начала трассировки, информация о размещении элементов, что было целью предыдущего этапа проектирования и т. п. Цель предоставления информации — помочь ЛПР в выборе приемлемого решения проблемы в сложных ситуациях.

Наличие альтернатив является необходимым фактором процесса выбора и принятия решений. Примеры альтернатив в нашей жизни многочисленны: выбор месяца отдыха, кинофильмы, телепередачи, различные программы размещения элементов, программы трассировки, целые системы проектирования аппаратуры и даже в классической задаче о разборчивой невесте мы сталкиваемся с проблемой выбора.

Решения принимаются с учетом различных выдвигаемых критериев. Это могут быть экономические соображения, эстетические, технические требования и т. п. В большинстве случаев мы имеем дело с принятием решений в многокритериальном пространстве. Например, при приобретении магнитофона в качестве критериев выдвигаются не только его акустические данные и стоимость, но и габариты, его внешнее оформление, источники питания и пр.

Одной из сложностей проблемы принятия решения является необходимость учета большого количества противоречивых критериев. В задаче выбора наилучшего решения из множества допустимых, которая возникает в этом случае, понятие наилучшего не может быть сформулировано на основе только информации о значениях оценок по каждому критерию отдельно.

В ряде случаев нам необходимо иметь дело с критериями-заменителями. Здесь возможны два случая: равнозначные критерии-заменители и неравнозначные заменители. Два критерия называются равнозначными тогда и только тогда, если их отображения совпадают, т. е. два критерия $K_1(x)$ и $K_2(x)$ равнозначны, если $K_1(x) \equiv K_2(x)$ для любых $x \in X$, где X — множество альтернатив. Примером равнозначных критериев может быть требование при компоновке минимума числа связей между модулями — критерий $K_1(x)$, и требование максимума связей между выводами одного модуля — критерий $K_2(x)$. Каждый из этих критериев может быть равнозначным заменителем другого. Использование

критериев-заменителей подобного рода позволяет оптимизировать вычислительный процесс вследствие специфики информации, необходимой для каждого критерия.

Неравнозначные критерии-заменители не приводят к тождественному получаемому решению, т. е. однозначно они не заменяют основной имеющийся критерий, учитывая его при этом неявно, приблизительно. Необходимость использования неравнозначных критериев-заменителей обусловлена трудностями реализации основного критерия либо невозможностью его реализации вообще. При проектировании двухслойных печатных плат примером неравнозначных заменителей может служить критерий минимизации числа пересечения проводников $K_{зам}(x)$ вместо основного критерия — минимизации числа переходных отверстий $K_{осн}(x)$. В данном случае $K_{зам}(x)$ позволяет косвенно учитывать предположительное число переходных отверстий, но не дает ответа на основной вопрос.

При разработке САПР нас интересуют вопросы принятия решений как при создании алгоритмов, моделирующих, насколько это возможно, действия конструктора-проектировщика алгоритмов, так и при минимизации времени счета.

Рассмотрим применение теории выбора и принятия решений на примере размещения элементов.

Глобальная цель в данном примере — разместить элементы на плате. Информацией является: модель платы для размещения (она может не совпадать с моделью платы для трассировки), связи между элементами, данные об элементах (габаритные размеры, расположение выводов, ориентация и т. п.), сведения о некоторых закрепленных директивно элементах. Критериями размещения служат результаты трассировки, что невозможно использовать на данном этапе; поэтому выдвигаются различные критерии-заменители; минимизация суммарной длины связей, минимум связей между выделяемыми областями, минимизация пересечений связями условных линий и т. п. (в гл. 3 будет показано, что улучшение результатов размещения перестановками элементов или группами элементов может быть сведено к модели последовательного принятия решения). Обозначим эти критерии K_1, K_2, \dots, K_j . При установке очередного $(i+1)$ -го элемента у нас имеется выбор среди свободных для размещения позиций Y_{i+1} — множество альтернативных позиций.

Пусть на плате уже размещено i из n элементов ($n > i$), имеется также $(m-i)$ свободных для размещения позиций ($m > n$). Предположим, что критерии расположены по предпочтению $K_1 > K_2 > \dots > K_j$. Размещая элемент $(i+1)$, можно предложить следующую процедуру. Из множества альтернатив Y_{i+1} выбираем подмножество перспективных альтернатив Y'_{i+1} ($Y'_{i+1} \subset Y_{i+1}$). Например, для критерия минимальной суммарной длины множество Y'_{i+1} будет состоять из свободных позиций, соседних с ранее установленными элементами. Размещая очередной эле-

мент $i+1$, минимизируем критерий $K_1 : \min_{Y'_{i+1}} K_1(i+1, y)$.

Результат может иметь два следующих варианта: первый — существует точка $y_0 \in Y'_{i+1}$, в которой и только в ней достигается минимум, — так называемое беспорное превосходство; второй — существует подмножество множества $Y''_{i+1} \subset Y'_{i+1}$ мощностью более единицы, на котором достигается минимум по критерию $K_1(i+1, y')$:

$$Y'_{i+1} = \{y'' \in Y'_{i+1} : K_1(i+1, y'') = \min_{y' \in Y'_{i+1}} K_1(i+1, y')\}.$$

При беспорном превосходстве решение о размещении $(i+1)$ -го элемента принимается однозначно. Если существуют эквивалентные с точки зрения примененного критерия позиции, можно элемент разместить в произвольную из них, можно передать вопрос о принятии альтернативного решения ЛПР через средства диалога либо оценить позиции по следующему критерию $K_2(i+1, y'')$:

$$Y''_{i+1} = \{y''' \in Y''_{i+1} : K_2(i+1, y''') = \min_{y'' \in Y''_{i+1}} K_2(i+1, y'')\}.$$

При $|Y'_{i+1}| = 1$ мы опять имеем дело с беспорным превосходством, при $|Y'_{i+1}| > 1$ процесс повторяется. Обобщенная схема рассмотренного процесса приведена на рис. 2.12.

Описанный выше процесс означает последовательное отображение множества альтернатив в свое подмножество (возможно, полностью в себя, когда число альтернативных позиций не сокращается).

Сложность одновременной реализации многих критериев заставляет порой разлагать их на составляющие, проводить анализ принятых решений и, с учетом дополнительно привлекаемой информации, корректировать ранее принятое решение. Типичным примером может служить распределение БЭ по корпусам ИС. После распределения БЭ по корпусам проводится анализ полученного решения; с учетом информации об используемой серии ИС проводится перераспределение БЭ в корпусе и переназначение выводов в связях (более подробно данный вопрос будет рассмотрен в последующих главах). Еще один пример анализа принятого решения перед окончательным принятием — автоматическая перетрассировка, когда по результатам трассировки выявляют конфликтные ситуации, определяют степень конфликтности, после чего часть ранее проведенных трасс снимается, проводятся новые, не реализованные на предыдущем шаге, а затем делается попытка проведения снятых трасс.

При проектировании сложных систем принятия решений анализ полученных результатов необходим, так как он может открыть новые, не замеченные ранее аспекты.

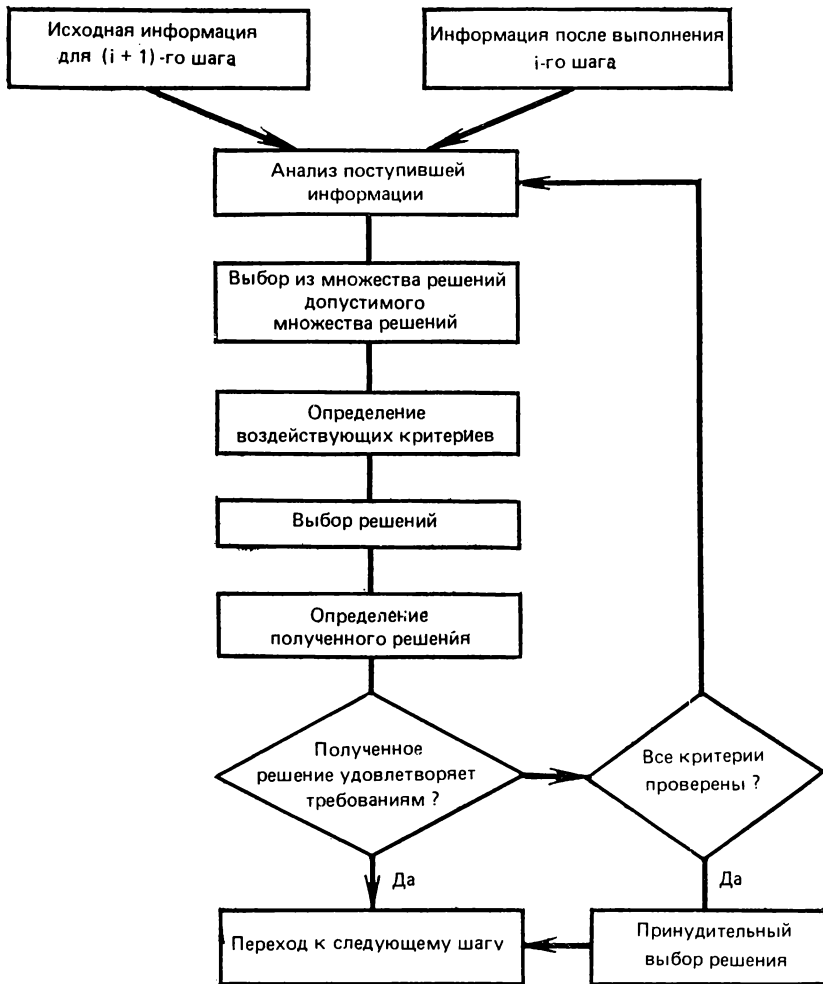


Рис. 2.12

При последовательном принятии решений будущие действия могут зависеть от информации, которая становится известной в процессе продвижения к будущему. С этим знакомы все разработчики САПР. Динамическая стратегия принятия решений должна учитывать информацию, поступающую в процессе такого решения проблемы и использовать ее для снятия «неопределенности» при принятии решений. Если принимаемое решение не приводит к поставленной цели (дает неоднозначное или неудовлетворительное решение), необходимо повторить процесс анализа с учетом вновь полученной информации.

При выдвигании расплывчатых целей формальный анализ не всегда позволяет делать выбор программно, здесь лучше приме-

нять средства диалога, так как личный опыт проектировщика позволяет лучше учитывать тонкие места.

Следует иметь на вооружении и возможность сведения нескольких критериев в один, что в дальнейшем лучше поддается программной реализации. Такая возможность широко используется при реализации волновых алгоритмов, где с весовыми коэффициентами учитываются в виде суммы критерии числа пересечений трасс, числа поворотов, желательное направление и т. д.

2.4. МЕТОДЫ СОКРАЩЕНИЯ ПЕРЕБОРА

Решение задач конструирования связано с необходимостью выбора решений из множества вариантов. Порой число вариантов столь велико, что оценить каждый не представляется возможным. Например, число возможных размещений 50 одинаковых элементов на 50 равнозначных позициях равно $50!$, что немислимо оценить ни на одной из известных самых мощных ЭВМ, даже если на оценку одного варианта потребуется 0,1 наносекунды (конечной оценкой следует считать результаты трассировки). Другой, более простой пример. Если нам необходимо оценить все возможные кратчайшие пути между двумя диагонально противоположными углами квадрата со стороной n на ортогональной сетке, то для этого придется просмотреть C_n^{2n} вариан-

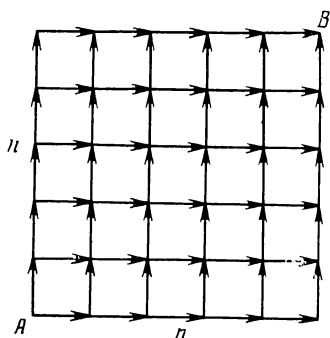


Рис. 2.13

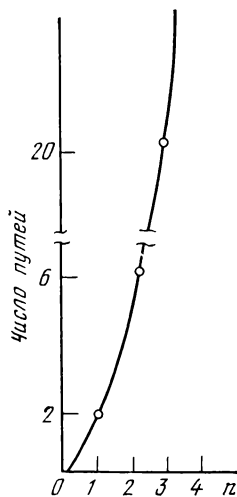


Рис. 2.14

тов (при предположении, что препятствий на пути нет) (рис. 2.13). При возрастании n число вариантов резко увеличивается (рис. 2.14). Число кратчайших путей между диагонально противоположными углами прямоугольника размером $n \times m$ на ортогональной сетке равно $C_{m+n}^n = C_{m+n}^m$. И это только для кратчайших путей! Такая многовариантность на различных этапах конструирования выдвигает проблему сокращения числа просматриваемых вариантов на одно из первых мест при автоматизации конструкторских работ.

Прежде чем рассматривать вопросы сокращения просматриваемых вариантов, остановимся на некоторых методах поиска решений. При решении задач комбинаторными методами бывает удобно пользоваться представлением процесса решения в виде направленного графа. В этом случае имеется начальная вершина, которую называют истоком, корнем или правершиной, от которой направлены дуги к дочерним вершинам. При разложении задачи на подзадачи вершины соответствуют подзадачам некоторого уровня, при решении задачи в пространстве состояний — это очередные состояния. Дуги соответствуют переходам от подзадачи к подзадаче при решении задач или переходу от состояния к состоянию во втором случае. Необходимо отличать конечные вершины, из которых не выходит ни одна дуга. Их также именуют вершинами стока.

Представление процесса решения в виде графа имеет сложный и недостаточно наглядный характер, поэтому целесообразно иметь дело с деревьями.

Напомним, что деревом называется односвязный граф без циклов.

Известно несколько определений деревьев [73], которые между собой эквивалентны. Приведем некоторые из них. Граф с n вершинами и k ребрами будет деревом, если: любые две вершины соединены единственной простой цепью, или G — связный граф и $n=k+1$, или G — ациклический граф и $n=k+1$, или G — односвязный граф и удаление любого одного ребра делает его двусвязным графом.

Простой цепью называется маршрут, в котором все вершины различны. Маршрутом, в свою очередь, называется чередующаяся последовательность ребер и вершин.

Выше мы говорили о неориентированных деревьях. При построении дерева решений мы имеем дело с ориентированными деревьями, отличительной чертой которых является ориентация ребер, т. е. упорядочение пар различных вершин. Покажем, как сводить процесс представления решения на графе к процессу решения на дереве.

Рассмотрим орграф, изображенный на рис. 2.15,а. Пронумеруем вершины последнего, четвертого уровня: 4.1; 4.2; 4.3. Каждую вершину, в которую входит более одной дуги, расщепим на подвершины по числу входящих в нее дуг и перенумеруем их снова: 4.1.1; 4.1.2; 4.2.1; 4.2.2; 4.3 (рис. 2.15,б). Теперь можно для наглядности перенести расщепленные подвершины каждую под свою родительскую (рис. 2.15,в). Пронумеруем теперь вершины предыдущего уровня (рис. 2.15,г). Видно, что в результате предыдущего шага все подграфы с родительскими вершинами третьего уровня являются деревьями. Теперь расщепим, как и в предыдущем случае, вершины третьего уровня, только при этом теперь расщепляются и дочерние вершины (рис. 2.15,д). Переносим расщепленные вершины каждую под свою родительскую вместе с дочерними вершинами и подграфами, получим граф,

изображенный на рис. 2.15,е. Нетрудно заметить, что любой путь от истока к любой вершине стока графа на рис. 2.15,а имеется в дереве, показанном на рис. 2.15,е.

Некоторые оргграфы нельзя представить подобным образом в виде деревьев, если, например, две вершины будут дочерними друг к другу, т. е. соединены между собой встречными дугами. Тем не менее оргграфы, представляющие собой пространство состояний или процесс разложения задач на подзадачи, всегда могут быть представлены описанной выше процедурой в виде дерева.

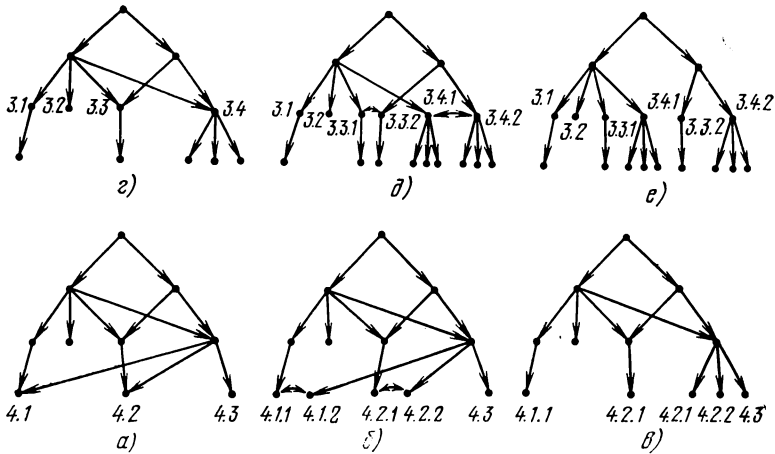


Рис. 2.15

В дальнейшем мы будем рассматривать вопросы принятия решений только на деревьях, а не на графах, так как этот способ нагляден, хотя и имеет один недостаток: в рассматриваемом примере число вершин стока увеличилось на 6, т. е. в три раза, что при реализации на ЭВМ повлечет за собой увеличение объема используемой памяти; кроме того, необходимо запоминать расщепленные вершины, так как при отсечении в одном поддереве расщепленной вершины желательно исключать из рассмотрения и все остальные, соответствующие этой расщепленной, что необходимо для сокращения перебора.

Остановимся подробнее на деревьях.

Наибольший интерес представляют деревья со стекающей структурой, подобной водной системе реки (рис. 2.16,а) и с растекающейся, подобной водоснабжению (газо-, электроснабжению и т. п.) (рис. 2.16,б). В деревьях с растекающейся структурой вершины, не имеющие дочерних, т. е. не являющиеся корнем никакого поддерева, именуется листьями.

Целесообразно рассматривать процесс решения задач с помощью разбиения на подзадачи и с помощью пространства состояний. Многие методы принятия решений на дереве состояний

и на дереве подзадач близки друг к другу по идеям и реализации, поэтому, описав построение деревьев для указанных процессов, мы в дальнейшем будем рассматривать какую-нибудь одну структуру.

При построении дерева решений разбиением задачи на составляющие части корнем (истоком) дерева является исходная задача. Причиной, заставляющей строить такое дерево решений, являются трудности решения исходной задачи, а порой и невозможность этого.

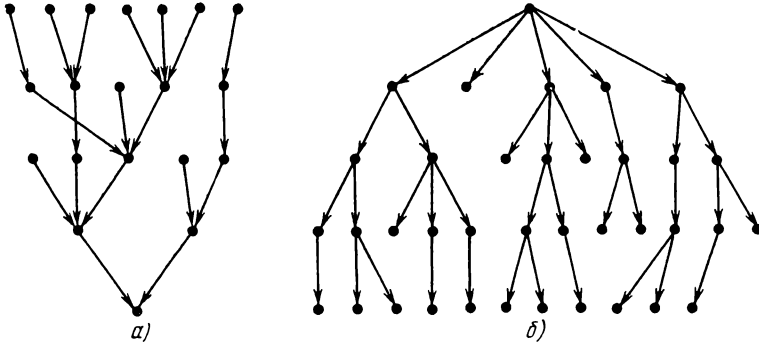


Рис. 2.16

Рассмотрим методику построения дерева решений. Номером уровня подзадачи будем называть число дуг от корня до данной подзадачи дерева. Пусть мы рассматриваем j -ую вершину i -го уровня дерева решений — x_i^j , и пусть эта задача может быть разбита на k подзадач $x_{i+1}^1, x_{i+1}^2, \dots, x_{i+1}^k$.

Можно построить следующие ветвления: рассмотреть: 1) на $(i+1)$ -м шаге каждую подзадачу (рис. 2.17,а); 2) на $(i+1)$ -м шаге группы из подзадач, предварительно разбив подзадачи на группы (рис. 2.17,б); 3) на $(i+1)$ -м шаге одну задачу и ее дополнение, т. е. все остальные подзадачи (рис. 2.17,в). Вообще способы ветвления 1 и 3 можно рассматривать как частные случаи способа 2, но они порождают различные методы решения задачи.

При сведении решения задачи к подзадачам следует различать способы ветвления И и ИЛИ. Поясним подробнее. Пусть у нас имеется вершина x_i^j , подлежащая разбиению на подзадачи. В результате ветвления получаются две подзадачи x_{i+1}^1 и x_{i+1}^2 . Если для решения исходной задачи x_i^j необходимо решить как подзадачу x_{i+1}^1 , так и подзадачу x_{i+1}^2 , то такое ветвление называется И. Если для решения задачи x_i^j достаточно решить задачу x_{i+1}^1 либо задачу x_{i+1}^2 , то такое ветвление называется ИЛИ. Несложно распространить данные понятия и на большее число подзадач. Следует иметь

в виду, что при разбиении задачи x_i^j возможны, да это и встречается на практике, комбинации описанных выше ветвлений, что может звучать следующим образом: чтобы решить задачу x_i^j необходимо решить подзадачу $x_{i+1}^{j_1}$ или подзадачу $x_{i+1}^{j_2}$ или подзадачи $x_{i+1}^{j_3}$ и $x_{i+1}^{j_4}$ одновременно и т. д. (рис. 2.18).

Примером ветвления типа И может быть разбиение задачи: *купить все продукты для приготовления шашлыка*, на подзадачи: *купить баранину, лук, уксус, вино* и т. п.

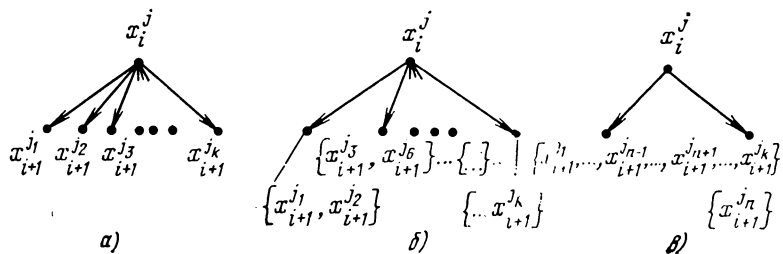


Рис. 2.17

Примером ветвления типа ИЛИ может служить разбиение задачи: *купить что-нибудь для чая*, что можно представить в виде решения одной из подзадач: *купить торт, купить печенье или купить баранки* и т. п.

Листья ветвей подзадач могут располагаться на различных уровнях; так, в задаче *к чаю* можно купить торт — задача будет решена уже на первом уровне; или купить блинной муки — первый уровень, испечь блины — второй уровень.

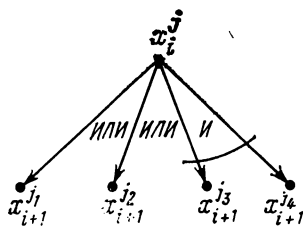


Рис. 2.18

Другой подход к построению дерева решений связан с пространством состояний. Пространство состояний — это множество всех состояний, достижимых из начального состояния. Перед началом решения задачи мы имеем некоторое начальное состояние. Например, в качестве начального состояния перед этапом трассировки будет поле трассировки без проводников с нанесенными

на него контактами для соединений и различными препятствиями. Одним из конечных состояний может быть рисунок трассировки с проложенными по возможности трассами. Следует заметить, что конечное состояние может совпадать с целевым, а может и не совпадать. Так, если трассировка закончена, но проложены не все связи, то получили конечное состояние, отличающееся от целевого, где подразумевается реализация всех связей (данный пример говорит о том, что достижение листа дерева вовсе не означает решения задачи, тем более оптимального). Пре-

образование из одного состояния в другое происходит с помощью оператора перехода.

Из сказанного видно, что пространство состояний удобно изображать в виде орграфа, где вершины соответствуют состояниям, а дуги — операторам. При явном задании пространства состояний должны быть перечислены явным образом все дуги и вершины. Для больших размерностей такой способ практически неприменим. При неявном задании пространства состояний задается начальное состояние в виде описания исходного множества вершин, а также оператор (ы) образования дочерних вершин. Возможно задание множества операторов, но в этом случае область действия каждого оператора должна быть строго определена. Вполне естественно, что целевое состояние должно быть определено и при явном, и при неявном задании пространства состояний, т. е. мы должны знать, к чему стремиться.

Итак, для полного представления задачи в пространстве состояний необходимо иметь: описание всех состояний при явном задании, а при неявном только начального состояния; множество операторов перехода от состояния к состоянию (операторов образования дочерних вершин); описание конечного состояния.

Покажем на примере определения шестизначного номера телефона решение задачи в пространстве состояний [74].

Начальное состояние — шесть неизвестных цифр в номере. Промежуточные состояния — цифры на различных (из шести) позициях, на каждом следующем уровне на одну цифру в номере больше: целесообразно, чтобы на каждом уровне появлялась цифра в одной и той же позиции. Если задавать пространство состояний неявным образом, то оператор образования дочерних вершин описывается как добавление в родительскую вершину на одну из свободных позиций по очереди одной из десяти цифр, не считая первой позиции, так как телефонные номера с 0 обычно не начинаются. Конечное состояние — номера-наборы из шести цифр.

Условия задачи следующие: 1) две средние цифры в номере, третья и четвертая, равны; 2) вторая половина в номере телефона в четыре раза больше первой; 3) вторая цифра вдвое больше первой; 4) третья цифра либо в два раза, либо на две единицы больше второй и одна из них двойка.

Так как оператор перехода — проставление цифры в одну из свободных позиций, то очевидно, что из каждой родительской вершины будет выходить по десять дуг. Исключение составляет вершина, дочерней к которой будут вершины, в которые проставляются цифры на первую позицию. Здесь будет девять дуг. Таким образом, получаем ниже корневой вершины шесть уровней. Номер заполняемой цифрой позиции не обязательно должен совпадать с номером уровня.

Построение пространства состояний целесообразно вести с учетом условий задачи в следующем порядке: на 1-м уровне заносим цифру в третью (четвертую) позицию, на 2-м — в четвертую (третью), на 3-м — в первую, на 4-м — во вторую, на 5-м — в шестую, на 6-м в пятую позицию.

Можно предложить следующий процесс решения задачи. На 1-м уровне рассматриваются все дочерние вершины. На 2-м только вершины, у которых третья и четвертая цифры совпадают (условие 1), а остальные исключаются из рас-

смотрения вместе со всеми дочерними. На 3-м уровне рассматриваются вершины с цифрами на первой позиции, равными 1 или 2 ($x_1 = [x_4/4]$ (по условию 2)). На 4-м уровне выделяются вершины со второй цифрой 2 или 4 (по условию 3), а по условию 4 исключаются из дальнейшего рассмотрения все, кроме соответствующих вершин. На 5-м уровне рассматривается вершина с цифрой 6 на шестой позиции (по условию 2). На 6-м, последнем уровне по условию 2 определяем решение. На рис. 2.19 показан процесс решения этой задачи в пространстве состояний, где изображены только анализируемые на каждом уровне вершины. Дуги к дочерним вершинам, которые не участвуют в рассмотрении, перечеркнуты.

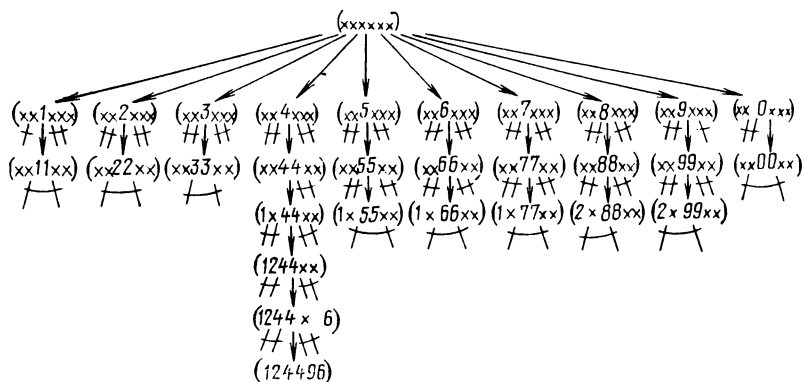


Рис. 2.19

Рассмотрим некоторые методы поиска решений на деревьях. Пусть у нас имеется множество операторов перехода $\{G_i\}$, определяющих ветвление вершин i -го уровня: $G_i(x_i^j) = \{x_{i+1}^{h_j}\}$ и x_0 — корень дерева. Методы перебора на дереве решений можно разбить на два принципиально различных подхода.

Один подход — поиск в глубину. При таком подходе для очередного рассмотрения выбирается вершина следующего уровня, пока не достигнем очередного листа. Если достижение листа дерева не дает желательного решения задачи, то просматриваем следующий путь от корня до другого листа и т. д. Последовательность анализа вершин можно записать в виде $x_0, G_1x_0, G_2(G_1x_0), G_3(G_2(G_1x_0)) \dots$ С этой целью упорядочения поиска в глубину при первом подходе очередной выбирается либо последняя получившаяся при образовании уровня вершина, либо первая вершина уровня. После достижения листа таким образом возвращаются на предыдущий уровень и из точки ветвления просматривают очередную вершину, двигаясь аналогичным образом по порожденному данной вершиной поддереву. Если просмотрено все поддерево этой вершины, то возвращаемся обратно, еще на один уровень выше и т. д. Поэтому поиск в глубину называют еще поиском с возвращением. На рис. 2.20 показан пример обхода дерева при поиске в глубину.

Другой подход — поиск в ширину. При этом просмотр вершин дерева ведется по уровням, а переход к вершинам другого уровня происходит только после просмотра всех вершин данного уровня, т. е. сначала рассматриваются вершины $\Gamma_1 x_0$, затем вершины $\Gamma_2(\Gamma_1 x_0)$, затем $\Gamma_3(\Gamma_2(\Gamma_1 x_0))$ и т. д.

Очевидно, что при отыскании всех решений оба подхода эквивалентны. Если нам достаточно найти одно любое положительное решение, то целесообразность применения поиска решения по глубине или по ширине определяется структурой дерева решений. Здесь можно дать только общие рекомендации. Для широких деревьев, листья которых расположены на сравнительно близких уровнях, целесообразно вести поиск по глубине (рис. 2.21). Для глубоких узких деревьев, когда листья могут встречаться на различных уровнях и разброс листьев по уровням велик, целесообразно применять поиск по ширине (рис. 2.22). Эффективность нахождения одного решения в обоих методах зависит от удачного выбора вершин.

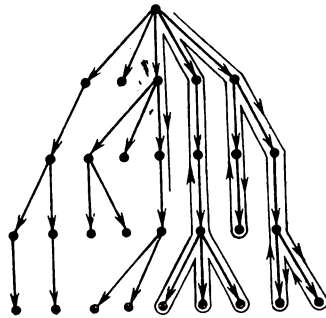


Рис. 2.20

При поиске оптимальных решений необходимо просматривать большое число вершин. Одним из наиболее распространенных методов сокращения поиска является метод ветвей и границ [75]. Метод хорошо описан в литературе, поэтому изложим только его идею.

При наличии дерева решений определяется числовое значение предполагаемого решения. Одним из возможных методов получения этой оценки может быть просмотр любой ветви до листа, соответствующего решению. При минимизации функции решаемой задачи данная оценка принимается за нижнюю границу. Далее, действуя, как и в поиске в глубину, для каждой порождающей вершины определяем ее оценку. Если получаем большую оценку, то вершина вместе с порождаемым ею поддеревом исключается из рассмотрения; если меньшую оценку, то просматриваются дочерние вершины с определением оценки. Если достигнем листа с меньшей оценкой, чем имела ранее, то старая нижняя оценка заменяется на новую, и запоминается новое соответствующее решение. Метод ветвей и границ позволяет получить оптимальное решение.

Однако при большом объеме просматриваемых вариантов метод ветвей и границ неприемлем, так как объем вычислений остается велик, да и для отсеивания по нижней оценке приходится просматривать большое количество вершин. В связи с этим часто прибегают к построению эвристических процедур нахождения квазиоптимальных решений. Например, для выбора пути движения по дереву решений применяют оценочную функцию. С помощью оценочной функции определяется путь, наиболее перспективный при движении по дереву. В ряде случаев строят усеченное дерево перебора, т. е. ограничивают число уровней ветвления. Такой подход эффективен при улучшении уже имеющихся решений.

Следует остановиться на методе горизонта, предложенном в работе [76]. Идея метода заключается в ограничении глубины просмотра в каждой точке местонахождения. Автор метода

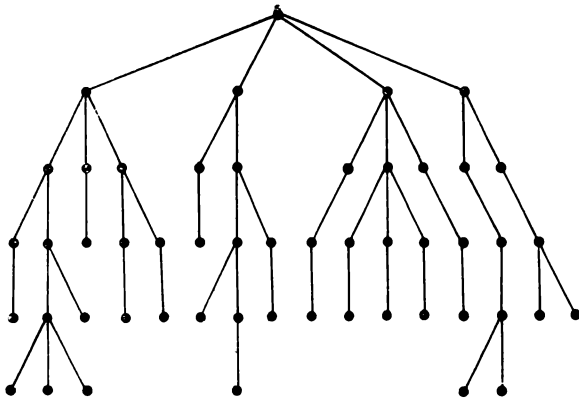


Рис. 2.21

успешно применил его для обучения ЭВМ игре в шахматы. В повседневной жизни мы часто применяем такой метод. Так, при переходе нерегулируемого перекрестка мы также пользуемся методом горизонта, сами того не замечая. Сначала оцениваем возможность перейти одну полосу движения, достигаем промежуточной цели, а затем оцениваем возможность перейти встречную полосу движения и снова достигаем цели.

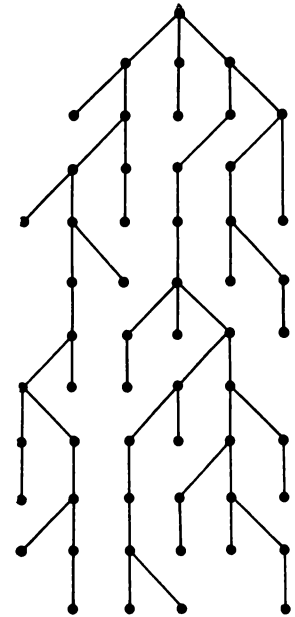


Рис. 2.22

Приведем пример применения этого метода при автоматизации конструирования. Пусть у нас на плате размещено уже несколько элементов и необходимо разместить еще. Если для размещения очередного элемента выбирается претендент, наиболее связанный с ранее установленными, то при установке нет смысла пробовать разместить его на все свободные позиции. Имеет смысл определить для установки перспективные места и анализировать только их. Определение перспективных мест зависит от минимизируемого функционала. При минимизации суммарной длины можно, например, ограничить горизонт видимости смежными позициями с ранее установленными элементами. При линеаризации соединений в горизонт в первую очередь включают столбцы и колонки, в которые ранее установлены элементы, связанные с ранее установленными.

Метод горизонта применяется и в ряде алгоритмов трассировки, правда, авторы не подчеркивают этой терминологии.

МАТЕМАТИЧЕСКИЕ МЕТОДЫ И МОДЕЛИ ЗАДАЧ ОПТИМИЗАЦИИ КОНСТРУИРОВАНИЯ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ

3.1. АНАЛИЗ ПРОЦЕССА КОНСТРУИРОВАНИЯ ПЕЧАТНЫХ ПЛАТ

Постоянный анализ процесса конструирования, а также существующих и разрабатываемых методов помогает выработать новые взгляды на известные позиции, переосмыслить с новой точки зрения имеющиеся подходы, открыть новые закономерности, ранее остававшиеся незамеченными. Поэтому процесс проектирования печатных плат (ПП) проанализирован авторами с точки зрения теории принятия решений.

На входе процесса проектирования ПП имеется функциональная схема, которая содержит информацию о базовых элементах (БЭ), о связях между ними и о внешних связях. Кроме того, задаются технологические параметры: размер платы, разрешенные для применения в данной разработке серии микросхем, шаг конструкторской сетки и т. п. В результате проектирования мы должны получить рисунок трассировки. Общей целевой функцией всего процесса проектирования ПП следует считать число проведенных связей $\Phi(\cdot)$ (в данном случае под связью понимается соединение между двумя контактами одной цепи электрической схемы). Целевая функция $\Phi(\cdot)$ зависит как от случайных входных параметров ξ (например, числа БЭ, их качественного состава, числа задействованных выводов элементов схемы и т. п.), так и от переменных \bar{r} , статистически устойчивых для класса разработок (например, размеров платы, серий микросхем, разрешенных для применения и т. п.).

Хотя оптимальный размер платы может быть определен с достаточной точностью как функция входных параметров и вопрос об определении габаритов платы неоднократно ставился в литературе, однако практического применения не нашел, так как изготовление и применение плат различных размеров в одной разработке связано с изменением технологической оснастки производства, с перенастройкой оборудования и с изменением габаритов аппаратуры. Этим можно объяснить подход, когда типоразмер задается, как правило, единым для целой разработки, поэтому подобные величины считаются статистически устойчивыми, т. е. не являются случайными.

Действия над входными величинами (ξ , \bar{r}) в процессе проектирования можно представить следующим образом. Обозначив

совокупность используемых в системе алгоритмов компоновки (в данном случае мы имеем в виду задачу покрытия) K , совокупность алгоритмов размещения P и совокупность алгоритмов трассировки T можно представить схему последовательной реализации основных алгоритмов (рис. 3.1,а).

Использование алгоритмов совместного решения задач компоновки и размещения отображено на рис. 3.1,б; размещения и трассировки — на рис. 3.1,в; компоновки и трассировки — на рис. 3.1,г, что соответствует организации различных обратных связей. Относительно рис. 3.1,г следует заметить, что трассировка на этапе компоновки делается на условном рабочем поле, а пос-

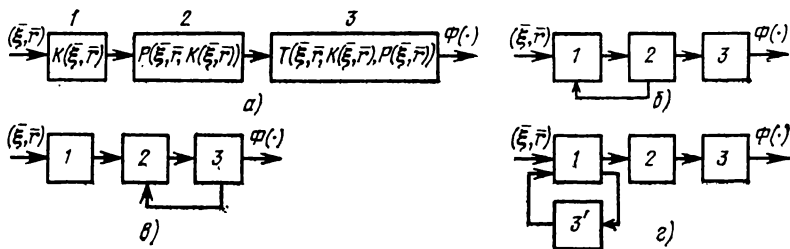


Рис. 3.1

ле размещения элементов критериями трассировки предусматривается приоритетная реализация оптимизируемых на этапе компоновки связей.

Рассмотрев схемы, представленные на рис. 3.1, сформулируем общую задачу. Имея заданные условия \bar{r} и случайные воздействия на систему $\bar{\xi}$, найти такую совокупность алгоритмов и критериев $\{K, P, T\}$, которая обеспечивала бы получение максимального конечного значения целевой функции $\Phi(\cdot)$. Таким образом, мы имеем задачу о выборе и принятии решения в условиях неопределенности.

Создание системы, реализующей общую схему, в которой на каждом шаге принятия промежуточного решения делался бы выбор алгоритмов после обработки входных данных этого шага в настоящее время практически неосуществимо, поэтому перед началом разработки методов и алгоритмов проводится анализ проектируемой на предприятии аппаратуры, изучаются технологические возможности производства с учетом перспективы развития, после чего создается ориентированная на определенный класс схем система алгоритмов. При достаточной гибкости создаваемого программного обеспечения возможна перестройка системы на ряд других классов схем и на другую технологию с небольшими затратами и без существенной потери качества.

Системы, работающие по схемам 3.1,б—3.1,г, при прочих равных условиях имеют преимущества перед системами, соответствующими схеме 3.1,а, так как оптимизация решения на каждом шаге отдельно не всегда дает в сумме оптимальное решение, осо-

бенно если на промежуточных этапах слабо учитывается конечный критерий. Указанный недостаток является в настоящее время причиной поиска связей между основными этапами конструкторского проектирования.

Траектория, соответствующая принятию решения, проходит в рассматриваемой задаче по образующим дискретной решетки n -мерного гиперкуба (точнее — в пределах n -мерного прямоугольного параллелепипеда); размерность n зависит от числа этапов проектирования, выбранных критериев и алгоритмов. Конечная цель проектирования каждой отдельной платы — реализация в виде печатных проводников максимального числа связей цепей.

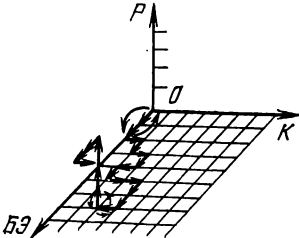


Рис. 3.2

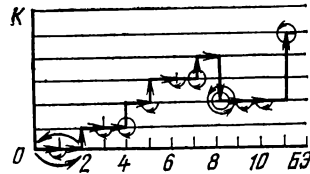


Рис. 3.3

Отсюда следует, что целевая функция $\Phi(\bar{\xi}, \bar{r}, K(\cdot), P(\cdot), T(\cdot))$, отражающая качество проектирования, принимает значения на конечном множестве целых положительных чисел. Для каждой платы число связей определяется как $\sum_{\{n_i\}} (i-1)n_i$, что соответ-

ствует $\max \Phi(\cdot)$, где n_i — число цепей, содержащих по i контактов. Число связей, подлежащих реализации, косвенно отражает сложность поведения целевой функции.

Процесс проектирования на отдельных этапах для наглядности удобно изобразить в трехмерном пространстве. Рассмотрим пространство траекторий, соответствующее этапам компоновки и размещения. Ось $BЭ$ соответствует выбранному на каждом шаге следующему базовому элементу. Ось K — скомпонованным модулям (не обязательно полностью), ось P — помещению очередного элемента на посадочное место (рис. 3.2).

Если компоновка и размещение ведется по схеме рис. 3.1,а, то можно рассматривать сначала проекции движения в плоскости $КOBЭ$ (рис. 3.3), а затем в плоскости POK (рис. 3.4).

На рис. 3.3 выбор очередного элемента соответствует одному шагу по оси $BЭ$. Если при выборе очередного элемента набор в новый корпус закончен окончательно или временно, то траектория поднимается на шаг вверх. Если же выбранный базовый элемент помещается в ранее заполняемый корпус, то траектория поднимается или опускается на соответствующий уровень. Циклические стрелки в данном примере около 1-го и 2-го $BЭ$ означа-

ют оптимизацию расположения этих элементов в пределах корпуса, а окружность со стрелкой вокруг скомпонованного корпуса [в данном примере вокруг точки (8.2)] соответствует оптимизации назначения эквивалентных выводов.

На рис. 3.4 изображена схема последовательной установки элементов. Более общий случай, которому соответствует процесс принятия решения, отраженный на рис. 3.2, показывает совместное решение задач компоновки и размещения. В этом случае возможен возврат в точку размещения корпуса, если устанавливаемый БЭ помещается в не полностью скомпонованный корпус. Из несложных рассуждений вытекает, что траектория движения,

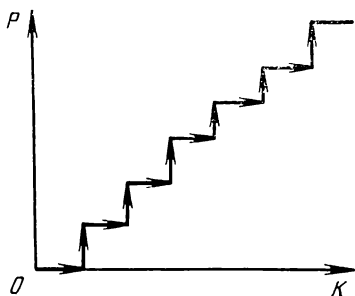


Рис. 3.4

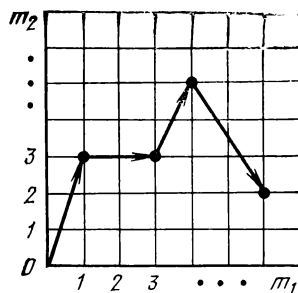


Рис. 3.5

соответствующая совместному решению задач компоновки и размещения, лежит в пределах четырехугольной пирамиды, основание которой лежит в плоскости, перпендикулярной оси БЭ и проходящей через точку, соответствующую числу БЭ. Основанием пирамиды является квадрат со стороной, также равной числу БЭ, что соответствует случаю, когда каждому такому элементу предоставляется один корпус. Вершиной пирамиды является начало координат. Выбор траектории движения предопределен используемыми алгоритмами и критериями. При работе системы в диалоговом режиме в конфликтных ситуациях на траекторию движения влияет также и пользователь. Заметим, что траектория движения отражает установку элемента на плату (БЭ в корпус), не определяя места установки, что резко повышает число возможных траекторий.

Ввиду того, что полный перебор немыслим, ставится задача выбора траекторий квазиоптимальных относительно некоторых принятых критериев. Задача облегчается тем, что в большинстве случаев входным параметрам (ξ, \bar{r}) соответствует множество траекторий, по которым возможно приблизиться к $\max \Phi(\cdot)$ либо достигнуть ее. Кроме того, тах достигается не в одной точке, как правило, а на целом множестве.

В качестве целевой функции всего процесса проектирования ПП можно выбрать число пересечений трасс, число поворотов трасс (что влияет на число переходных отверстий); длину от-

дельных проводников [77]. На практике применяется также композиция ряда критериев, не обязательно линейная.

Рассмотрим процесс размещения с определением конкретных посадочных мест для каждого элемента. Пусть задано N элементов и $m_1 \times m_2 = M$ посадочных мест на плате ($N \leq M$). Тогда процесс установки соответствует нахождению оптимальной траектории по выбранному критерию P из множества траекторий обхода N из M точек (рис. 3.5).

Проанализируем последовательную установку элементов на плату. При установке последовательным методом находят обычно элементы, наиболее связанные с соединителем [8, 41], и уста-

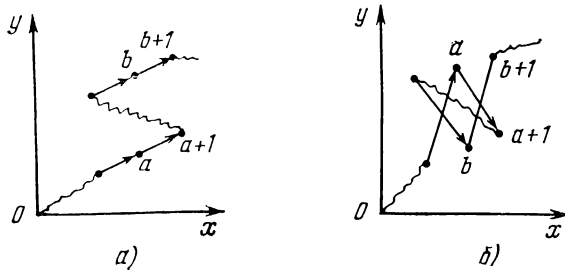


Рис. 3.6

навливают их около него (предположим, что на нашей диаграмме соединитель расположен слева и ориентирован вертикально). После установки всех элементов, наиболее связанных с соединителем, выбирают элемент, наиболее связанный с ранее установленными, и устанавливают его на одно из перспективных мест. Перспективными обычно являются места, соседние с уже установленными элементами. Выбор очередного элемента и перспективных мест зависит от принятых в алгоритмах размещения критериев. Описанная схема отражает процесс установки, предложенный в работе [8]. Реже решают обратную задачу: для выбранного свободного посадочного места подбирают элемент для размещения [44].

Подобным же образом действует и конструктор при размещении элементов, хотя он и ведет параллельно трассировку. Процесс размещения заканчивается после построения N векторов.

Процесс перестановки элементов соответствует изменению траектории обхода посадочных мест (рис. 3.6). Эта же диаграмма соответствует одному шагу алгоритма парных перестановок. Подобным будет и вид диаграммы групповых перестановок, но только более сложным. Заметим, что для алгоритма перестановок необходимо иметь начальное размещение, т. е. определить начальную траекторию. Таким образом, большинство известных практических алгоритмов размещения можно трактовать как выбор траектории оптимальным образом.

Проектирование опытным конструктором топологии представляется как одновременное решение всех трех задач с оптимизацией. Конструктор при проектировании топологии даже одной платы пользуется для решения одной и той же локальной задачи в разных ситуациях различными критериями, например в одном случае принимает решение минимизировать в данной области длину трасс, а в другом — при размещении на этой же плате минимизирует число пересечений трасс или число переходов со слоя на слой.

Словесное описание действий конструктора в терминах, определенных выше, могло бы выглядеть так. Имея заданные параметры $(\bar{\xi}, \bar{r})$, конструктор выбирает для компоновки некоторый БЭ из схемы, определяет для него корпус, устанавливает корпус на плату, реализует некоторые фрагменты. Все это он делает, имея в виду последующее определение корпуса для следующих элементов и дальнейшего размещения оставшихся корпусов, сочетая это с возможностями дальнейшей прокладки будущих фрагментов (при этом надо учесть, что возможность учета этих факторов будущего, глубины учета зависит от субъективных способностей конструктора к пространственному воображению; как и в шахматах, здесь есть свои «гроссмейстеры», способные просчитать варианты возможных будущих решений на большую глубину). Подобрав очередную БЭ в этот или новый корпус, конструктор опять максимизирует функцию $\Phi(\cdot)$, т. е. определяет ее траекторию при каждом своем действии. Ввиду того, что создание столь обширного математического обеспечения, когда мы можем варьировать на каждом шаге различными алгоритмами, требует больших материальных затрат, а полная алгоритмизация действий конструктора немыслима [34], приходится решать задачу поэтапно, отыскивая алгоритмизируемые связи между этапами, и главное, строить преемственные критерии всех этапов, так как несогласованный набор оптимальных алгоритмов каждого этапа в отдельности может и не дать удовлетворительных результатов.

Исходя из сказанного, при создании систем ставятся несколько упрощенные задачи, а именно: на каждом шаге принятия решения приблизиться к оптимальной траектории процесса, влияющей на проведение связей. При этом критерием качества компоновки и размещения является то, насколько точно удастся предопределить оптимальную траекторию $\Phi(\cdot)$ или к ней приблизиться.

В заключение отметим, что процессы перестановки элементов на плате, БЭ в корпусе и т. п. не являются выдуманными разработчиками САПР. В процессе проектирования конструктор снимает один, ранее установленный модуль, ставит на его место другой, меняет БЭ в корпусе и в различных корпусах местами, перекладывает проводники и т. п., что говорит о том, что при столь великом объеме информации и его постигает «проклятие» многомерности, и он не может сразу определить оптимальное

решение на каждом шаге последовательного построения и вынужден частично, а порой и полностью изменять свои старые решения. Разработчики САПР используют приемы перебора решений более полно как по упоминавшимся выше причинам, так и в связи с преимуществом ЭВМ перед человеком в возможности быстрого просмотра большого количества вариантов.

3.2. РАЗРАБОТКА МОДЕЛЕЙ ПОСАДОЧНЫХ МЕСТ И РАБОЧЕГО ПОЛЯ ТРАССИРОВКИ

Успех автоматизации любого процесса зависит, в числе прочих факторов, и от успешного выбора математических моделей, на основе которых разрабатываются и реализуются методы и алгоритмы.

Ниже мы опишем предлагаемые модели посадочных мест и рабочего поля платы, на основе которых был проведен анализ, выработаны критерии, методы и алгоритмы оптимизации задач компоновки, размещения и трассировки. В работах [78, 79] указывается, что наибольшее применение в автоматизированных системах нашли квазиоптимальные алгоритмы последовательного и итерационного размещения элементов.

При размещении элементов приблизительно равных размеров желательно использовать регулярную сетку посадочных мест элементов, чтобы монтажные дорожки были как можно более прямыми.

Недостатками последовательного размещения является жесткое закрепление устанавливаемых на позициях элементов, что отрицательно сказывается на качестве выбираемых позиций для последующих размещаемых элементов, а также определение наиболее приемлемого критерия выбора очередного элемента для установки.

С целью устранения первого из них предлагается расширить модель посадочных мест. Считаем заданной модель в виде регулярной сетки посадочных мест размером $m \times n$, где m — число мест по горизонтали, а n — по вертикали (рис. 3.7,а). Расширим нашу модель до размеров $M = (2m - 1) (2n - 1)$ (рис. 3.7,б). Учитывая, что первый устанавливаемый элемент должен иметь высший приоритет установки, он помещается в центр расширенной модели (на рисунке — кружок), что, как будет видно из дальнейших рассуждений, соответствует предоставлению ему в конечном счете любого места на плате. Последующие элементы могут устанавливаться на любые свободные места, но не выходить за пределы трафарета T размером $m \times n$, содержащего в обязательном порядке внутри себя в любом месте ранее установленные элементы. Разрешенные для установки места определяются следующим образом. Пусть трафарет T перемещается по расширенной модели так, что внутри него всегда остаются ранее установленные элементы, т. е. трафарет «блуждает» в некоторых преде-

лах, тогда за пределами его перемещений остаются места, на которые нельзя устанавливать элементы, а в пределах любого из положений трафарета на свободное место может быть установлен элемент.

В процессе установки число разрешенных для размещения очередного элемента посадочных мест быстро сокращается. Действительно, если до установки первого элемента на расширенной модели было $M_1 = (2m-1)(2n-1)$ свободных мест и после установки первого элемента число мест уменьшилось на единицу, то после установки второго элемента число свободных посадочных мест уменьшилось не менее чем на $2\min(m, n) - 1$.

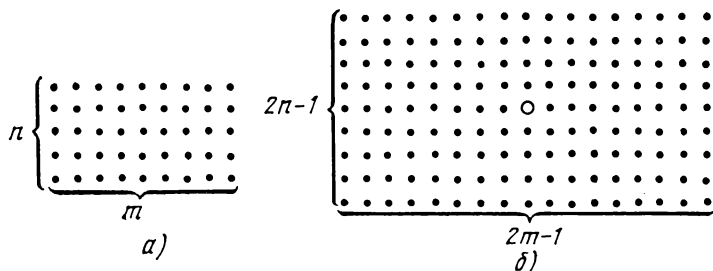


Рис. 3.7

Допустим, что установлено i элементов и разница между максимальной и минимальной абсциссами установленных элементов будет Δm_i , а разница между ординатами $-\Delta n_i$. В этом случае число допустимых для установки мест $(i+1)$ -го элемента определяется так:

$$M_{i+1} = (2m - (\Delta m_i + 1)) \cdot (2n - (\Delta n_i + 1)) - i.$$

На рис. 3.8,б приведен график скорости убывания числа посадочных мест для примера, изображенного на рис. 3.8,а.

Следует заметить, что при последовательной установке элементы, стоящие рядом в кортеже, как правило, связаны между собой, поэтому из допустимых мест установки можно выделять

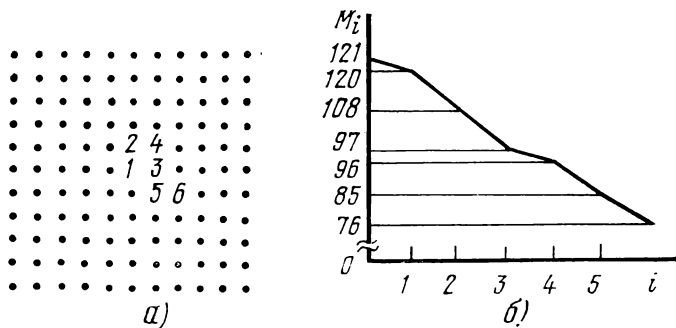


Рис. 3.8

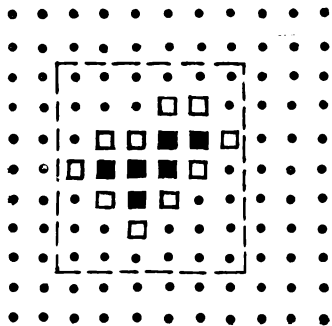


Рис. 3.9

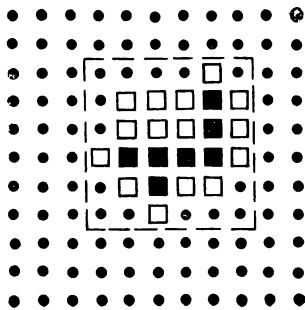


Рис. 3.10

«перспективные» места (светлые квадраты), расположенные рядом с уже занятыми (черные квадраты), сокращая тем самым число просматриваемых мест (рис. 3.9). Ограничение трафаретом допустимых мест показано на рисунке штриховыми линиями. Можно в качестве перспективных мест для установки очередного элемента выбирать посадочные места, расположенные не только рядом с уже установленными, но лежащие на пересечении образующих прямых, на которых расположены уже установленные элементы (рис. 3.10, обозначения те же, что на рис. 3.9).

Заметим также, что сетка посадочных мест должна быть регулярной, но не обязательно ортогональной, как это будет использоваться в дальнейшем при оптимизации расположения БЭ в корпусах микросхем, а перемещаемый трафарет не обязательно должен быть прямоугольным: он может иметь различные вырезы, содержать запрещенные для установки места. Такие требования труднее реализуются программно, но более соответствуют реальным условиям размещения.

Опишем теперь модель платы, которой будем пользоваться на этапе трассировки при проектировании плат с ИС. При разработке модели учитывалось, что лучшие результаты получаются, если корпуса на плате ориентированы одинаково, а также, что наиболее перспективными при проектировании считаются ИС с двумя параллельными рядами выводов.

Пусть на плате заданы контур, ограничивающий поле трассировки, и размещение ИС. Не те-

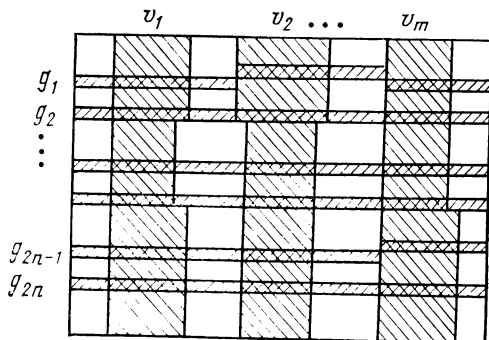


Рис. 3.11

ря общности, можно сказать, что ряды выводов ИС расположены горизонтально. Координаты посадочных мест образуют регулярную сетку. Координатой модуля на плате будем считать левый нижний угол минимального прямоугольника, описывающего выходы модуля. Размеры корпусов предполагаются приблизительно равными. На рис. 3.11 через v_i обозначено вертикальное граничное пространство, а через g_j — горизонтальное. Для трассировки плата разбивается ортогональными прямыми на множество дискретов согласно принятому шагу конструкторской сетки.

В дальнейшем будем пользоваться следующей терминологией. Под каналом будем понимать совокупность дискретов, по которым может проходить одна трасса в одном направлении (вертикальном или горизонтальном). Традиционное понятие канала в работах по данной тематике предполагает, что канал от своего начала и до конца имеет постоянную пропускную способность, если это не оговорено. В некоторых работах предлагается ввести понятие «канал переменной пропускной способности». В связи с этим считаем возможным ввести понятие ствола.

Ствол — совокупность каналов, ограниченная по направлению ствола рядами контактных площадок, а по ортогональному направлению — контуром поля трассировки. Пропускная способность ствола может меняться от сечения к сечению. Первый и последний стволы как горизонтального, так и вертикального направлений имеют одной из своих границ контур поля трассировки или контакты соединителя.

Этот термин аналогичен используемым при описании радиорелейных линий, где ствол означает совокупность каналов от одной станции до другой. Число каналов, образующих ствол, является пропускной способностью ствола. Канал может быть занят от одной станции до другой, а на остальных участках будет свободен. Аналогично на плате канал может быть занят на одном участке одним проводником, а на другом другим, а на остальных участках будет свободен для проведения еще не проложенных трасс.

Под границей стволов будем понимать множество отрезков прямых, отделяющих ствол от контактных площадок модулей; отрезки имеют свойство инерционности, т. е. горизонтальные отрезки, например, будут пересекать вертикальные стволы, а вертикальные отрезки — горизонтальные стволы. Под межствольным пространством (что эквивалентно граничному пространству) будем понимать пространство между границами двух соседних стволов. В отдельных случаях, что будет очевидным из контекста, граничное пространство будет отождествляться с границей. Модель поля трассировки приведена на рис. 3.11.

3.3. РАЗРАБОТКА СОГЛАСОВАННЫХ КРИТЕРИЕВ АЛГОРИТМОВ КОМПОНОВКИ, РАЗМЕЩЕНИЯ И ТРАССИРОВКИ

Качество решения задач, предшествующих трассировке, сильно влияет на результаты последней, а следовательно, и на конечный результат проектирования.

Анализ проектируемых печатных плат показал, что более половины проводников имеют простую конфигурацию. Однако необходимо учитывать, что при постоянном суммарном числе контактов в схеме трассировка решается проще в тех случаях, когда число цепей меньше. Следует иметь в виду, что цепи с большим числом контактов в процессе трассировки имеют больше возможностей подсоединения фрагментов проводимых к проведенным, что повышает вероятность реализации соединений.

Для выработки критерия выделим на плате некую выпуклую область Q , содержащую несколько модулей, и рассмотрим связи, находящиеся в ней. Если обозначить все связи, находящиеся в области, через S , то их можно представить в виде $S = S_1 \cup S_2 \cup S_3$, где S_1 — связи, проходящие через область Q , но не подсоединяющиеся ни к одному контакту области; S_2 — связи, соединяющие контакты области Q и полностью расположенные в Q ; S_3 — все остальные связи; эти связи подсоединяются как к контактам в Q , так и к контактам в \bar{Q} .

Так как перестановкой элементов в области Q и переназначением контактов в цепи мы не можем влиять на связи S_1 , то необходимо расположить элементы области и связи S_2 и S_3 области Q таким образом, чтобы связям S_1 были предоставлены наиболее благоприятные условия для реализации. При этом необходимо учитывать длину занятости каналов, число занятых каналов, число поворотов трасс. Учитывая, что на условия и качество трассировки влияет число пересечений кратчайших трасс, необходимо оптимизировать конфигурацию связей S_2 .

Так как число действующих контактов для каждой конкретной схемы является числом постоянным, то в первую очередь необходимо учитывать связи с меньшей степенью ребер $|X(U)|$ гиперграфа схемы $G(X, U; R)$, т. е. при интерпретации модуля как вершины x_i гиперграфа образовывать на этапе компоновки максимальное число ребер со степенью 1. На этапе же компоновки, учитывая эквивалентность БЭ и их входов, можно оптимизировать переназначение таким образом, чтобы ребра со степенью 1 реализовывались в виде трасс оптимальным образом (необходимо минимизировать число занимаемых такими ребрами каналов, длину занятости каналов, число пересечений). Таким образом, уже на этапе компоновки можно предопределить поведение траектории целевой функции $\Phi(\cdot)$ при попадании траектории в область расположения данного модуля.

На этапе размещения встает та же задача, но уже укрупненная. Для определения оптимального взаимного расположения

модулей необходимо рассмотреть поведение трасс, соединяющих различное число контактов. Рассмотрим сначала цепи, состоящие из двух контактов различных модулей (рис. 3.12,а).

Наименьшее отклонение от минимального связывающего дерева (имеется в виду дерево на ортогональной сетке) будет при соединении двух контактов на одной горизонтальной границе, а также если соединяются контакты в пределах вертикального граничного пространства и ветви при этом не пересекают горизонтальных границ (рис. 3.12,б).

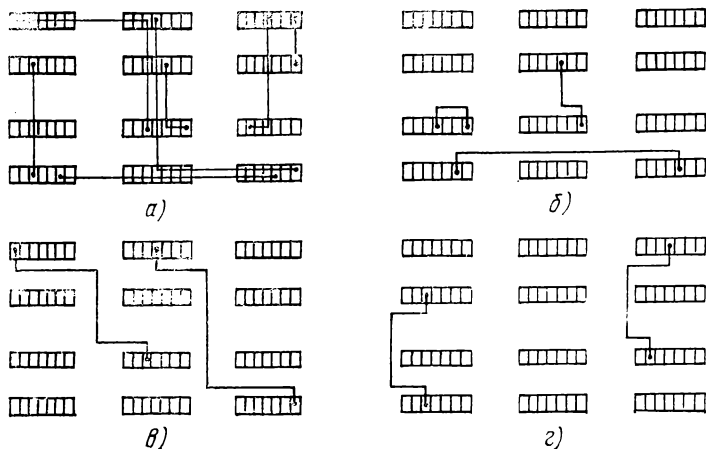


Рис. 3.12

Если контакты расположены на одной образующей ортогональной сетке, то в большинстве случаев для реализации такого соединения требуется не менее трех каналов, что порождает дополнительную загрузку стволов и увеличивает число переходных отверстий со слоя на слой (рис. 3.12,в).

Если дерево связывает два контакта из одного вертикального граничного пространства, которые разделены более чем одним горизонтальным стволом, то для реализации такой ветви печатным проводником требуется выход в ближайший вертикальный ствол, что в дополнение к предыдущему случаю увеличивает еще и длину связывающего дерева (рис. 3.12,г).

Из рассмотренных примеров следует, что оптимальным для деревьев из двух вершин является расположение контактов на границах одного горизонтального ствола. Для деревьев из трех вершин среднее удлинение трассы относительно минимального дерева будет меньше, так как вероятность расположения всех трех вершин в одном вертикальном граничном пространстве с разделением каждой друг от друга более чем одним горизонтальным стволом меньше, чем подобное расположение дерева из двух вершин, а в других случаях необходимое удлинение будет сделано в сторону ствола, ближайшего от вертикального гранич-

ного пространства, содержащего две вершины из трех, и направленного к третьей вершине. В этом случае при подсчете относительного удлинения разница (абсолютное удлинение) делится на три, а не на два, как в предыдущем рассматриваемом случае, хотя абсолютное удлинение остается таким же.

Подобные рассуждения для большого числа точек показывают, что среднее удлинение трасс относительно построенного без учета препятствий связывающего дерева с возрастанием числа контактов в цепи уменьшается. Соответствующий график приведен на рис. 3.13, где ΔL — увеличение длины построенного дерева, а L — длина первоначально построенного квазиминимального дерева. Из указанной закономерности получаем следующее правило.

Выделив максимально связанные группы элементов, оптимизируем взаимное расположение элементов внутри группы на плате в области Q с учетом рассмотренных выше критериев распределения связей S_2 , отдавая при этом предпочтение коротким связям; учитываем при этом и расположение связей S_3 , что сводится

к предпочтительному расположению контактов этих связей на границе области Q расположения группы. Этим самым мы частично предопределяем, как и на этапе компоновки, траекторию целевой функции $\Phi(\cdot)$ при попадании в область расположения рассматриваемой группы моделей.

С целью большей согласованности критериев оптимизации на этапах компоновки и размещения с трассировкой реализация связей в стволе должна проводиться от меньшей ветви к большей, что согласуется с требованием меньше блокировать контакты для последующих прокладываемых соединений, а также с требованием минимизации числа переходных отверстий, что вытекает из теоремы Жордана о разделении области на две несвязные части [80]. Подробнее на последнем вопросе остановимся в гл. 6.

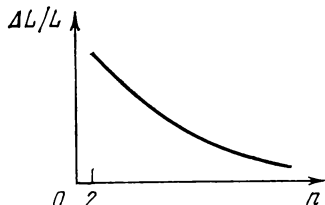


Рис. 3.13

3.4. РАЗРАБОТКА КРИТЕРИЕВ И АЛГОРИТМОВ РАЗБИЕНИЯ МНОЖЕСТВА ВЕРШИН ГИПЕРГРАФА НА ГРУППЫ МАКСИМАЛЬНО СВЯЗАННЫХ ЭЛЕМЕНТОВ

Проблема разбиения электрической или функциональной схемы устройства на составные части, связанные между собой минимальным числом соединений, возникает на различных этапах проектирования РЭА.

В настоящее время не известны методы решения задачи разбиения, которые давали бы оптимальное решение за приемлемое время счета [31]. Одним из недостатков существующих алгоритмов является разбиение множества вершин на заранее опреде-

ленное число вершин в подграфах. При этом не всегда можно получить оптимальные результаты.

Описание электрической схемы устройства графом не является адекватным. Как уже отмечалось ранее, при таком представлении схемы возникают трудности при описании цепей, содержащих более двух контактов. В большинстве работ советских авторов эта проблема решалась введением полных графов, что вносит избыточную информацию и в дальнейшем в некоторых случаях приводит к худшим решениям [81]. О недостатках при описании схемы устройства графом указывают многие авторы, например [18, 82]. В работе [22] для решения указанной проблемы вводятся дополнительные вершины, соответствующие цепям, и схема представляется графом Кенига. Аналогичное представление схемы можно найти в работе [2]. По вопросу применения графов Кенига следует привести слова одного из ведущих советских специалистов в области теории графов А. А. Зыкова: «Не только любой конечный гиперграф допускает кенигово представление, но и, наоборот, каждый граф Кенига является представлением $K(H)$ некоторого конечного гиперграфа H и определяет его однозначно с точностью до изоморфизма. Было бы, однако, ошибкой отказаться на этом основании от построения теории гиперграфов, заменив ее изучением одних лишь графов Кенига, ибо многие положения о гиперграфах приобретают в терминах кениговых представлений искусственный и громоздкий характер, затуманивающий существо дела...» [83].

С большей точностью схема устройства описывается гиперграфом [32]. О целесообразности и преимуществах использования гиперграфов можно найти сведения в работах [18, 81].

Пусть электрическая схема устройства описывается гиперграфом $H = (X, U, R)$. (В дальнейшем описании будем использовать обозначения и определения работы [83], отступления от них будут оговорены.)

Учитывая специфику решаемой задачи, при описании схемы из множества ребер U удаляются петли. Пространство возможных решений задачи разбиения содержится в классе $S(X)$, элементы которого, как мы ранее условились, будем обозначать рукописными прописными буквами \mathcal{X}_i . Напомним, что под $|U(\mathcal{X}_i)|$ будем понимать степень элемента \mathcal{X}_i класса S . При рассмотрении элемента \mathcal{X}_i множество ребер гиперграфа можно записать в виде трех непересекающихся подмножеств:

$$U = U_1 \cup U_2 \cup U_3,$$

где U_1, U_2, U_3 — соответственно множество ребер, инцидентных вершинам из \mathcal{X}_i ; вершинам из \mathcal{X}_i и $X \setminus \mathcal{X}_i$ одновременно; вершинам только из $X \setminus \mathcal{X}_i$.

В этом случае справедливо равенство $|U(\mathcal{X}_i)| = |U_1(\mathcal{X}_i)| + |U_2(\mathcal{X}_i)|$.

Один из возможных вариантов решения проблемы разбиения был предложен Люцио и Сэми [22]. Приведем основные опреде-

ления и результаты из работы этих авторов, на которые в дальнейшем будем опираться.

Определение 1. Группа \mathcal{X}_i называется минимальной, если для $\forall \mathcal{X}_j \subset \mathcal{X}_i$ справедливо

$$|U_2(\mathcal{X}_j)| > |U_2(\mathcal{X}_i)|, \quad (3.1)$$

элементы множества X считаются минимальными группами по определению.

Теорема 1. Если \mathcal{X}_i и \mathcal{X}_j — минимальные группы, $\mathcal{X}_i \not\subset \mathcal{X}_j$ и $\mathcal{X}_j \not\subset \mathcal{X}_i$, то $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$.

Теорема 2. Если \mathcal{X}_i и \mathcal{X}_j — минимальные непересекающиеся группы, $\mathcal{X}_k = \mathcal{X}_i \cup \mathcal{X}_j$ и $|U_2(\mathcal{X}_k)| < \min(|U_2(\mathcal{X}_i)|, |U_2(\mathcal{X}_j)|)$, то \mathcal{X}_k — минимальная группа. Перевод алгоритма образования минимальных групп можно найти в книге [81].

Докажем ряд теорем, позволяющих существенно сократить время работы алгоритма.

Теорема 3. Если \mathcal{X}_i — минимальная группа, то $\forall x_k \in \mathcal{X}_i \exists$ ребро u , инцидентное только вершинам из \mathcal{X}_i .

Доказательство. Допустим, что существует вершина $x_k \in \mathcal{X}_i$, такая, что все инцидентные ей ребра инцидентны и вершинам из $X \setminus \mathcal{X}_i$, тогда $|U_2(\mathcal{X}_i)| \geq |U(x_k)|$, что противоречит минимальности группы \mathcal{X}_i .

Теорема 4. Если \mathcal{X}_i — минимальная группа в гиперграфе $H = (X, U; R)$, то полнореберный подграф $H' = (\mathcal{X}_i, U; R)$ будет связным.

Доказательство. Допустим, что H' не связан, тогда существует, по крайней мере, две компоненты связности, и \mathcal{X}_i разлагается на два непустых непересекающихся подмножества \mathcal{X}_{i1} и \mathcal{X}_{i2} , откуда следует $|U_2(\mathcal{X}_i)| = |U_2(\mathcal{X}_{i1})| + |U_2(\mathcal{X}_{i2})|$, т. е. $|U_2(\mathcal{X}_i)| \geq |U_2(\mathcal{X}_{i1})|, |U_2(\mathcal{X}_{i2})|$, что противоречит минимальности группы \mathcal{X}_i .

Теорема 5. Если \mathcal{X}_i — минимальная группа в гиперграфе $H = (X, U; R)$, то усеченный подграф $H'' = (\mathcal{X}_i, U''; R)$ связан.

Доказательство. Согласно теореме 4 полнореберный подграф $H' = (\mathcal{X}_i, U; R)$ связан. Удаление из H' , как и из H , одного любого ребра, инцидентного вершине из $X \setminus \mathcal{X}_i$, не нарушает равенства (3.1), следовательно, группа \mathcal{X}_i остается минимальной, и соответствующий полнореберный подграф H' остается связным согласно теореме 4. Так как гиперграф H конечен, то последовательное удаление ребер, инцидентных вершинам из $X \setminus \mathcal{X}_i$, обращает правую часть неравенства (3.1) в ноль, что соответствует построению усеченного подграфа, который остается связным, что и требовалось доказать.

Из хода доказательства теоремы 5 следует, что при образовании минимальной группы из j элементов определяющими могут быть только те ребра, степень которых не превышает j .

Теоремы 3—5 позволяют определить очередную вершину для проверки группы на минимальность и глубину перебора по реб-

рам гиперграфа для данной группы. Для этого целесообразно расположить ребра для просмотра по возрастанию их степеней.

Практическая реализация выявила следующие недостатки алгоритма Люцио и Сэми: в реальных электрических и функциональных схемах устройств минимальные группы не всегда образуются (за исключением тривиальных); если минимальные группы и образуются, то они, как правило, состоят из большого количества элементов; время счета даже для 40 вершин остается неприемлемо большим, несмотря на применение теорем 3—5.

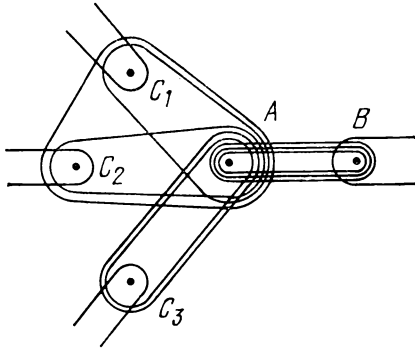


Рис. 3.14

Следующий пример, приведенный на рис. 3.14, показывает еще один недостаток рассмотренного алгоритма. Из практических соображений вершины A и B должны быть объединены, однако нетрудно видеть, что $|U(A)| < 2|U(B)|$, следовательно, эти вершины не будут объединены, по крайней мере на первом этапе. В то же время вершина B отдает в предполагаемую группу (A, B) 75% своих связей, и если она войдет когда-нибудь в минимальную группу, то только с вершиной A . Вершина A отдает в группу (A, B) только 37,5%

своих связей, поэтому получаем $\rho(A, B) = 5/8 + 1/4 = 7/8$, т. е. вершина B притягивается к вершине A , но внешние связи вершины A не дают ей притянуться к вершине B .

С целью устранения указанных недостатков предлагается следующий критерий максимальной связности [84]. Пусть имеется группа \mathcal{X}_i , состоящая из n вершин. Не теряя общности, можно предположить, что это вершины x_1, x_2, \dots, x_n . Через $k_j = k_j(x_j)$ обозначим число ребер, которые инцидентны в \mathcal{X}_i только вершине x_j . Коэффициент связности для \mathcal{X}_i определим следующим образом:

$$\rho(\mathcal{X}_i) = \sum_{j=1}^n \frac{k_j}{|U(x_j)|}.$$

Определение. Группу считаем максимально связной, если

$$\rho(\mathcal{X}_i) < 1. \quad (3.2)$$

Данное требование слабее, чем требование минимальности по Люцио и Сэми, но не противоречит последнему, что показывает следующая теорема.

Теорема 6. Если $\mathcal{X}_i = \{x_1, x_2, \dots, x_n\}$ — минимальная группа, то $\rho(\mathcal{X}_i) < 1$.

Доказательство. Из определения минимальности группы

\mathcal{X}_i следует справедливость следующих неравенств:

$$\begin{aligned} |U(x_1)| &> |U_2(\mathcal{X}_i)|, \\ |U(x_2)| &> |U_2(\mathcal{X}_i)|, \\ &\vdots \\ |U(x_n)| &> |U_2(\mathcal{X}_i)|, \end{aligned}$$

на основании которых получаем

$$\frac{k_1}{|U(x_1)|} + \frac{k_2}{|U(x_2)|} + \dots + \frac{k_n}{|U(x_n)|} < \frac{k_1 + k_2 + \dots + k_n}{|U_2(\mathcal{X}_i)|},$$

но $|U_2(\mathcal{X}')| \geq k_1 + k_2 + \dots + k_n$, и левая часть неравенства — коэффициент связности, откуда следует:

$$\rho(\mathcal{X}_i) < \frac{k_1 + k_2 + \dots + k_n}{|U_2(\mathcal{X}_i)|} < 1,$$

что и требовалось доказать.

На основании критерия связности (3.2) и доказанных теорем 3—5 разработаны следующие алгоритмы.

Алгоритм 1.

1. Упорядочиваем множество ребер по возрастанию $|X(u)|$.
2. Полагаем j равным двум.
3. Проверяем на максимальную связность непересекающиеся группы из j элементов по ребрам со степенью $|X(u)| \leq j$, используя теоремы 3—5 и критерий (3.2).
4. Если группы по j элементов не образовались, то переходим к пункту 9.
5. Заносим в список результатов образовавшиеся группы $\{\mathcal{X}_i\}$.
6. Гомоморфно отображаем обрабатываемый гиперграф H следующим образом: элементы, не вошедшие в группы, отображаются сами в себя; группа отображается в элемент с минимальным номером элемента в группе; если число вершин вновь образованного гиперграфа равно 2, то переходим к пункту 11.
7. Удаляем образовавшиеся петли.
8. Переходим к пункту 1.
9. Увеличиваем значение j на 1.
10. Если j меньше числа вершин в гиперграфе, обрабатываемом в данный момент, то переходим к пункту 3.
11. Конец.

Примечание. При повторном рассмотрении на связность после образования максимально связных групп проверка идет по ребрам полнороберного подграфа, вершинами которого служат в том числе и образовавшиеся максимально связные группы.

Для выделения связных групп с ограниченным числом вершин (допустим, m), как это бывает удобно, например, при некоторых подходах к размещению элементов на плате, а также для сокращения времени счета, реализована следующая модификация алгоритма.

Алгоритм 2.

1. Упорядочиваем множество ребер по возрастанию $|X(u)|$.

2. Полагаем j равным двум.

3. Проверяем на максимальную связность непересекающиеся группы из j элементов по ребрам со степенью $|X(u)| \leq j$, используя теоремы 3—5 и критерий (3.2); одновременно определяем коэффициент связности при каждой проверке, сравниваем с предыдущим и запоминаем минимальное из этих значений — ρ_{\min} .

4. Если группы по j элементов образовались, то переходим к пункту 8.

5. Если $j < m$, то увеличиваем значение j на 1 и переходим к пункту 6, иначе переходим к пункту 7.

6. Если j равно числу вершин обрабатываемого в данный момент гиперграфа, то переходим к пункту 13, иначе переходим к пункту 3.

7. Образовываем связные группы, соответствующие значению ρ_{\min} .

8. Заносим в список результатов образовавшиеся группы $\{\mathcal{R}_i\}$.

9. Гомоморфно отображаем обрабатываемый гиперграф, как и в алгоритме 1.

10. Если число вершин образовавшегося гиперграфа равно 2, то переходим к пункту 13.

11. Удаляем образовавшиеся петли.

12. Переходим к пункту 1.

13. Конец.

1 4 0 0	1 4 0 0	1 4 0 0	1 4 0 0	1 4 0	1 5
1 2 0 0	7 10 0 0	4 5 0 0	4 5 0 0	4 5 0	5 6
7 9 0 0	4 5 0 0	6 7 0 0	5 6 0 0	5 6 0	1 6
7 10 0 0	6 7 0 0	5 6 0 0	3 4 0 0	3 4 0	e)
4 5 0 0	5 6 0 0	3 4 0 0	6 8 0 0	1 3 6	
10 11 0 0	3 4 0 0	6 7 8 0	1 3 6 8	д)	
10 11 0 0	6 7 8 0	1 3 6 8	г)		
6 7 0 0	1 3 6 8	в)			
5 6 0 0	б)				
3 4 0 0					
1 5 0 0					
6 7 8 0					
1 3 6 8					
а)					

Рис. 3.15

На рис. 3.15 и рис. 3.16 показан пример работы алгоритма 1. В матрице на рис. 3.15 строка соответствует ребру гиперграфа, каждое число в строке — номеру вершины, входящей в ребро. При первом образовании связных групп для трех пар (1, 2), (7, 9) и (10, 11) получилось $\rho < 1$. Заметим, что группа (10, 11) является минимальной по Лючио и Сэми. На следующем шаге образовалась группа из элементов (7, 10) и т. д. На рис. 3.16 наглядно продемонстрирован пункт 6 алгоритма 1, а также представлен результат в виде дерева. Для этого же примера приведем результаты доказанных теорем. Если придерживаться методики Лючио и Сэми, то для этого примера при первой проверке

необходимо сделать просмотр 55 пар, а пользуясь теоремами 3—5, достаточно просмотреть 11 пар элементов.

Практическое применение находит также следующий алгоритм выделения непересекающихся групп не более чем из t элементов с минимальным ρ .

Алгоритм 3.

1. Упорядочиваем множество ребер по возрастанию $|X(u)|$.
2. Полагаем j равным двум.

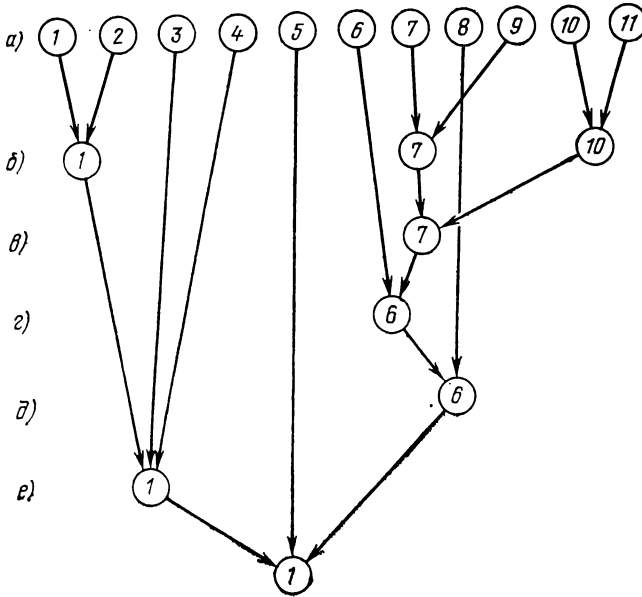


Рис. 3.16

3. Проверяем на максимальную связность непересекающиеся группы из j элементов по ребрам со степенью $|X(u)| \leq j$, используя теоремы 3—5 и критерий (3.2); одновременно определяем коэффициент связности при каждой проверке, сравнивая с предыдущим, и запоминаем минимальное из этих значений — ρ_{\min} .

4. Если группы по j элементов образовались, то переходим к пункту 8.

5. Если $j < t$, то увеличиваем значение j на 1 и переходим к пункту 6, иначе переходим к пункту 7.

6. Если j равно числу вершин обрабатываемого в данный момент гиперграфа, то переходим к пункту 13, иначе переходим к пункту 3.

7. Образовываем связанные группы, соответствующие значению ρ_{\min} .

8. Заносим в список результатов образовавшиеся группы $\{S_i\}$.

9. Для дальнейшей работы алгоритма из обрабатываемого в данный момент гиперграфа образуем полнореберный подграф (подгиперграф) удалением вершин, вошедших в образовавшиеся группы.

10. Если число вершин образовавшегося гиперграфа равно 2, то переходим к пункту 13.

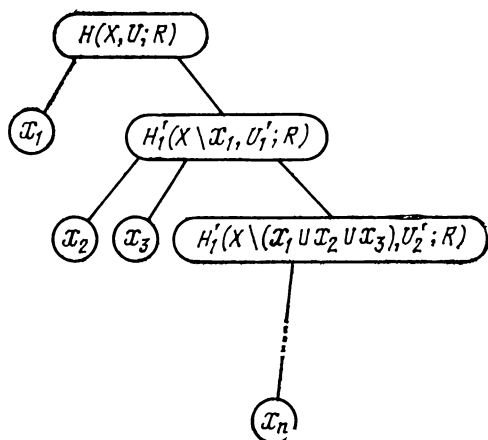


Рис. 3.17

11. Удаляем из подграфа дуги со степенью 0 и 1; в дальнейшем работаем с этим подграфом как с гиперграфом.

12. Переходим к пункту 1.

13. **Конец.**

На рис. 3.17 показан пример результата работы алгоритма.

3.5. ПОСТРОЕНИЕ КВАЗИОПТИМАЛЬНОГО ДЕРЕВА НА ОРТОГОНАЛЬНОЙ СЕТКЕ

При некоторых подходах к трассировке печатного монтажа, при трассировке проволочного монтажа и в ряде других случаев возникает необходимость построения связывающего дерева минимальной длины.

В евклидовой геометрии наиболее известны два алгоритма построения минимального связывающего дерева. Краскалом [85] предложен алгоритм, основанный на построении полного графа с последующим удалением на каждом шаге самого длинного ребра из множества ребер графа с требованием сохранения связности. Процесс продолжается до тех пор, пока не останется связный граф без циклов, т. е. дерево. Алгоритм, предложенный Примом, можно назвать алгоритмом противоположного действия. Пусть имеется n изолированных вершин. На каждом очередном шаге строится ребро минимальной длины из тех ребер, которые сокращают число компонент связности на единицу. Когда будет построено с таким условием $n-1$ ребро, процесс заканчивается [86].

Задача построения связывающего дерева минимальной длины с введением дополнительных точек известна как задача Штейнера.

Особый интерес для разработчиков систем автоматизированного конструирования радиоэлектронной аппаратуры представляет задача построения дерева

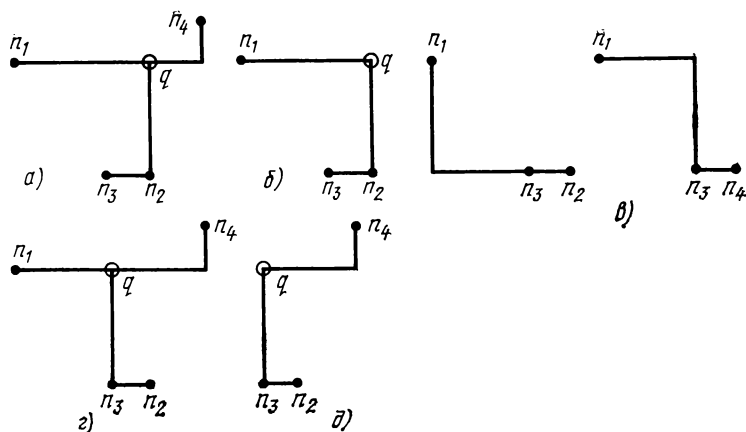


Рис. 3.18

Штейнера на ортогональной сетке. Из-за неоднозначности в общем случае минимального пути между двумя точками на ортогональной сетке алгоритмы Краскала и Прима не позволяют получить минимальное дерево. По этой же причине получение точного решения для произвольного числа вершин в дереве пока затруднительно. Известен точный алгоритм для $n \leq 5$ [87]. Заметим, что в работе [88] приведен неправильный перевод теоремы из статьи [87] для $n=3$.

Ряд теорем, позволяющих проводить поиск квазиоптимальных деревьев Штейнера на ортогональной сетке, приведен в монографии [31]. Однако и в этой книге приведена ошибочная лемма под номером 7.8, утверждающая, что если G является деревом Штейнера для p вершин n_1, n_2, \dots, n_p , если при этом G' содержит t вершин и является связным подграфом графа G , то G' является деревом Штейнера для этих t вершин. Опровергающий пример аналогичен рис. 7.38,д из этой же книги.

Пусть у нас имеется дерево Штейнера, соединяющее четыре вершины и имеющее точку Штейнера q (рис. 3.18,а). Связный подграф $G': n_1-q-n_2-n_3$ не является деревом Штейнера (рис. 3.18,б), так как согласно теореме Ханана возможны два варианта дерева Штейнера (рис. 3.18,в), что в данном случае является очевидным. При другом построении дерева Штейнера для графа G (рис. 3.18,г) получаем, что подграф G'' , содержащий вершины n_2, n_3, n_4 (рис. 3.18,д), не является минимальным связывающим деревом. Возможно и здесь ошибка переводчика.

В работе [2] анализируется применение булевых функций для построения деревьев Штейнера; здесь тоже отмечается неэффективность метода для больших n .

Построение дерева Штейнера на ортогональной сетке на основании алгоритма Прима описывается в работах [89, 90]. Получающиеся деревья — квазиоптимальные, на число вершин ограничений не накладывается. Применение алгоритма Прима малоэффективно из-за относительно большого объема вычислений, хотя и получаются приближающиеся к оптимальным результаты.

ник Π_k может вырождаться в точку (рис. 3.21,а) или в отрезок (рис. 3.21,б), либо остаться прямоугольником (рис. 3.21,в).

После построения системы описывающих прямоугольников переходим к непосредственному построению дерева. На границе B_1 берем любую вершину из N . Пусть это будет n_1 . Находим расстояние $\rho_1(n_1, n \in (B \cap N) \setminus n_1)$ и $\rho_2(n_1, B_2)$ (здесь считаем, что $B_2 \neq \emptyset$). Расстояние от точки до множества определяется как расстояние от точки до ближайшей точки из множества в данном метрическом пространстве. Расстояние от фрагмента до множества определяется как минимальное расстояние между точками этого фрагмента и множества.

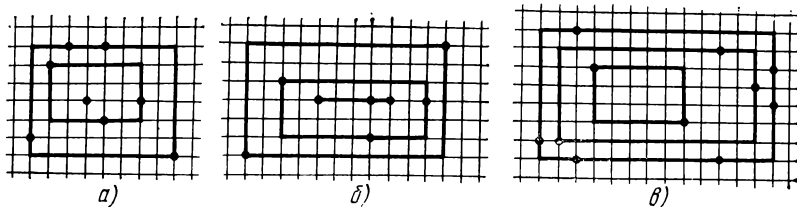


Рис. 3.21

На следующем шаге анализируем соотношение ρ_1 и ρ_2 . Если $\rho_1 > \rho_2$, то вершина n_1 соединяется с ближайшим узлом на B_2 . Если в этом узле нет вершины, принадлежащей N , то фиксируется дополнительная вершина.

Если $\rho_1 \leq \rho_2$, то n_1 соединяется с ближайшей вершиной на B_1 . Предположим, что это вершина n_2 . Если вершины n_1 и n_2 не лежат на одной прямой, то они соединяются двумя отрезками, обращенными углом к следующему (меньшему) прямоугольнику; если обе вершины лежат на одной прямой образующей сети, то они соединяются отрезком прямой (рис. 3.22).

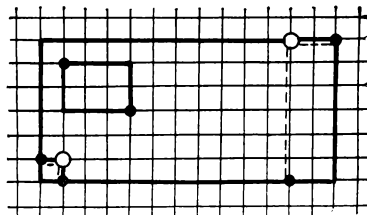


Рис. 3.22

После построения фрагмента определяется расстояние от него до вершин $(B_1 \setminus (n_1 n_2) \cap N)$ и $(B_2 \cap N)$. Далее повторяется описанный выше процесс. Если построение окончилось соединением с периметром прямоугольника B_2 , то переходим к следующей точке, лежащей на периметре B_1 , но не принадлежащей построенным фрагментам. Возможен случай, когда все точки на B_1 соединены между собой. Тогда построенная ветвь соединяется с B_2 кратчайшим отрезком. Если все вершины периметра B_1 соединены с B_2 непосредственно или через другие вершины, для продолжения построения переходим к прямоугольнику Π_2 , для которого соединяем как основные, так и образовавшиеся дополнительные вершины.

Таким образом процесс продолжается до последнего описывающего прямоугольника Π_k , на котором вершины соединяются между собой аналогичным образом.

После построения дерева проводим анализ дополнительных вершин. Напомним, что дополнительными вершинами считаются не только точки соединения трех или четырех ребер, где нет основных вершин, но и точки пересечения с периметрами прямоугольников, а также точки поворота.

Анализ дополнительных вершин заключается в следующем. Берем произвольную вершину со степенью 1. Пусть это будет вершина n_1 . Если она соединена ребром с основной вершиной либо с дополнительной со степенью более 2, то исключаем ее из рассмотрения и переходим к следующей вершине. Если вершина n_1 соединена с дополнительной вершиной со степенью 2, то считаем длину пути ρ_{n_1} в указанной метрике от вершины n_1 до первой встретившейся основной вершины либо до дополнительной вершины со степенью более 2, двигаясь при этом по ребрам построенного дерева. Сравниваем длину пути с расстоянием ρ'_{n_1} от n_1 до ближайшей вершины, лежащей на том же прямоугольнике, что и n_1 . Если $\rho_{n_1} > \rho'_{n_1}$, то в дереве путь, соответствующий ρ_{n_1} , стирается, а строится ребро, соответствующее расстоянию ρ'_{n_1} . Заметим, что связность дерева при этом не нарушается.

Последний этап — выравнивание углов. Если дополнительная вершина имеет общее ребро с другой дополнительной вершиной, то она переводится в диагонально противоположный угол прямоугольника, построенного на двух соседних вершинах (рис. 3.23); A и B — дополнительные вершины, которые после анализа заменяются одной дополнительной вершиной C . Если дополнительная вершина имеет степень более 2, то в диагонально противоположный угол переводится соседняя дополнительная вершина (рис. 3.24, обозначения те же, что на рис. 3.23). Нетрудно заметить, что построенное таким образом дерево полностью лежит на образующих, проходящих через основные точки-вершины.

При реализации процедуры массив вершин удобно переформировать в два: первый — упорядоченный по координате x , второй — по координате y . Пусть вершины в массивах расположены в порядке возрастания x и y соответственно. Тогда первые точки списка, допустим, упорядоченного по x , с равными абсциссами определяют одну вертикальную сторону прямоугольника Π_1 , а последние точки этого же списка с равными абсциссами — вторую вер-

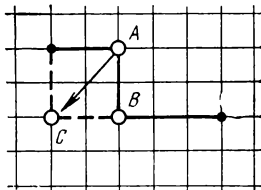


Рис. 3.23

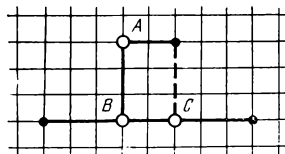
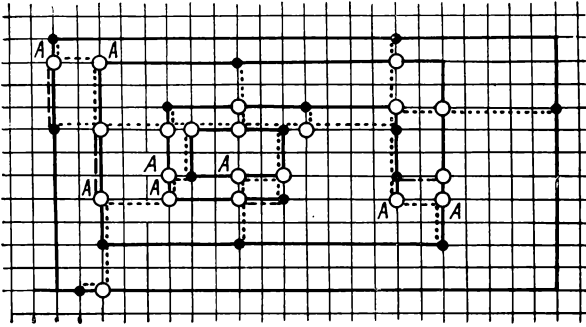
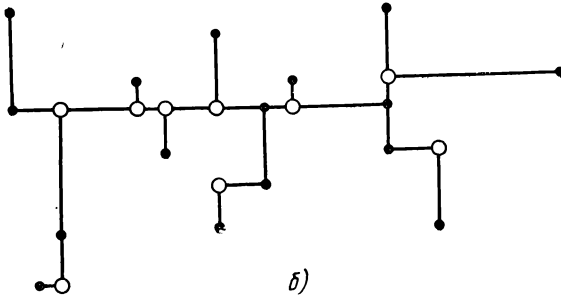


Рис. 3.24



a)



b)

Рис. 3.25

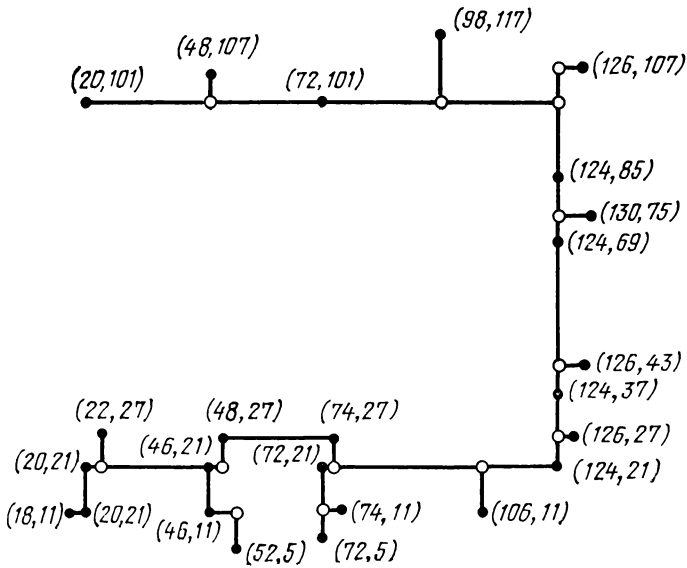


Рис. 3.26

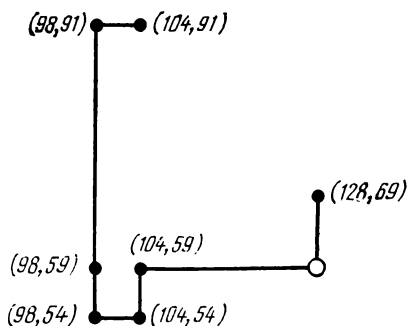


Рис. 3.27

тикальную сторону прямоугольника Π_1 . Аналогично определяют по списку, упорядоченному по y , горизонтальные стороны прямоугольника Π_1 . Выбираем эти точки из списка, а по оставшимся аналогично определяем стороны второго прямоугольника и т. д. На рис. 3.25,а показана последовательность реализации алгоритма, а на рис. 3.25,б — построенное дерево (точками обозначено первоначально построенное дерево, штриховыми и штрихпунктирными линиями—ветви со-

ответственно после 1-го и 2-го анализа; вершины, обозначенные A , сняты в результате анализов).

На рис. 3.26 приведен пример деревьев с большим числом вершин, полученных по описанной выше процедуре. При анализе полученных результатов для числа вершин менее семи неоптимальных деревьев на реальных платах не получалось. Для большого числа вершин за счет промежуточных построений получались неоптимальные деревья (рис. 3.27). Анализ построенных деревьев показал, что для числа вершин более семи суммарная длина построенного дерева в отдельных случаях больше длины минимального связывающего дерева, но не более чем на 10%, что не хуже известных по публикациям результатов.

Время работы программ, реализующих описанную процедуру, вполне приемлемо. Например, для схемы, содержащей 115 цепей общим объемом около 420 контактов, время построения всего леса на ЭВМ ЕС-1022 составляет около 90 с.

Глава 4

РЕШЕНИЕ ЗАДАЧИ КОМПОНОВКИ С ИСПОЛЬЗОВАНИЕМ СООТНОШЕНИЙ ЭКВИВАЛЕНТНОСТИ

4.1. РАСПРЕДЕЛЕНИЕ БАЗОВЫХ ЭЛЕМЕНТОВ ЛОГИЧЕСКОЙ СХЕМЫ ПО КОРПУСАМ МИКРОСХЕМ

Первым этапом технического проектирования печатных плат (ПП) является задача распределения базовых элементов (БЭ) логической схемы по корпусам микросхем (ИС). Ее можно решать в такой последовательности:

определить оптимальный набор ИС для проектируемой схемы из числа разрешенных к применению в данной разработке или на данном предприятии;

распределить БЭ по корпусам ИС таким образом, чтобы минимизировать число межмодульных связей;

используя соотношения эквивалентности БЭ и их входов, решить задачу оптимального назначения для каждой ИС по согласованному с трассировкой критерию.

Кроме указанных теоретических задач, необходимо предусмотреть ряд требований, позволяющих обеспечить высокие эксплуатационные качества: система должна легко перестраиваться либо допускать использование самых различных компоуемых единиц, разрешенных в данной разработке или на данном предприятии. При решении в САПР задачи компоновки появляется возможность включить элементы функционального контроля, что также необходимо использовать вместе с более полным контролем предыдущих этапов; расширяются возможности получения конструкторской документации (КД), а также появляется перспектива коррекции этой документации после этапов размещения и трассировки. Конечно результатом компоновки должна быть информация, используемая для дальнейшего проектирования, желательно без предварительных коренных преобразований. К выходной информации предъявляется при этом требование минимального увеличения объема относительно информации, используемой на этапах размещения и трассировки.

При макетировании аппаратуры определение оптимального набора ИС для компоновки необязательно. При разработке аппаратуры для серийного производства эту задачу необходимо решать, так как количество используемых ИС отражается на стоимости изделий. Примером получения оптимального набора может служить использование незадействованных в корпусе БЭ с более широкими возможностями, например базовый элемент 4И-НЕ для выполнения им функции 3И-НЕ или 2И-НЕ. При этом сокращается число используемых корпусов. Решение данной задачи нашло достаточное отражение в литературе, поэтому на ней здесь мы останавливаться не будем.

Распределение БЭ логической схемы по корпусам ИС предлагается делать по следующей методике. Пусть логическая схема устройства описывается гиперграфом $G=(X, U; R)$, где X — множество вершин гиперграфа, соответствующих БЭ и прочим электро-радиоизделиям (ЭРИ), а U — ребра гиперграфа, соответствующие электрическим связям. Каждое ребро содержит эквипотенциальные контакты схемы. Множество вершин можно представить следующим образом: $X = \bigcup_{i,j} X_{ij}$, где множество X_0 включает

в себя вершины, соответствующие ЭРИ, как не требующие компоновки, а также ИС, скомпонованные по указанию разработчика; вершины x_{ij} соответствуют БЭ, выполняющим некую i -ю логическую функцию; j изменяется от 1 до k_i , где k_i — число элементов

i -й логической функции. Множество ребер гиперграфа U можно записать в виде $U = U_0 \cup U_1$, где U_0 — множество ребер со степенью 1, т. е. $|X(u; u \in U_0)| = 1$, а U_1 — множество ребер со свойством $|X(u; u \in U_1)| > 1$. Ребер со степенью $|X(u)| = 0$ на электрических схемах не существует, так как элемент должен быть соединен с каким-нибудь другим элементом, однако по указанию разработчика при макетировании иногда в корпуса включают «резервные» БЭ. Если есть электрическая цепь, соединяющая как выводы одного элемента между собой, так и выводы других элементов, то такую цепь при описании схемы гиперграфом будем расщеплять на два ребра, одно из которых принадлежит U_0 , а другое — U_1 . Заметим, что это пока единственное слабое место при описании электрической схемы устройства гиперграфом, не позволяющее получить адекватную математическую модель, однако пока не известен и способ точного описания электрической схемы графовой моделью в общем случае. Предлагаемое расщепление согласуется с утверждением в работе [91], где отмечается, что для лучшей трассируемости целесообразно рассматривать только те соединения, которые инцидентны только распределяемым контактам.

Для решения задачи компоновки считается заданным также множество допустимых для применения в данной разработке типов микросхем $\{X^k\}$. В зависимости от назначения разработки (макет, образец, серийное производство) к множеству $\{X^k\}$ может предъявляться условие $X^i \cap X^j = \emptyset$ при $i \neq j$, что соответствует включению в микросхемы i -го типа строго определенных видов БЭ. Если же выполнение этого условия не является обязательным, то это соответствует использованию при необходимости БЭ в корпусе с неполностью задействованными входами. Указанное различие в представлении множества $\{X^k\}$ в виде подмножеств не оказывает существенного влияния на алгоритмы компоновки и не влечет за собой принципиальных трудностей при программной реализации, поэтому особого внимания на этом в дальнейшем акцентировать не будем.

Задача компоновки заключается в построении гомоморфизма ψ :

$$G'(X', U'; R) = \psi G(X, U; R),$$

Напомним, что гомоморфизмом ψ называется отображение гиперграфа $G(X, U; R)$ в гиперграф $G'(X', U'; R)$, если каждой вершине $x \in X$ и каждому ребру $u \in U$ гиперграфа G однозначно отнесены их образы $\psi x \in X'$ и $\psi u \in U'$ в гиперграфе G' с сохранением инцидентности R .

К гомоморфизму ψ предъявляются следующие требования:

1) X' должно содержаться в $\{X^k\}$, т. е. получаемые результаты отображения должны содержаться в множестве типов ИС, разрешенных для применения в данной разработке;

2) в X' должен быть минимум вершин;

3) выражение $\sum_{u' \in U_i'} |X'(u')|$ должно быть минимальным;

4) максимально должна сохраняться функциональная связность.

Первое требование обеспечивает условие корректности при покрытии схемы заданным набором. Второе и третье требования в данной задаче взаимно связаны и противоречивы: при выполнении второго без учета третьего прообразом x'_m могут оказаться несмежные вершины x_{i_l} . Тогда число связей между вершинами может быть не минимальным, а выполнение третьего требования без учета второго может вести при одновременном уменьшении числа связей к увеличению числа ИС, т. е. к возрастанию стоимости разработки.

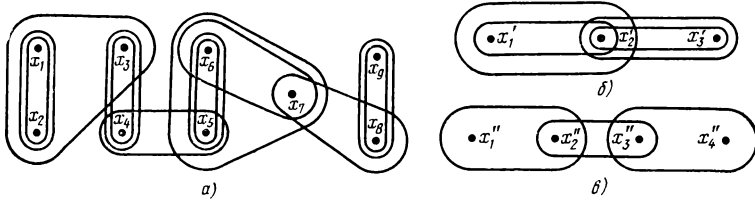


Рис. 4.1

Подтвердим это на примере гиперграфа, показанного на рис. 4.1,а. Пусть $\{X^k\}$ допускает отображение любых трех вершин представленного гиперграфа в одну, тогда возможно следующее отображение согласно второму требованию (рис. 4.1,б):

$$\psi_1^{-1}x'_1 = \{x_{i_1}, x_{i_2}, x_{i_3}\}; \quad \psi_1^{-1}x'_2 = \{x_{i_4}, x_{i_5}, x_{i_6}\}; \quad \psi_1^{-1}x'_3 = \{x_{i_7}, x_{i_8}, x_{i_9}\}.$$

Минимум вершин в X' достигнут — их три; очевидно, что при другом любом отображении по второму требованию множество U' будет содержать не менее четырех ребер.

При учете только третьего требования получаем (рис. 4.1,в):

$$\psi_2^{-1}x''_1 = \{x_{i_1}, x_{i_2}\}; \quad \psi_2^{-1}x''_2 = \{x_{i_3}, x_{i_4}\}; \quad \psi_2^{-1}x''_3 = \{x_{i_5}, x_{i_6}, x_{i_7}\}; \quad \psi_2^{-1}x''_4 = \{x_{i_8}, x_{i_9}\}.$$

При этом вершин в множестве X'' получается четыре, а число ребер в U'' — три.

В автоматических системах в зависимости от характера разработки в конфликтных ситуациях дается приоритет одному из этих требований. Например, если дать приоритет третьему требованию, то если ИС не полностью скомпонована и нет БЭ, которые имеют связь с установленными в компоуемый корпус, могут быть помещены в него, выбираются для помещения в корпус те БЭ, которые между собой сильно связаны.

В интерактивных системах целесообразно для принятия решения в конфликтных ситуациях привлекать пользователя. Для большей обоснованности принятия решения пользователем целесообразно привлечение дополнительной информации, например о связях между еще не распределенными и претендующими на установку в корпус элементами, информации о заполняемости других корпусов, о возможности оставления незадействованными БЭ в корпусе и т. д.

Выполнение четвертого требования обеспечивается следующим правилом выбора. При определении первой вершины прообраза при всех прочих равных условиях выбирается вершина из X , наиболее связанная с прообразами последней

отображенной вершины x'_m ; при наличии нескольких претендентов с одинаковым числом связей из них выбирается наиболее связанный с последними отображенными вершинами, т. е. просмотр делается в порядке, обратном отображению; в случае и такого равенства берется первый претендент. При подборе остальных вершин для отображения в вершину x'_m , если после подбора по второму и третьему требованиям остается больше претендентов, чем может быть помещено в корпус, выбираются вершины по числу необходимых, дающих максимум выражения $\Delta(x'_m) = \Sigma |X(u)| - \Sigma |X'(u')|$ и имеющих одновременно максимум связей с предыдущей отображенной вершиной. При равенстве числа связей глубина обратного зондирования увеличивается на единицу, т. е. учитывается

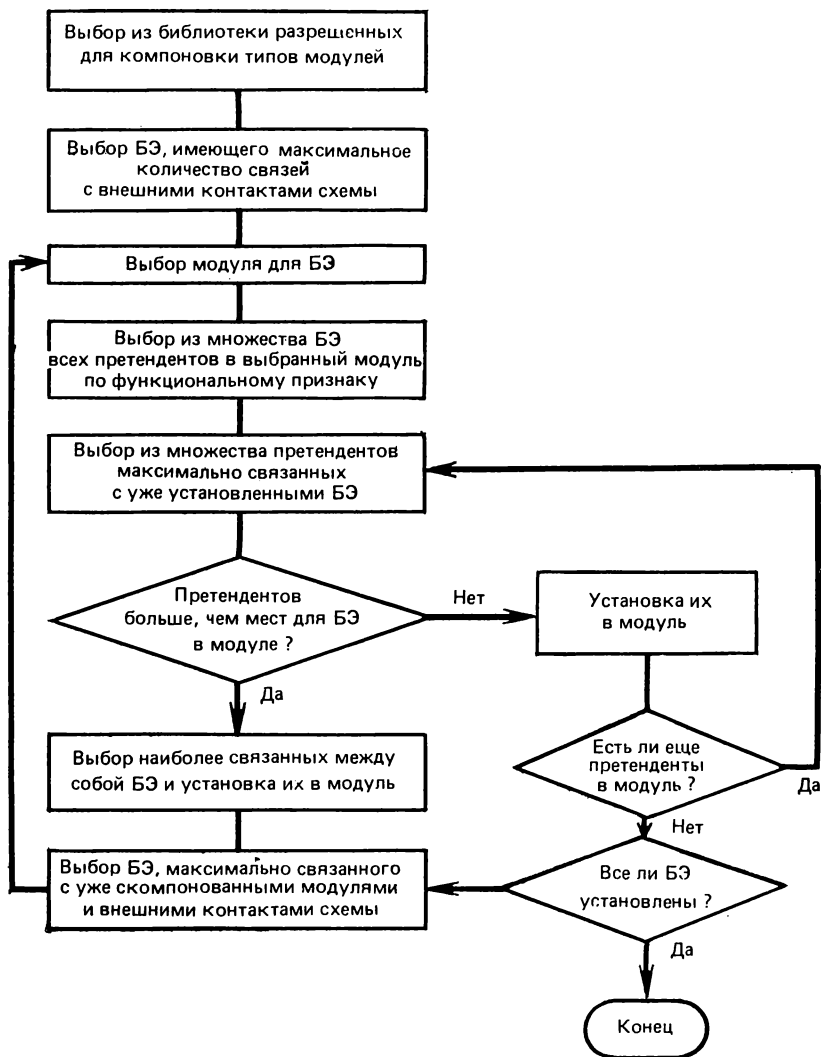


Рис. 4.2

число связей с двумя последними отображенными вершинами и т. д. Суммирование при вычислении $\Delta(x'_m)$ в первом члене выражения ведется по тем ребрам и гиперграфа G , которые инцидентны вершинам x_{ij} , отображенным в x'_m , а во втором члене — по ребрам u' гиперграфа G' , которые инцидентны вершине x'_m .

Изложенные выше требования заложены в следующем алгоритме.

1. Формируем $\{X^k\}$.
2. Отображаем $X_0 \xrightarrow{\psi} X'_0$.
3. Полагаем $m=1$.
4. Выбираем $x'_m \in \{X^k\}$ и один прообраз по четвертому требованию.
5. Если x'_m не допускает в себя еще отображений, то переходим к пункту 8.
6. Выбираем $\psi^{-1}x'_m \in \{x_{ij}\}$ с учетом указанного приоритета третьего требования над вторым и с окончательным подбором по четвертому.
7. Переходим к пункту 5.
8. Увеличиваем значение m на 1.
9. Если отображены не все вершины, то переходим к пункту 4.
10. Конец.

На рис. 4.2 приведена укрупненная блок-схема алгоритма распределения БЭ по корпусам ИС по предложенному методу.

Отличительными чертами изложенного алгоритма можно считать следующие: 1) поименный принцип сравнения БЭ при подборе в корпус, позволяющий применять при проектировании любую серию и тип ИС, имеющийся в базе данных; 2) открытая для пополнения база данных, способствующая быстрой настройке системы на любую серию ИС; 3) подбор БЭ в корпус в данном алгоритме благодаря четвертому требованию осуществляется не индивидуально, а сразу максимально связанными группами до полного комплектования [92].

4.2. ОПРЕДЕЛЕНИЕ ЭФФЕКТИВНОСТИ РАСПРЕДЕЛЕНИЯ БАЗОВЫХ ЭЛЕМЕНТОВ ПО КОРПУСАМ

Целесообразность включения задачи компоновки в САПР определяется всесторонним и комплексным анализом результатов автоматической компоновки и результатов всего проектирования в целом. Ниже приведены результаты исследования автоматического распределения БЭ по корпусам ИС. Для исследования было взято шесть электрических схем (табл. 4.1).

Как видно из приведенной таблицы, число корпусов, используемых при автоматической компоновке, уменьшилось в среднем на 1—2 на схему, т. е. приблизительно на 4 %. Этот результат, однако, не может быть определяющим, так как при столь небольшом числе корпусов можно предположить, что большого внимания уменьшению их числа не уделялось. В то же время при компоновке больших электрических схем число корпусов имеет большое значение.

При анализе целесообразно подсчитывать также число связей между корпусами по результатам автоматической и неавтоматизированной компоновки. Данные по этим же схемам приведены в табл. 4.2.

Согласно результатам, приведенным в табл. 4.2, число связей между различными корпусами на этих шести схемах уменьшилось на 74. (Заметим, что в цепи с n корпусами будет $n-1$ связь между ними.) При этом только на последней схеме стало на одну связь больше, а на остальных уменьшилось: на

Таблица 4.1

Номер схемы	Число корпусов при компоновке		Номер схемы	Число корпусов при компоновке	
	автоматической	неавтоматизированной		автоматической	неавтоматизированной
1	33	36	4	34	34
2	33	35	5	32	34
3	39	39	6	34	35

Таблица 4.2

Номер схемы	Число корпусов в цепи	Число цепей с данным числом корпусов, полученных компоновкой		Номер схемы	Число корпусов в цепи	Число цепей с данным числом корпусов, полученных компоновкой	
		автоматической	неавтоматизированной			автоматической	неавтоматизированной
1	2	88	48	4	2	33	32
	3	8	33		3	23	20
	4	6	6		4	12	14
	5	0	2		5	10	11
					6	11	11
2	2	89	88		5	2	78
	3	4	1	3		5	9
	4	6	7	4		4	3
	5	1	3	5		2	4
	7	1	1	6		1	0
	8	1	1	7		4	2
	11	1	1	8		4	1
	12	1	1	9		0	1
				10		0	3
				11		0	1
3	2	83	84	6		2	92
	3	12	13		3	16	29
	4	5	2		4	0	1
	5	4	5		5	1	1
	6	1	4				
	7	2	1				
	8	0	1				
	9	1	1				

1-й — на 18; на 2-й — на 4; на 3-й — на 30; на 4-й — на 3; на 5-й — на 19. В среднем на одну схему при автоматической компоновке число связей уменьшилось на 12. Более существенным является то, что на всех схемах увеличилось число цепей с меньшим числом контактов, что облегчает условия трассировки.

Для определения влияния этапа компоновки на результаты проектирования было осуществлено автоматическое размещение и трассировка по результатам компоновки как разработчика, так и ЭВМ. В табл. 4.3 приведены результаты, отражающие два основных показателя проектирования двухслойных печатных плат: число непроведенных связей (ЧНС) и число переходных отверстий (ЧПО).

Таблица 4.3

Номер схемы	Число непроведенных связей при компоновке		Число переходных отверстий при компоновке	
	автоматической	неавтоматизированной	автоматической	неавтоматизированной
1	2	5	227	253
2	20	13	384	305
3	12	18	402	344
4	28	25	433	465
5	12	9	321	344
6	9	4	254	330
Сумма	83	74	2021	2041

Из таблицы видно, что число непроведенных связей в среднем уменьшилось на 11 %, т. е. на 1—2 на одну схему в среднем, однако при этом увеличилось число переходных отверстий в среднем по 2 на одну схему. Если принять во внимание, что непроведенные связи, как правило, трудно реализуются, то можно предположить, что для реализации каждой непроведенной связи понадобится сделать в среднем три перехода со слоя на слой. При данном предположении подсчитаем приведенное число переходных отверстий:

$$p_n^1 = p^1 + 3l^1 = 2021 + 3 \cdot 83 = 2270, \quad p_n^2 = p^2 + 3l^2 = 2041 + 3 \cdot 74 = 2263,$$

т. е. $p_n^1 - p_n^2 = 7$, где p^1 — суммарное число переходных отверстий при неавтоматизированной компоновке; p^2 — то же, но при автоматической компоновке; l^1 — число непроведенных связей при неавтоматизированной компоновке, l^2 — то же, но при автоматической компоновке.

Таким образом, в результате автоматической компоновки получаем фактический выигрыш не только в числе проведенных связей, но и в числе переходных отверстий.

При необходимости корректировки полученных после трассировки результатов и доработки одним из важных факторов, влияющих на сложность доработки, является конфигурация проложенных трасс, заполнение трассами рабочего поля платы. Результаты анализа загрузки стволлов при исследовании горизонтальной ориентации корпусов на плате представлены в табл. 4.4.

Напомним, что исследуется горизонтальная ориентация корпусов на плате. Отметим также, что при различных методах компоновки получалось различное число корпусов.

Таблица 4.4

Номер схемы	Способ компоновки	Число участков, в которых имеется свободный канал		
		между рядами выводов одного корпуса	между различными корпусами	
			в горизонталь- ном стволе	в вертикаль- ном стволе
1	Автоматическая Неавтоматизированная	9	18	24
		14	19	28
2	Автоматическая Неавтоматизированная	11	16	22
		15	20	14
3	Автоматическая Неавтоматизированная	18	22	30
		12	19	25
4	Автоматическая Неавтоматизированная	7	19	11
		9	20	22
5	Автоматическая Неавтоматизированная	6	20	24
		7	21	27
6	Автоматическая Неавтоматизированная	4	12	26
		6	12	28

Из табл. 4.4 видно, что если колебания в загрузке межкорпусных частей стволов и есть, то в среднем разница равна нулю. В то же время увеличивается загрузка стволов между выводами одного корпуса. Из этого следует сделать вывод о необходимости оптимизации конфигурации трасс в пространстве около корпусов, что можно сделать, используя эквивалентность БЭ в корпусе и эквивалентность их входов.

4.3. ИСПОЛЬЗОВАНИЕ ЭКВИВАЛЕНТНОСТИ БАЗОВЫХ ЭЛЕМЕНТОВ И ИХ ВХОДОВ НА ЭТАПЕ КОМПОНОВКИ

Покажем на примере некоторых типов ИС возможность оптимизации расположения БЭ в пределах корпуса и назначения эквивалентных входов БЭ.

Анализ результатов компоновки показал, что задача не ограничивается распределением БЭ по корпусам ИС, так как на качество трассировки влияет и порядок установки БЭ в корпус, и порядок назначения выводов в цепях. В своей работе опытные конструкторы широко используют соотношения эквивалентности, переставляя местами в процессе прокладки трасс БЭ в корпусах, меняя номера выводов БЭ в цепях. В автоматизированных системах соотношения эквивалентности используются не полностью и не во всех системах. Укрупненно схема второго этапа компоновки выглядит следующим образом.

После распределения БЭ по корпусам ИС и приведения списка цепей к модульному виду выявляются ребра гиперграфа со степенью 1, т. е. фрагменты цепей, соединяющие выводы одной ИС; затем оптимизируется расположение БЭ в корпусе и назначение входов БЭ для каждой ИС; на следующем шаге корректируется порядок БЭ в корпусах и массив цепей; после этого формируются и выдаются результаты. Более подробно схема оптимизации показана на рис. 4.3.

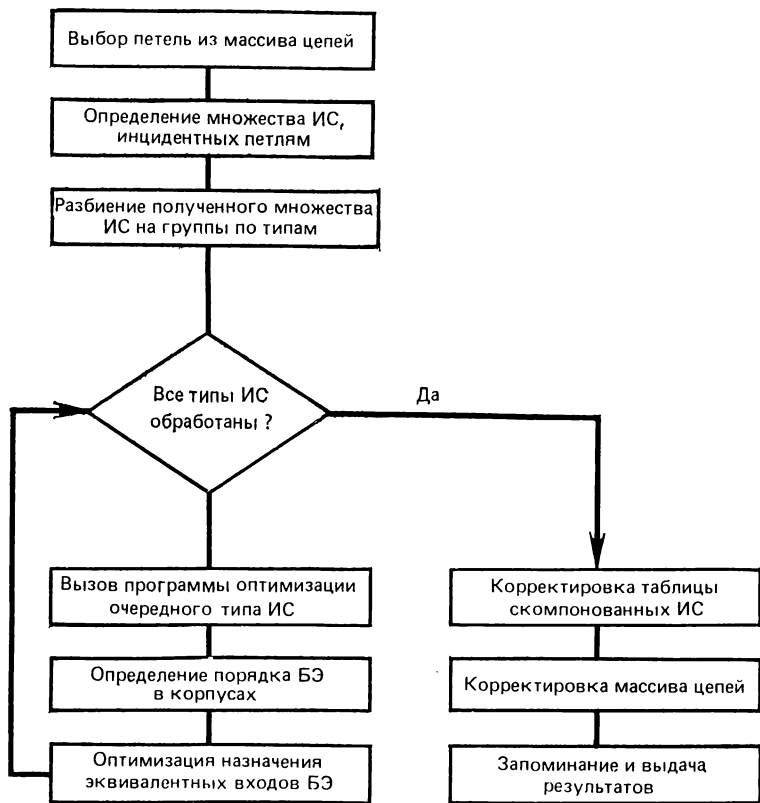


Рис. 4.3

Основным критерием оптимизации на данном этапе служила минимальная длина проводников, параллельных рядам выводов ИС. Этот критерий выбран потому, что при принятой ориентации в горизонтальном направлении между рядами выводов могут проходить сквозные проводники, соединяющие разные корпуса, а также проводники к выводам рассматриваемой ИС от других ИС или разъема, в то время как при существующей технологии производства ПП в большинстве случаев прокладка проводников в ортогональном направлении не представляется возможной. Таким образом, пространство около корпусов лучше использовать

для локальной разводки, т. е. для реализации ребер гиперграфа со степенью 1. В суммарной длине не учитываются горизонтальные соединения двух соседних выводов, так как при реализации на плате такие проводники не занимают каналы.

Программы оптимизации типов ИС базируются на различных алгоритмах. В некоторых ИС во время оптимизации базовые элементы просто переставляются местами, например в ИС, состоящих всего из двух триггеров. Для ИС, состоящих из одного или нескольких БЭ, строго закрепленных в корпусе, решается только задача оптимизации назначения эквивалентных выводов: для ИС, состоящих из двух эквивалентных БЭ с эквивалентными входами, кроме перестановки осуществляется также и оптимизация назначения входов. При оценке размещения элементов в корпусе выбирается лучшее по указанному выше критерию. Для оптимизации ИС, содержащих более трех эквивалентных БЭ, применялся метод блуждающего трафарета, описанный в § 3.2.

При оптимизации целесообразно применять следующий порядок обработки петель. Из массива цепей оптимизируемого модуля отделяются петли, содержащие только закрепленные выводы, т. е. не имеющие себе эквивалентных. Они не нужны для оптимизации назначения, однако при подсчете целевой функции эти петли учитываются, так как при перестановке БЭ расстояние между соответствующими выводами может меняться. После отбрасывания петель, содержащих только закрепленные выводы, программа обрабатывает петли, содержащие закрепленные выводы, имеющие своими соседями свободные выводы; затем обрабатывает петли, содержащие только свободные выводы. Такой порядок рассмотрения петель позволяет минимизировать не только длину горизонтальных соединений, но и число занимаемых горизонтальных каналов.

Как и в § 4.1, множество ребер гиперграфа $G(X, U; R)$ запишем следующим образом: $U = U' \cup U''$.

В свою очередь $U' = \bigcup_x U'_i$, где U'_i объединяет петли, относящиеся к i -му модулю.

Для оценки качества размещения БЭ в i -м корпусе и оценки назначения эквивалентных выводов предлагается следующая функция:

$$F_i = \sum_{U'_i} F_{ij}, \quad (4.1)$$

$$\text{где } F_{ij} = \begin{cases} \Delta x + \alpha, & \text{если } \Delta x > 1, \\ \Delta x + \alpha, & \text{если } \Delta x = 1 \text{ и } \Delta y = 0, \\ 0 & \text{— во всех остальных случаях,} \end{cases}$$

где, в свою очередь, $\Delta x = x_{\max} - x_{\min}$ — разница между максимальной и минимальной абсциссой выводов, содержащимися в j -й петле i -го корпуса аналогично $\Delta y = y_{\max} - y_{\min}$. Слагаемое α подби-

рается эмпирически. Приводимые ниже результаты соответствуют $\alpha=1$.

Минимизация функционала (4.1) согласуется с минимизацией длины используемых горизонтальных каналов и с минимизацией числа занимаемых горизонтальных каналов.

Рассмотрим ИС, содержащую три одинаковых базовых элемента ЗИ — НЕ. Число вариантов размещения БЭ в корпусе — $3!$ При очередном размещении БЭ решается задача оптимизации назначения эквивалентных входов. На рис. 4.4,а, приведен пример разводки петель из реальной платы ИС рассматриваемого типа до оптимизации. После оптимизации порядок БЭ в корпусе стал $3-2-1$; кроме того, в первом БЭ вместо входа z_2 в цепи задействован вход z_1 (рис. 4.4,б).

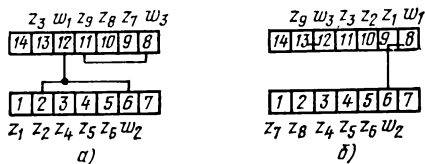


Рис. 4.4

Если при подсчете суммарной длины по формуле (4.1) получается $F_i=0$, то перебор прекращается.

Как уже указывалось выше, для оптимизации ИС, содержащих большее количество элементов, чем 3, применен метод блуждающего трафарета. Метод предполагает последовательную установку БЭ в корпус, а также регулярную структуру размещения. Рассмотрим применение этого метода на примере ИС, содержащей четыре базовых элемента 2И — НЕ. На приводимых рисунках входы БЭ обозначены буквами z , а выходы — w .

Последовательность установки БЭ в корпус определялась для рассматриваемой ИС по следующему алгоритму.

1. Выбираем элемент с наиболее задействованными выводами; если таковых более одного, то выбираем элемент, наиболее нагруженный на выход (нагрузка и связи здесь и в дальнейшем в этом параграфе определяются только из массива петель, описанного выше); если и таковых более одного, то выбирается любой из них, и выбранный считается первым претендентом — α_1 .

2. Из оставшихся трех элементов выбирается наиболее связанный с первым; если таковых более одного, то выбирается наиболее связанный с выходом первого претендента α_1 ; если и таковых более одного, то выбирается любой из них и выбранный считается вторым претендентом — α_2 .

3. Из оставшихся двух элементов выбирается наиболее связанный с двумя уже выбранными; если оба элемента имеют одинаковое число связей (число связей, отличное от нуля, в противном случае и третий, и четвертый претенденты равны между собой и заносятся в список претендентов в произвольном порядке), то выбирается связанный с выходом первого выбранного претендента; при равенстве и этих связей выбирается связанный с выходом второго выбранного претендента и выбранный считается третьим претендентом — α_3 . Если по

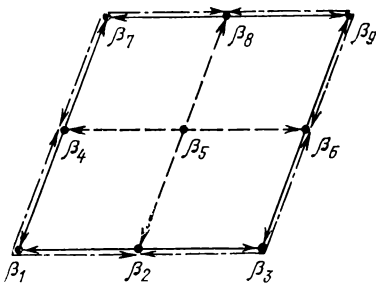


Рис. 4.5

перечисленным требованиям оба претендента равны, то они ставятся в список в произвольном порядке.

4. Оставшийся претендент будет в списке четвертым — α_4 .

Для данной ИС расширенной моделью посадочных мест будет параллелограмм, изображенный на рис. 4.5 с местами установки $\beta_1, \beta_2, \beta_3, \dots, \beta_9$, так как $n=2, m=2$ (установка 2-го элемента обозначена штриховой линией, 3-го — непрерывной, 4-го штрихпунктирной).

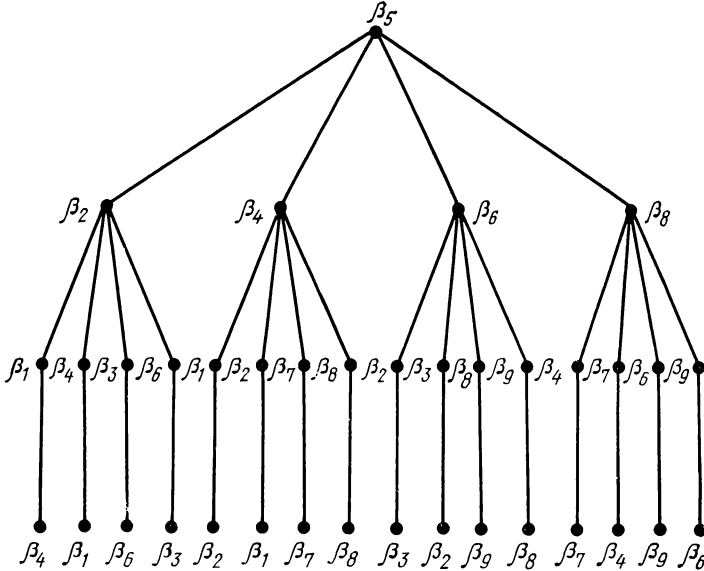


Рис. 4.6

Элемент α_1 устанавливается на позицию β_5 . Так как элемент α_2 выбирается наиболее связанным с α_1 , то перспективными местами для установки этого элемента следует считать позиции $\beta_2, \beta_4, \beta_6, \beta_8$. Допустим, что элемент α_2 по некоторым причинам установился на позицию β_2 ; тогда трафарет, занимая крайние левое и правое положения, опишет возможные позиции установки для элемента α_3 — β_4, β_7 и β_6, β_9 . После установки третьего элемента позиция для четвертого определяется однозначно.

На рис. 4.6 приведено дерево решений распределения БЭ для ИС рассматриваемого типа. Праввершиной дерева является позиция β_5 , соответствующая установке в центр элемента α_1 , на втором уровне — позиции $\beta_2, \beta_4, \beta_6, \beta_8$, соответствующие установке элемента α_2 и т. д. Процесс установки БЭ в корпус для данной ИС соответствует движению по одной из ветвей дерева решений от праввершины первого уровня к одной из вершин четвертого уровня.

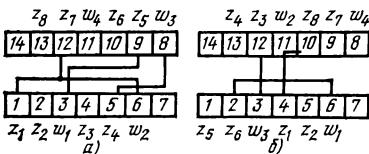


Рис. 4.7

На рис. 4.7 приведен пример оптимизации по описанному алгоритму распределения БЭ и выводов для микросхемы упоминаемого выше типа. На рис. 4.7,а

показана разводка петель до оптимизации; на рис. 4.7,б — после оптимизации. Для простоты ориентации на рис. 4.7,б сохранены номера старых БЭ. Если до оптимизации БЭ имели порядок установки в корпусе 1—2—3—4, то после оптимизации получился порядок 3—1—4—2, а при назначении выводов во втором БЭ менялись входы z_4 на z_3 , в третьем БЭ — z_5 на z_6 и в четвертом — z_7 на z_8 . Соответствующие изменения проводятся и во входной информации.

В рассматриваемом примере метод блуждающего трафарета позволил сократить число вариантов на дереве решений в 1,5 раза по сравнению с методом перебора. Дальнейшее сокращение объема вычислений зависит от метода поиска, использующего дерево решений: поиск по глубине, поиск по ширине, метод ветвей и границ, использование функции ветвления. Заметим, что для задач большой размерности целесообразно использовать последний метод, так как он не требует большого объема вычислений. Кроме того, метод блуждающего трафарета позволяет при установке очередного элемента в большинстве случаев увеличить число просматриваемых мест по принятым критериям для выбора лучшего места.

Приведем таблицу окончательных результатов после распределения БЭ по корпусам, оптимизации размещения БЭ в корпусе и оптимизации назначения эквивалентных выводов. (Схемы те же, что и в табл. 4.1—4.4.)

Из табл. 4.5 видно, что использование соотношений эквивалентности даже при оптимизации не всех ИС на плате позволило избавиться от непроведенных связей на одной плате, т. е. исключить этап коррекции; на оставшихся платах число непроведенных связей существенно уменьшилось, а число переходных отверстий сократилось на всех платах; в отдельных случаях число переходных отверстий уменьшилось на 15 % по сравнению с числом переходных отверстий до оптимизации.

Как и в § 4.2, подсчитаем общее число приведенных переходных отверстий: $p_n^3 = p^3 + 3l^3 = 1863 + 3 \cdot 27 = 1944$, т. е. $p_n^1 - p_n^3 = 326$, что составляет около 14,3 % от первоначального общего приведенного числа переходов.

4.4. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ЭТАПА КОМПОНОВКИ

При разработке системы компоновки кроме описанных выше критериев необходимо учитывать и ее эксплуатационные качества. Под эксплуатационными качествами подразумевается удобство пользования при эксплуатации, простота

Таблица 4.5

Номер схемы	Число оптимизируемых ИС	Число непроведенных связей	Число переходных отверстий
1	23	0	215
2	29	9	292
3	25	11	330
4	19	22	436
5	30	5	301
6	20	3	289
Сумма	—	50	1863

подготовки информации, максимальная достоверность получаемых результатов, а также получение выходной информации в виде, удобном для использования на последующих этапах.

Следует различать два возможных случая включения компоновки в САПР: первый — когда в системе уже автоматизированы этапы размещения и трассировки; второй — когда автоматизация всех трех этапов проводится одновременно.

В первом случае на предприятии уже сложились традиции по подготовке и прохождению документации при проектировании. Чтобы не увеличивать чрезмерно объем входной информации, по кодируемым ранее связям и вводимому перечню используемых элементов из базы данных извлекаются сведения о БЭ, структуре ИС. Таким образом, при включении компоновки в уже действующую систему основная работа по подготовке заключается в преобразовании имеющейся в базе данных информации об ИС. При этом скомпонованная проектировщиком схема снова разлагается до БЭ, после чего проводится компоновка с оптимизацией.

Во втором случае целесообразно сразу предусмотреть такой язык описания схем, который позволял бы описывать информацию в БЭ. Такой язык должен быть близок к мнемоническому, чтобы пользователю, не знакомому с программированием, не понадобилось много времени для его изучения.

Входной язык может быть ориентирован на проектирование специализированной аппаратуры отрасли, учитывать особенности проектирования определенного класса изделий и т. п., поэтому можно дать только общие рекомендации по его разработке.

Естественно, что все элементы схемы должны быть пронумерованы. Это необходимо как для описания связей, так и для описания выполняемых элементами функций. При описании связей обычно предусматривают возможность директивного порядка проведения цепей. Для этого в начале описания связи ставится число; его отсутствие означает определение порядка трассировки программным путем. Примером простого предложения, описывающего цепь, может быть запись В5 [R3] [C7] [D2 [8]]. Она означает, что 5-й контакт соединителя соединяется с одним из выводов резистора 3; в эту же цепь включается конденсатор 7 любым выводом и диод 8 вторым выводом.

Если в цепи задействованы входы (ВХ) базовых элементов, выходы (ВЫХ) и дискретные ЭРИ или любые из указанных компонентов, то такая цепь может быть описана сложным предложением: ВХ3[5] 6 [2 : 4] = ВЫХ1[7]. Здесь вход 3 базового элемента 5 соединяется с входами 6 базовых элементов 2, 3, 4 и с выходом 1 базового элемента 7. Символ «=» соединяет простые предложения в сложные. Предложение, описывающее только входы или только выходы, может быть только подчиненным. В противном случае оно считается ошибочным.

Кроме описания связей в ЭВМ должны вводиться данные о типах БЭ на проектируемой схеме. Описание можно начинать с описания функции, выполняемой БЭ, перечислить БЭ, выполняющие эту функцию. Например, 3И — НЕ [7, 9, 12 : 20], 8ИЛИ [10, 23, 25]. Эта запись означает, что БЭ 7, 9 и с 12 по 20 выполняют логическую функцию 3И — НЕ, а БЭ 10, 23, 25 — логическую функцию 8ИЛИ.

Целесообразно за разработчиком оставить право закреплять отдельные БЭ в корпусе ИС определенного типа, как то: 155ЛА3 [4, 16, 31]. Необходимо также вводить указания по использованию серий ИС, а также, при желании, указывать допустимые к применению конкретные типы ИС в указанных сериях.

В противном случае должны использоваться произвольные типы и серии, хранящиеся в базе данных, а в случае отсутствия — выдаваться сообщение.

В хранящемся описании ИС должны быть сведения о допустимой нагрузке, о функциональном назначении БЭ, о соответствии выводов БЭ выводам ИС. По этой информации проводится такой контроль, как корректность указаний разработчика при директивном заполнении корпусов, правильность использования БЭ.

Включение этапа компоновки в САПР позволяет расширить входной контроль, что особенно важно, так как обнаружение ошибки на входе и исправление ее обходится в сотни раз дешевле, чем исправление той же ошибки и ее последствий в выходной информации [94]. Кроме несоответствия информации грамматике транслятора входного языка возможно также выявление ошибок схемного характера. Например, в цепи не допускается использование только входов или только выходов БЭ, не допускается наличия в одной цепи двух выходов одного и того же БЭ, проверяется допустимая нагрузка и т. п.

Одним из достоинств системы, проводящей компоновку, является возможность без ввода дополнительной информации получить конструкторскую документацию по результатам компоновки: таблицу распределения БЭ по корпусам ИС, спецификации, перечень элементов, ведомость покупных изделий.

Эксплуатация описанной подсистемы имеет свои особенности, одной из которых является необходимость постоянного контроля за используемыми на предприятии типами ИС, целью которого является составление списка идентичных по конфигурации расположения БЭ в корпусе и одинакового расположения их эквивалентных выводов. На куратора системы возлагаются обязанности разработки алгоритмов и программ оптимизации, соответствующих новым типам ИС, отсутствующих в системе.

Другой особенностью эксплуатации подсистемы компоновки с использованием соотношений эквивалентности БЭ и их входов является изменение привычного порядка прохождения КД. Если при использовании САПР, решающей только задачи размещения и трассировки, в конструкторскую группу, занимающуюся автоматизированным проектированием, электрические схемы поступают полностью оформленными, то при включении этапа компоновки в САПР принадлежность БЭ к конкретному номеру ИС проставляется после компоновки, тогда же оформляются ведомость покупных изделий и спецификация. Использование соотношений эквивалентности требует после этапа компоновки проставлять еще и номера задействованных выводов БЭ.

Указанные необходимые изменения в порядке прохождения КД вызывают сопротивление со стороны конструкторов при внедрении систем подобного типа, поэтому при подготовке к эксплуатации необходимо создать максимальные удобства в пользовании, предусмотреть тщательный контроль на всех этапах, выдачу результатов в удобочитаемом виде.

Выполнение указанных требований одновременно с улучшением качества печатного монтажа и сокращением сроков разработки привлекает внимание разработчиков и конструкторов, занимающихся автоматизированным проектированием. Заметим, что включение в САПР подсистемы автоматического вычерчивания принципиальных схем снимает с конструкторов заботу по переоформлению схем в случае их изменения, так как информация о чертеже принципиальной схемы может храниться в базе данных до окончания проектирования схемы или даже до изготовления ее в производстве, а коррекция информации по результатам компоновки может проводиться автоматически. При изменении схемы кор-

рекция проводится через ввод дополнительной информации, что не представляет трудностей. Из сказанного следует, что вслед за внедрением в САПР подсистемы компоновки должен встать вопрос автоматизации выпуска КД, а в частности — автоматического проектирования принципиальных электрических схем.

Глава 5

ПОСЛЕДОВАТЕЛЬНО-ГРУППОВОЙ ПОДХОД К РАЗМЕЩЕНИЮ ЭЛЕМЕНТОВ

5.1. ОБЩИЕ ТРЕБОВАНИЯ К ЗАДАЧЕ РАЗМЕЩЕНИЯ

Задача размещения заключается в предоставлении каждому элементу схемы конкретных координат на плоскости с соблюдением следующих условий: 1) элементы должны быть установлены на ограниченной части плоскости заданной конфигурации; 2) проекции элементов на плоскость установки не должны пересекаться (для односторонней установки); 3) между элементами должны соблюдаться необходимые зазоры.

Конфигурация платы, порождающая первое условие, обычно диктуется назначением разработки (наземная, бортовая аппаратуры), принятыми стандартами предприятия, технологическими возможностями производства и т. п. Одной из основных причин второго условия является ремонтпригодность изделия. Третье условие диктуется технологичностью изготовления изделия, температурными режимами и т. п. Заметим, что третье условие не исключает второе, так как в случае отказа от второго условия возможна установка одного элемента над другим с соблюдением зазоров.

При размещении дискретных элементов их размеры при описании обычно увеличиваются до минимально допустимых с учетом третьего условия и размеров конструкторской сетки. При размещении ИС эта проблема практически не возникает, так как отношение плотности печатного монтажа к плотности установки элементов в этом случае гораздо выше, чем при установке дискретных ЭРИ.

При размещении дискретных элементов посадочные места, как правило, не выделяются. Установка разрешается с одним ограничением: выводы элементов должны находиться в узлах конструкторской сетки, т. е. расстояние между выводами должно быть кратным шагу сетки. Учитывая, что у дискретных элементов выводы в большинстве случаев подвергаются формовке с большим допуском, это ограничение не следует считать жестким.

При размещении ИС входными данными обычно являются: N микросхем — $a_1, a_2, a_3, \dots, a_N$, подлежащих установке (некоторые из них могут быть размещены по желанию разработчика

директивно); множество связей между элементами, на отдельные из которых может быть наложено условие минимальной длины, или наибольшей разнесенности отдельных связей между собой; регулярная сетка посадочных мест на плате — $\{\beta_{ij}\}$, где $i \leq m$, $j \leq n$ при этом $N \leq m \cdot n$. На плате могут быть указаны запрещенные места для установки, различные вырезы и т. п.

Ставя в соответствие каждому модулю a_i единственное посадочное место β_{ij} , необходимо учесть связи между элементами таким образом, чтобы создать благоприятные условия для трассировки. Под благоприятными условиями понимается такое размещение, при котором части рабочего поля платы при проведении трасс будут использоваться наиболее равномерно.

Наибольшее применение в действующих САПР нашли последовательные и итерационные методы размещения [78, 79]. Недостатком последовательных методов является зависимость последующих устанавливаемых элементов от ранее установленных, т. е. жесткое закрепление элементов на определенных позициях плохо сказывается на качестве установки последующих элементов. Преимущество их заключается в простоте реализации и в скорости работы, хотя они уступают при этом по качеству результатам, полученным итерационным методом [95]. Для размещения разномасштабных элементов более применимы последовательные алгоритмы.

Качество и скорость работы итерационных алгоритмов зависит от начального размещения. В работе [52] показано, что алгоритм перестановки связанных множеств значительно быстрее и качественнее обеспечивает получение окончательного результата, чем алгоритмы перестановок несвязанных множеств. О необходимости учитывать связность групп элементов при размещении указано также в работе [96].

На основе групповых и парных перестановок разработан ряд алгоритмов, описанных, например, в [97]. Эти алгоритмы предполагают разбиение платы на горизонтальные и вертикальные блоки. Оптимизация ведется перестановкой как блоков, так и элементов в пределах блока. Основными критериями в большинстве таких подходов является линеаризация соединений, и, хотя при этом не учитывается протяженность занимаемых каналов и их число, занимаемое одной трассой, авторы указывают в работе на хорошие конечные результаты.

Более гибким и перспективным следует считать метод, предложенный в работе [98]. В данном методе при оптимизации расположения блоков и элементов в блоке связи между ИС классифицируются по принадлежности к различным рядам выводов ИС, что позволяет учитывать условия трассировки более точно. Предложенная в работе схема вычислений обладает высоким быстродействием.

Общим недостатком большинства критериев установки (минимальная суммарная длина связей, минимум пересечений, линеаризация соединений, минимизация полупериметра описывающего

прямоугольника и т. п.) является то, что они не учитывают вообще или слабо учитывают удлинение трасс, которое получается при прокладке, а в оптимизируемых функциях не учитывается число возможно требуемых для реализации связей каналов, что снижает эффективность этих критериев. Анализ, проведенный в § 3.3, показал, что размещение на плате элементов должно проводиться с учетом возможного изменения конфигурации трасс, получаемого в процессе трассировки. Учитывая, что наибольшее среднее отклонение прокладываемых трасс от минимального построения дерева получается у коротких цепей и что более половины цепей на схемах содержат по 2—3 контакта, можно сделать вывод, что при размещении достаточно учитывать только короткие цепи [99].

В последнее время наметился и разрабатывается подход, когда корпуса размещаются в некоторой окрестности отведенного им посадочного места. Это объясняется следующим. На схемах используются корпуса ИС различных габаритов, поэтому при установке их делают по возможности равных, приведенных или кратных размеров. При анализе загрузки частей платы отдельные ИС или их ряды несколько сдвигают или раздвигают, освобождая один-два канала для прокладки трасс. Этим достигается более рациональное использование поля трассировки.

5.2. РАЗМЕЩЕНИЕ РАЗНОГАБАРИТНЫХ ЭЛЕМЕНТОВ

При оптимизации размещения на плате элементов разных геометрических размеров улучшение начального размещения итерационными методами не так эффективно, как при оптимизации размещения элементов равных размеров или при «рыхлом» размещении элементов приблизительно равных размеров. Поэтому плотному размещению элементов разных геометрических размеров необходимо уделять большее внимание. Последовательное размещение элементов как равных, так и разных геометрических размеров сильно зависит от порядка установки элементов.

Прежде чем описывать предлагаемый алгоритм, введем некоторые понятия и определения. Условимся называть контакты данной электрической схемы вершинами, а проводники, соединяющие контакты — ребрами. Тогда электрической схеме будет соответствовать гиперграф $G(X, U; R)$.

Через G^* обозначим гиперграф, в который входят ребра с вершинами, являющимися либо контактами соединителя, либо контактами элементов, уже вошедших в список очередности установки элементов. Наиболее распространены следующие два критерия определения порядка установки [2]. При использовании первого очередным выбирается элемент, имеющий наибольшее число связей, с уже установленными элементами. Недостаток этого критерия: начиная с некоторого момента будут устанавливаться только те элементы, у которых один контакт является вершиной ребра с наибольшей степенью. Например, на рис. 5.1

в список очередности установки сначала войдут элементы 1, 2, 3, 4, 7, 8, 9 и т. д. и только после того, как все номера элементов плюсовой цепи питания будут занесены в список очередности, возможно, в него будут занесены элементы 5 и 6.

Другой критерий основан на выборе очередного элемента, у которого отношение числа связей с уже выбранными элементами к общему числу связей достигает максимума. Его недостаток: элементы, контакты которых принадлежат ребрам с большей сте-

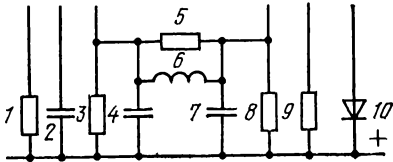


Рис. 5.1

пенью, заносится в список очередности последними; при этом элементы связных групп разбиваются.

Предложим следующий алгоритм определения порядка установки.

1. В начальный момент в гиперграф G^* входят только те ребра, которые содержат

контакты-вершины соединителя или директивно установленных элементов. На первом этапе выбирается элемент, у которого контакты являются вершинами хотя бы двух различных ребер гиперграфа G^* . Этот элемент заносится первым в список очередности установки элементов на плату и дальше не рассматривается. Затем гиперграф G^* пополняется ребрами, в которых нет вершин-контактов соединителя и которые одновременно содержат вершины-контакты только что установленного элемента.

2. После занесения номера элемента в список очередности строится гиперграф G^{**} , состоящий из ребер, содержащих вершины-контакты последнего занесенного в список установки элемента.

Следующими берутся элементы, у каждого из которых хотя бы два контакта являются вершинами ребер гиперграфа G^{**} . После этого снова пополняется гиперграф G^* , и для каждого из вновь занесенных элементов опять строится гиперграф G^{**} , вновь пополняется список очередности и достраивается гиперграф G^* .

Целесообразность этого иллюстрируется рис. 5.2 (цифры без квадрата означают номер элемента, цифры в квадрате — номер внешней среды). Если предположить, что в списке очередности элемент 7 уже есть и последним занесен элемент 8, то согласно

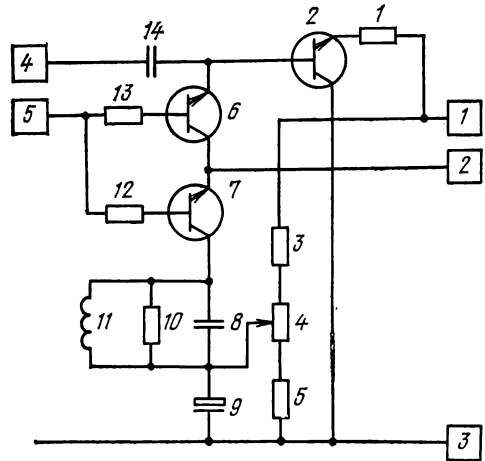


Рис. 5.2

сказанному выше, элементы будут выбираться в порядке: 10, 11, 9, а не 9, 10, 11.

Если такие элементы отсутствуют, то выбирается элемент, контакты которого являются вершинами хотя бы двух различных ребер гиперграфа G^* и продолжается описанный выше процесс, начиная с п. 1.

3. После того как будут установлены элементы по п. 1 и 2, следующим выбирается элемент, один контакт которого является вершиной гиперграфа G^* и который обладает тем свойством, что после его занесения в список очередности установки и после пополнения гиперграфа G^* появится другой элемент, у которого контакты будут вершинами двух различных ребер гиперграфа G^* . Далее расширяется гиперграф G^* и процесс продолжается, начиная с п. 2.

4. Если не окажется и таких элементов, а еще не все элементы занесены в список очередности, тогда из всех элементов, не вошедших в список очередности, контакты которых являются вершинами гиперграфа G^* , выбирается такой, у которого соответствующее ему ребро, принадлежащее гиперграфу G^* , имеет меньшую степень. После занесения такого элемента в список очередности гиперграф G^* расширяется, и процесс начинается с п. 3. Пункт 4 обеспечивает занесение в список очередности всех элементов электрической схемы, так как изолированные вершины в соответствующем гиперграфе отсутствуют.

Данный алгоритм лишен недостатков, которые присущи алгоритмам, основанным на описанных выше критериях: он сохраняет наиболее связанные между собой группы элементов. Лучшие результаты получены для схем, содержащих, в основном, дискретные элементы, имеющие до четырех контактов.

Перед описанием процедуры установки определим используемые в дальнейшем некоторые понятия. *Высотой фронта* установки назовем ту часть высоты платы, на которую разрешается устанавливать элементы (из высоты платы вычитается расстояние, необходимое для соблюдения технологических требований). В начальный момент фронт установки представляет собой один вертикальный отрезок прямой. По мере установки элементов, например от левого нижнего края платы, часть фронта, равная по высоте элементу, будет сдвигаться вправо. После установки нескольких элементов в общем случае фронт будет представлять собой несколько вертикальных отрезков, отделенных друг от друга горизонтальными скачками. Такой вертикальный отрезок между двумя ближайшими скачками либо между верхней или нижней границей и ближайшим скачком будем называть *локальным фронтом установки*. Очевидно, что в начальный момент локальный и первоначальный фронты совпадают и высоты их равны.

Предлагается следующий алгоритм установки. После установки $(i-1)$ -го элемента из списка очередности выбирается по порядку элемент и ставится на первый по высоте локальный фронт (рис. 5.3,а). Подсчитывается целевая функция, после чего эле-

мент ориентируется и целевую функцию подсчитывают снова. После сравнения двух полученных результатов запоминаются данные о расположении элемента с лучшим значением целевой функции. Если высота локального фронта больше, чем высота элемента, то элемент сдвигается вверх на эту разность по локальному фронту (рис. 5.3, б) и снова прodelываются описанные выше подсчеты.

Далее переходим к следующему локальному фронту и снова прodelываем весь процесс установки, который продолжается по всей ширине фронта установки.

При установке элемента на локальный фронт делается проверка: помещается ли устанавливаемый элемент на плате по ее дли-

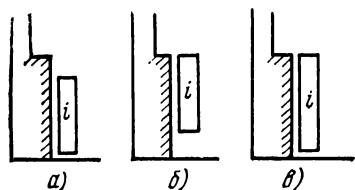


Рис. 5.3

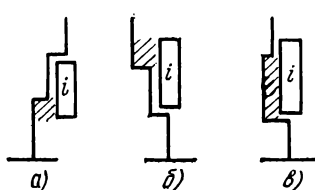


Рис. 5.4

не. Если элемент не устанавливается, то переходят к следующему локальному фронту.

Возможен случай, когда высота локального фронта равняется высоте элемента, тогда, очевидно, сдвига по фронту не делается (рис. 5.3, в). Если же высота локального фронта меньше высоты элемента, то делается проверка на возможность его передвижения:

элемент смещаем вниз, если предыдущий локальный фронт меньше настоящего (рис. 5.4, а);

элемент оставляем на месте, если не выполнено первое условие и последующий локальный фронт меньше настоящего (рис. 5.4, б);

элемент смещаем вправо до ближайшего фронта — предыдущего или последующего (рис. 5.4, в), если не выполнены два первых условия.

При описанной выше установке элемента образуется площадь, на которую уже нельзя поставить элемент; эта площадь учитывается в целевой функции, т. е. при всех прочих равных условиях элемент устанавливается на то место, где меньше надо «забывать» площадь (штриховка на рис. 5.4).

После установки очередного элемента выравнивают фронт, так как возможно появление локального фронта, меньшего по высоте, чем высота минимального по высоте элемента.

Выравнивание проводится следующим образом.

Если предыдущий локальный фронт меньше настоящего, то выравнивание идет по последующему фронту (рис. 5.5, а). Если предыдущий фронт больше настоящего, а последующий меньше,

то его выравнивают по предыдущему локальному фронту (рис. 5.5,б). Если же и предыдущий и последующий фронты больше настоящего, то выравнивание проводится по меньшему из них (рис. 5.5,в). Выравнивается только тот фронт, высота которого меньше высоты минимального еще не установленного элемента.

Первые два предположения основаны на том, что фронт, который меньше минимальной возможной высоты элемента, не может быть одновременно больше и предыдущего и последующего фронтов. Кроме того, после установки одного элемента может появиться не более одного такого фронта.

Минимизируемая целевая функция имеет вид

$$F_i = \sum_k \sum_j (\alpha_j \Delta x_{ij}^k + \beta_j \Delta y_{ij}^k) + S_i,$$

где $\alpha_j = \beta_j = 0$, если i -й элемент не связан с j -м. Кроме того, $\alpha_j = 0$, если элемент связан с корпусом. Во всех остальных

случаях $\alpha_j = 1$, а $\beta_j = 2$, если i -й элемент связан с j -м; $\beta_j = 4$, если элемент связан с корпусом, и $\beta_j = 0$, если элемент связан со свободным выводом разъема; Δx_{ij}^k — разность между абсциссами выводов устанавливаемого элемента с уже установленными; Δy_{ij}^k — то же самое, только с ординатами. Суммирование по j означает перебор по всем уже установленным элементам, а суммирование по k — перебор по всем выводам устанавли-

ваемого элемента; S_i — площадь, «забываемая» при установке элемента на данный локальный фронт.

Исследования показывают, что использование описанного порядка определения установки дискретных элементов на плату с данным размещением дает результаты, близкие к результатам конструктора.

Представляет интерес развитие подхода, описанного в работе [45]. Он основан на представлении электрической схемы устройства графовой моделью следующим образом. Вершинами

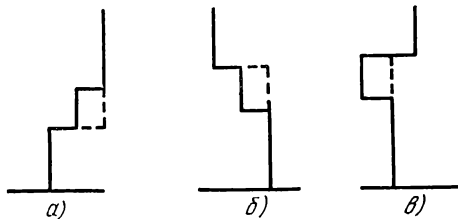


Рис. 5.5

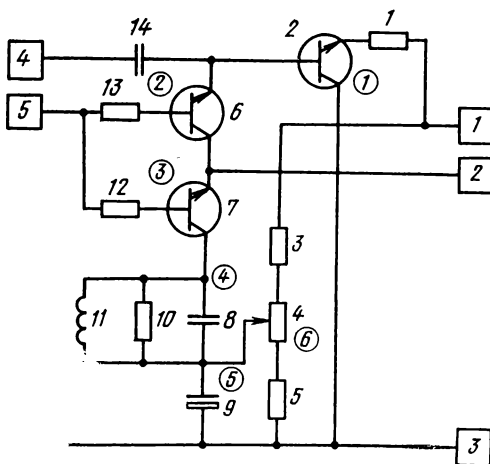


Рис. 5.6

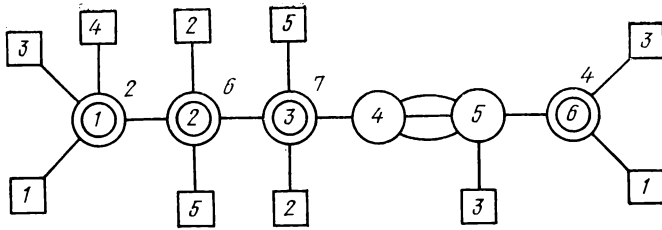


Рис. 5.7

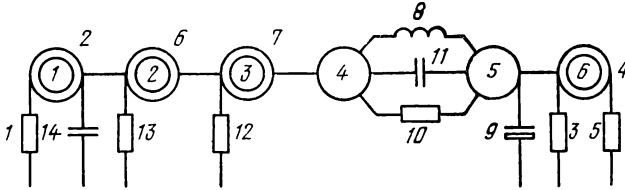


Рис. 5.8

графа считаются многовыводные элементы, точки ветвления, не лежащие между многовыводными элементами (обозначены на рис. 5.6 цифрами в кружке), а также внешние контакты схемы (на рис. 5.6— цифры в квадратах). На рис. 5.6 показана пронумерованная схема, соответствующая рис. 5.2. Ребрами в графе считаются связи между вершинами в виде либо проводников, либо двухвыводных элементов. Если ребро соответствует двухвыводному элементу, то этот элемент может представлять ребро единожды (элемент 14 на рис. 5.6 соответствует ребру, соединяющему 4-й внешний контакт с элементом 2). На многовыводные элементы такого ограничения не накладывается (на рис. 5.7 обозначено: цифра — номер элемента в схеме, цифра в квадрате — внешняя связь, цифра в кружке — вершина, соответствующая узлу, цифра в двойном кружке — вершина, соответствующая многовыводному элементу).

На следующем шаге отбрасываются ребра, не содержащие двухвыводных элементов (рис. 5.8; обозначения те же, что и на рис. 5.7). По полученным результатам формируются группы эле-

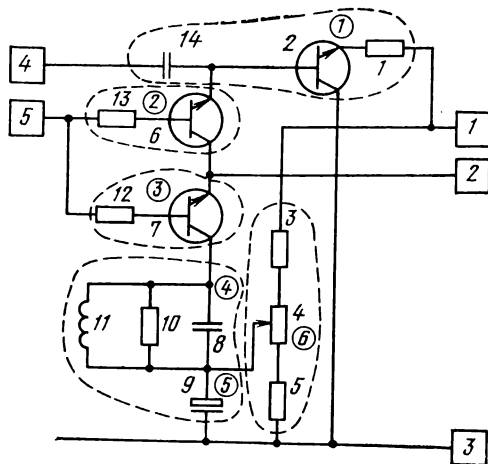


Рис. 5.9

ментов. Многовыводные элементы собирают в «свою» группу элементы, соответствующие инцидентным им ребрам; вершины, соответствующие узлам, если они смежные, входят в одну группу вместе с инцидентными им ребрами. Так, в рассматриваемом примере образовались следующие группы: $1-2-14$; $6-13$; $7-12$; $8-10-11-9$; $3-4-5$ (рис. 5.9; обозначения те же, что и на рис. 5.7). Последующая установка ведется образовавшимися группами: сначала устанавливается многовыводный элемент, затем остальные элементы его группы. Оптимизация установки элементов будет описана в § 5.5. Здесь отметим только, что данный подход может быть исследован на применимость к автоматическому размещению при проектировании схем, содержащих как ИС различных размеров, так и дискретные элементы.

5.3. ОПТИМИЗАЦИЯ ПОРЯДКА УСТАНОВКИ МИКРОСХЕМ

Вернемся снова к размещению ИС на плате. На основании анализа, проведенного в § 5.1, можно предложить следующий подход к размещению.

Гиперграф $G(X, U; R)$, описывающий электрическую схему проектируемой платы, разбивается на группы максимально связанных между собой элементов по алгоритму, описанному в § 3.4. При разбиении учитываются только короткие цепи, содержащие до 13 элементов. На следующем шаге строится дерево максимально связанных групп и определяется его планарный вид. После определения планарности по коэффициентам связности определяется приоритетный порядок установки групп и элементов в группе. На расширенную модель посадочных мест элементы устанавливаются максимально связными группами. При этом определяется оптимальное расположение групп друг относительно друга и элементов в пределах групп. При этом учитывается возможная конфигурация вновь порождаемых трасс. При определении предположительного места расположения для устанавливаемой группы относительно ранее установленных учитываются связи между ними. После установки всех групп вычисляются координаты установки всех элементов относительно начала координат. Укрупненная блок-схема приведена на рис. 5.10.

Рассмотрим порядок установки групп и установки элементов в пределах группы.

Результатом работы алгоритма разбиения гиперграфа $G(X, U; R)$ на максимально связанные группы является список максимально связанных групп. Этот список удобно представить в виде дерева, у которого вершинами служат образовавшиеся группы, а ребра отражают процесс образования этих групп. Для определения оптимального порядка установки необходимо найти планарное представление этого дерева.

Для определения планарности может быть применен алгоритм обратного обхода. Обход начинается с вершины дерева, соответ-

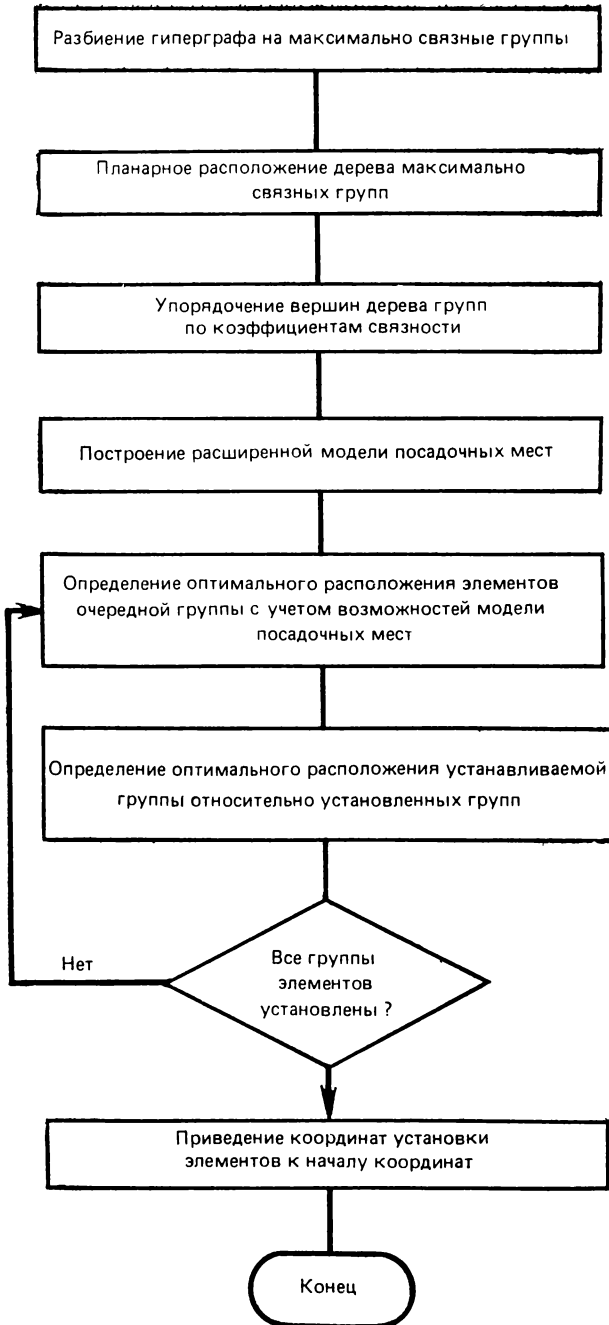


Рис. 5.10

ствующей всей схеме, и продолжается по самой крайней, например левой, ветви дерева. Двигаясь постоянно по самой левой ветви дерева, последовательно запоминаем все вершины, где происходит ветвление, т. е. все вершины со степенью более двух. На некотором шаге достигаем висячей вершины. По последней ветви возвращаемся к первой встретившейся при возвращении вершине ветвления и двигаемся от нее по самой левой непросмотренной ветви. Достигнув висячей вершины, ставим ее сразу справа от последней найденной висячей вершины и снова возвра-

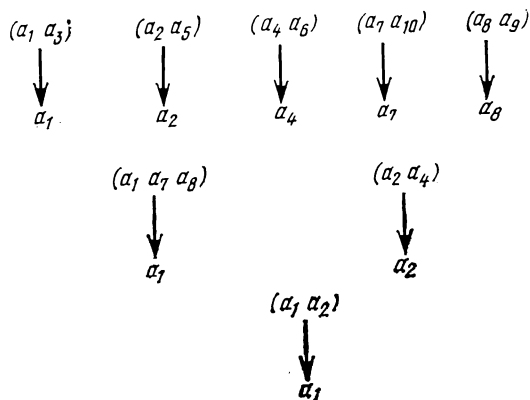


Рис. 5.11

щаемся к первой встретившейся вершине ветвления и т. д. При таком подходе можно достигнуть снова первоначальной вершины, от которой начиналось движение по самому левому непросмотренному ребру. Просматривая и переставляя таким образом вершины, получаем планарное представление дерева. Процесс нахождения планарного представления дерева аналогичен процессу нахождения пути в лабиринте.

В качестве примера рассмотрим процесс обработки списка групп. На рис. 5.11 показан процесс образования максимально связанных групп с отображением группы после образования в вершину с наименьшим номером в группе, чему соответствует следующее списочное представление: $(a_1 a_3)$ $(a_2 a_5)$ $(a_4 a_6)$ $(a_7 a_{10})$ $(a_8 a_9)$ $(a_1 a_7 a_8)$ $(a_2 a_4)$ $(a_1 a_2)$ (a_1) .

При работе со списочным представлением образовавшихся групп в списке берется вершина a_1 и, начиная с правого края списка, группы просматриваются с целью обнаружения группы, содержащей вершину a_1 — в данном примере это группа $(a_1 a_2)$. Из этой группы берется вершина a_2 и список просматривается дальше аналогичным образом. Действуя по такому алгоритму, мы находим вершину a_6 , которая содержится в группе $(a_4 a_6)$, но более не встречается при дальнейшем просмотре списка до его левого края. Этим самым мы получили самую правую ветвь

$\bar{a}_1 - a_2 - a_4 - a_6$. Возвращаемся к последней непросмотренной до конца в списке вершине a_4 и просматриваем дальше список аналогичным образом. При первичной обработке группы для дальнейшего просмотра выбирается последний в группе элемент, при второй обработке — предпоследний и т. д. Результат работы алгоритма для рассматриваемого примера показан на рис. 5.12.

Получив планарное представление дерева, определим порядок реализации ветвей, сходящихся у корня дерева. Анализ результатов работы алгоритмов разбиения вершин гиперграфа на максимально связанные группы показал, что наиболее приемлемым для

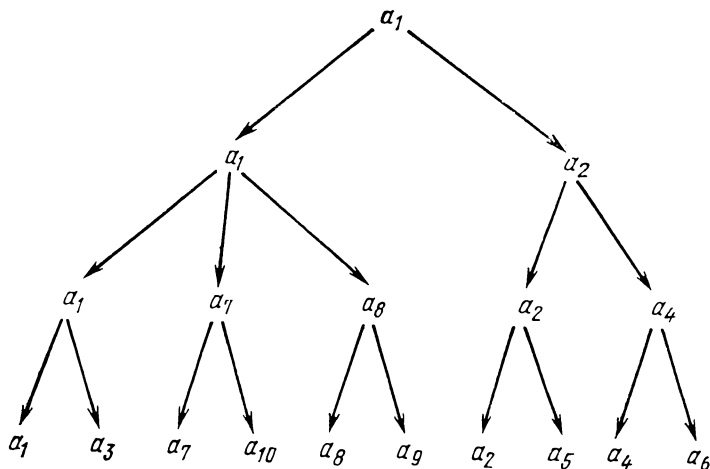


Рис. 5.12

реальных схем являются алгоритмы 2 и 3, описанные в § 3.4. Рассмотрим результат работы алгоритма 2, в котором для большей эффективности дальнейшей реализации оптимального расположения элементов в группе ограничиваем число вершин в одной группе ($m \leq 4$). Максимально связанные группы образуются при этом на разных этапах как изолированными, так и постепенно наращивающимися.

Так как для размещения большей группы элементов необходимо больше посадочных мест, то наиболее естественным для начала установки является выбор из всех ветвей, сходящихся у корня дерева, той, которая содержит в себе большее число групп и элементов. В последнюю очередь следует размещать элементы, соединяющиеся на последних этапах. Такой метод позволяет меньше распылять связанные группы на плате, что выгодно отличает его от других методов определения последовательности установки элементов.

Для определения порядка установки групп в пределах одной ветви и элементов в пределах групп проделаем следующую процедуру упорядочения по коэффициенту связности ρ . Пусть при

образовании групп из элементов получились значения $\rho_{1i_1}, \rho_{1i_2}, \rho_{1i_3}, \dots, \rho_{1i_{n_1}}$. Предположим, что при последующем образовании в группу объединились вершины, соответствующие $\rho_{1i_1}, \rho_{1i_3}, \rho_{1i_4}$, и получилась вершина третьего уровня $x_{1,3}$. Упорядочиваем значения коэффициентов связности по возрастанию: предположим $\rho_{1i_3} < \rho_{1i_4} < \rho_{1i_1}$. Считаем $x_{1,3}$ корнем дерева подграфа с вершинами $\{x_{i,3}\}$, переставляем полностью ветвь, соответствующую

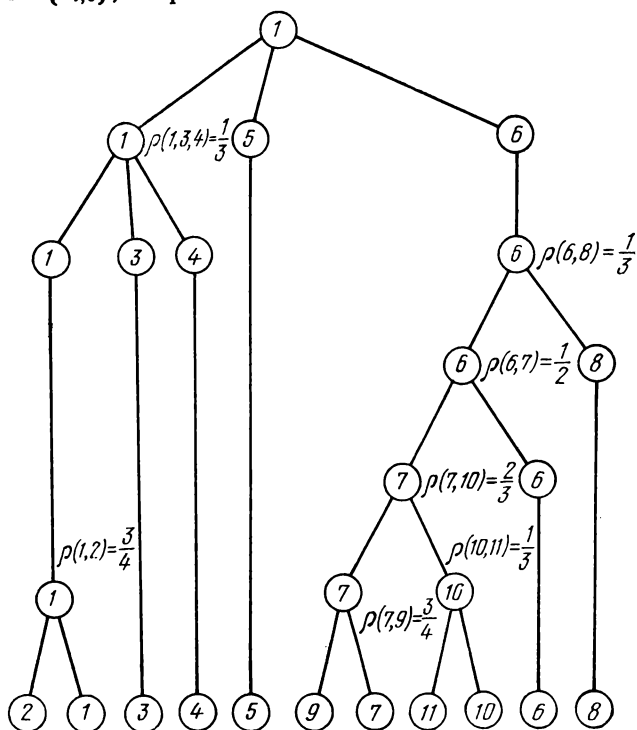


Рис. 5.13

щую ρ_{1i_3} на самую левую позицию; на следующем шаге переставляем на соседнюю позицию ветвь, соответствующую ρ_{1i_4} , и на последнюю позицию ветвь, соответствующую ρ_{1i_1} .

Проделав подобную операцию со всеми вершинами третьего уровня, переходим к вершинам четвертого уровня и т. д., пока не достигнем корня дерева. На рис. 5.13 показано планарное представление упорядоченного дерева максимально связанных групп, соответствующее примерам, приведенным на рис. 3.15 и рис. 3.16.

При оптимизации порядка установки элементов по результатам разбиения на группы по алгоритму 3 (§ 3.4) процедура упрощается. Конечный результат работы этого алгоритма представляется в виде дерева с правершиной и всех листьев на первом

уровне. Здесь стоит задача упорядочения листьев по коэффициентам связности ρ , которая решается так же, как и в предыдущем случае, при вращении ветвей деревьев вокруг правершин различного уровня. Оптимизация порядка установки элементов в пределах группы проводится по связям в пределах группы.

5.4. РАЗРАБОТКА ФУНКЦИИ ОПТИМИЗАЦИИ И ЕЕ СВЯЗЬ С ТРАССИРОВКОЙ

Основной целью при разработке функции оптимизации является максимальное приближение учитываемых условий при установке элементов к реальным условиям реализации связей на этапе трассировки. Кроме того, так как мы устанавливаем элементы максимально связными группами и оптимизируем их размещение, то следует при установке группы учитывать в первую очередь конфигурацию и длину связей S_2 (см. § 3.3) в пределах группы, а при определении взаимного положения групп оптимизировать конфигурацию связей S_3 .

Более точные оценки размещения, чем упоминаемые в § 5.1, должны учитывать возможные изменения конфигурации связей при установке очередного элемента (или оценке перестановки элементов для итерационных алгоритмов) из-за имеющихся или предполагаемых препятствий.

Для вывода функций оптимизации обратимся к модели платы, описанной в § 3.2 (рис. 3.11). Пронумеруем граничные пространства на модели; горизонтальные сверху вниз: $g_1, g_2, g_3, \dots, g_{2n}$ — вертикальные слева направо v_1, v_2, \dots, v_m . Изменение конфигурации проводников при укладке относительно построенного минимального дерева с увеличением длины происходит, если ветвь проходит через дискреты пересечения вертикальных и горизонтальных границ: $\{g_i\} \cap \{v_j\}$. Горизонтальные границы образуются каждым из двух рядов выводов, поэтому недостаточно учитывать связи только между ИС, необходимо принимать во внимание и принадлежность контактов рассматриваемой цепи к конкретному ряду выводов, т. е. для различных вариантов связей между двумя ИС (рис. 5.14,а) может быть оптимальным один из четырех вариантов взаимного расположения двух элементов. Оптимальным следует считать такое расположение элементов, когда число пересечений связей с горизонтальными границами в принятой ориентации будет минимальным с одновременной минимизацией суммарной взвешенной длины, порождаемой устанавливаемым элементом (группой элементов), связей. При равенстве этих условий для разных позиций установки предпочтение следует отдавать вертикальным связям.

Обозначив через B множество контактов верхнего ряда выводов ИС, а через H — нижнего, получаем, что для связей $S'_{AB} = \{B_A, B_B; H_A, H_B\}$ оптимальным будет расположение элемента B либо правее либо левее элемента A (рис. 5.14,б); для связей $S''_{AB} = \{H_A, B_B\}$ —

расположение A непосредственно над B (рис. 5.14, θ); для связей $S''_{AB} = \{B_A, H_B\}$ — наоборот, B над A (рис. 5.14), z .

Несмотря на возможное увеличение числа пересечений трасс, при таком расположении связей S''_{AB} и S'''_{AB} получается более рациональное использование поля трассировки, так как больше загружается вертикальное граничное пространство и меньше горизонтальное. Отсюда следует, что при размещении очередного элемента или группы элементов необходимо добиваться максимума числа связей, соединяющих горизонтальные границы одного ствола, отдавая при этом предпочтение вертикальному расположению модулей, что можно обеспечить, включив в оптимизируемую функцию и длину соединений.

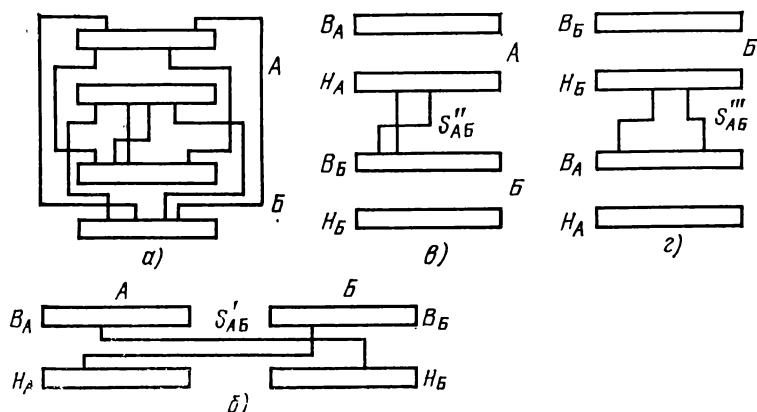


Рис. 5.14

Итак, при установке очередного элемента α_i в вертикальное граничное пространство v_j мы можем выделить четыре типа связей элемента α_i с ранее установленными границами в пределах одного вертикального граничного пространства — S^1 ; такие связи при укладке практически не удлиняются; 2) связи в пределах одного вертикального граничного пространства, которые пересекают горизонтальные границы — S^2 ; для реализации таких связей необходимо выйти по горизонтальным стволам в вертикальный ствол для соединения, т. е. одновременно с увеличением длины возрастает и число занимаемых каналов до 3; 3) связи, соединяющие горизонтальные границы одного ствола, но выходящие за пределы одного вертикального граничного пространства — S^3 , как и связи типа S^1 ; эти связи практически не удлиняются при укладке и для них не требуется дополнительных каналов, так как переход из канала в канал или подход к контактным площадкам осуществляется в вертикальном граничном пространстве, т. е. в таких случаях достаточно учитывать только длину соединения; и наконец, 4) остальные связи — S^4 , которые пересекают горизон-

тальные границы и соединяют контакты, лежащие в разных вертикальных граничных пространствах. Связи типа S^4 при реализации практически не удлиняются, однако для них требуется три различных канала, как и для связей S^2 ; трассы таких связей образуют дополнительные точки перегибов.

В процессе установки множество вершин гиперграфа $G(X, U; R)$ можно подразделить на два непересекающихся подмножества: $X = X_1^i \cup X_2^i$, где X_1^i — множество вершин, соответствующих установленным элементам на плате после размещения i -й группы; X_2^i — множество вершин, соответствующих еще не установленным после i -го шага элементам.

Обозначим X^i группу вершин, соответствующих установке элементов на i -м шаге. Соединитель считается установленным до начала размещения, т. е. X^0 содержит соединитель. Таким образом, $X_1^i = \bigcup_{k=0}^i X^k$.

При данном разбиении множества вершин множество ребер разлагается по формуле $U = U_1^i \cup U_2^i \cup U_3^i$, где U_1^i , U_2^i , U_3^i — множество ребер, инцидентных вершинам соответственно только из X_1^i , из X_2^i и из X_1^i и X_2^i одновременно. Множество U_3^0 соответствует цепям, содержащим соединитель.

В процессе установки происходит следующее преобразование множеств. После завершения установки $(i-1)$ -й группы элементов формируются множества $X_1^{i-1} = X_1^{i-2} \cup X^{i-1}$ и $X_2^{i-1} = X_2^{i-2} \setminus X^{i-1}$. Из множеств U_2^{i-1} и U_3^{i-1} выбираются те ребра, которые стали инцидентны только вершинам из $X_1^i = X_1^{i-1} \cup X^i$ и формируется множество U_1^i . Из множества U_2^{i-1} выбираются такие ребра, которые стали инцидентны множеству вершин $X_1^i = X_1^{i-1} \cup X^i$, и остались инцидентны множеству $X_2^i = X_2^{i-1} \setminus X^i$. Эти ребра переносятся в множество U_3^{i-1} . Выполнив описанные выше действия, мы получаем множества U_1^i , U_2^i , U_3^i .

Для определения качества взаимного расположения групп элементов, равно как и элементов внутри группы, использовалось понятие расстояния между множествами. Обозначив через A множество установленных элементов за $(i-1)$ шаг, через B — группу элементов, устанавливаемых на i -м шаге, определим функцию качества размещения группы B на предполагаемых позициях

$$F(A, B) = \sum_{u \in U_3^i} \min_{\substack{a \in A, b \in B \\ R(a, u) \cdot R(b, u) = 1}} (\rho(a, b) + \gamma(a, b)), \quad (5.1)$$

где $\rho(a, b) = |x_a - x_b| + |y_a - y_b|$, где, в свою очередь, (x_a, y_a) и (x_b, y_b) — координаты вершин выводов на плоскости;

$$\gamma(a, b) = \begin{cases} 0, & \text{если } u \in \{S^1, S^2\}, \\ 2l, & \text{если } u \in \{S^3, S^4\}, \end{cases}$$

где l — усредненная ширина вертикального граничного пространства, что обусловлено дополнительной загрузкой каналов.

Заметим, что предлагаемая для минимизации функция качества размещения $F(A, B)$ хорошо согласуется с требованием минимизации длины соединений и отвечает одному из основных принципов построения проводников при трассировке — соединять эквипотенциальные точки кратчайшим путем. Вместе с тем в ней учитывается и возможное удлинение трасс на этапе прокладки проводников, а также предположительное число занимаемых каналов. Таким образом, можно сделать вывод о согласовании критерия размещения с трассировкой, так как предопределение длины и конфигурации трасс определяет поведение целевой функции проектирования $\Phi(\cdot)$ в каждой локальной области уже при установке элементов.

5.5. ПРИМЕНЕНИЕ РАСШИРЕННОЙ МОДЕЛИ ПЛАТЫ ПРИ УСТАНОВКЕ ЭЛЕМЕНТОВ

Чтобы обеспечить лучшие условия для трассировки, одной из локальных целей при размещении ставится задача минимизации связей S_1 в различных областях платы (см. § 3.3). Одним из возможных способов минимизации числа таких связей является установка наиболее связанных между собой элементов в односвязной области Q . Для этого необходимым условием является существование такой траектории обхода мест установки элементов рассматриваемой в данный момент группы на графе посадочных мест, которая начиналась и кончалась бы на элементах данной группы и могла бы быть проложена только по элементам этой группы. Построение траектории должно осуществляться только по образующим ортогональной сетки, проходящей через посадочные места элементов.

Из сформулированного выше следует, что при определении перспективных мест для установки очередных элементов, сильно связанных между собой, достаточно просматривать только те места, которые будут смежными с местами ранее установленных элементов из рассматриваемой группы. Кроме того, для обеспечения большей свободы для очередной устанавливаемой группы элементов может возникнуть необходимость «сдвинуть» на плате ранее установленные элементы в ту или иную сторону на несколько позиций. Рассмотрим пример размещения группы из четырех элементов. Для оптимизации размещения в пределах группы и группы относительно установленных элементов желательно сдвинуть все установленные элементы сначала, например, вверх, оценить результат пробной установки, затем сдвинуть те же элементы вниз, снова оценить полученное размещение, затем выбрать лучшее.

Избежать просмотра вариантов сдвига можно применением расширенной модели посадочных мест и метода блуждающего трафарета, описанного в § 3.2. Заметим, что при применении метода блуждающего трафарета для оптимизации расположения БЭ в корпусе микросхемы на этапе компоновки координаты БЭ не

имели значения — важен был лишь порядок элементов в корпусе. При решении задачи размещения элементов на плате после установки всех элементов следует осуществить сдвиг полученной модели в принятое при проектировании начало координат.

Кроме того, в отличие от размещения БЭ, которые устанавливались в корпус индивидуально, при размещении элементов на плате с использованием метода блуждающего трафарета предлагается учитывать принадлежность элементов к максимально связным группам. При этом необходимо анализировать возможности установки группы в оптимальной конфигурации, т. е. должно выполняться условие минимизации функции $F(A, B)$ (5.1).

Указанные отличия и требования формируют следующий подход. После установки очередной группы элементов на расширенную модель выбирается наиболее связанная с уже установленными за $(i-1)$ -й шаг элементами группа максимально связанных элементов G_i . Если группа состоит из одного элемента, то проводится непосредственно установка этого элемента на позицию по выбранному критерию (5.1). Если группа состоит более чем из одного элемента, то определяются связи группы G_i с элемен-

тами $\bigcup_{j=1}^{i-1} G_j$:

$$S\left(G_i, \bigcup_{j=1}^{i-1} G_j\right) = \bigcup u \left(a \in G_i, b \in \bigcup_{j=1}^{i-1} G_j, R(a, u) \cdot R(b, u) = 1 \right).$$

На следующем шаге определяются контакты, принадлежащие этим связям и лежащие на границах уже установленных элементов: v_n, v_l, g_n, g_v , и выбирается граница, содержащая максимальное число таких контактов (v_n, v_l — соответственно правая и левая вертикальная граница, g_n, g_v — соответственно нижняя и верхняя горизонтальная граница).

В оговоренной ранее ориентации принято, что основной соединитель расположен на левой границе платы, поэтому, если очередная группа G_i претендует на установку у левой границы v_l , то к оценочной функции $F(A, B)$ (5.1) добавляется слагаемое $\Delta x \cdot p$, где Δx — расстояние, на которое отодвигается влево соединитель, а p — число связей элементов из множества $\bigcup_{i=1}^{i-1} G_j$ с со-

единителем. Таким образом, оценочная функция имеет вид

$$F(A, B) = \Delta x \cdot p + \sum \min(\rho(a, b) + \gamma(a, b)). \quad (5.2)$$

Исходя из этого, если при установке выбрана граница v_l , то после установки группы проводится пробное размещение и у следующей по числу контактов границы, после чего из полученных двух результатов выбирается лучшее расположение элементов.

Для определения допустимой конфигурации установки группы элементов G_i удобно использовать расширенную матрицу $S(2m-1, 2n-1)$ при размерах модели посадочных мест $m \times n$.

В этой матрице нули будут обозначать допустимые для установки места; различными кодами, например номером помещаемого в данную позицию элемента будем обозначать занятые места, а числом, превышающим максимальный номер устанавливаемых элементов, места, которые выходят за пределы блуждающего трафарета после установки очередного элемента.

Рассмотрим на примере применение метода блуждающего трафарета при последовательно-групповом размещении элементов. Пусть $m=4$, $n=3$, число элементов $k=10$. Последовательность установки следующая: группа из четырех элементов — $G_1=\{8, 9, 4, 1\}$; группа из двух элементов — $G_2=\{5, 7\}$; группа из трех элементов — $G_3=\{2, 6, 10\}$; группа из одного элемента $G_4=\{3\}$. Расширенная модель установочных мест платы, соответствующая указанным размерам, приведена на рис. 5.15; штриховыми линиями на рисунке нанесены крайние положения трафарета, а точками — посадочные места.

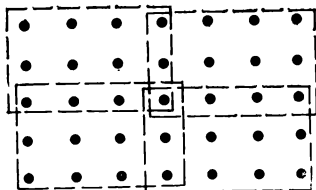


Рис. 5.15

Допустим, что после оптимизации элементы первой группы расположились по квадрату. Проверяем по матрице C допустимость размещения в центре модели такой конфигурации в пределах трафарета. Если конфигурация допустима, то помещаем квадрат, полученный при оптимизации в пределах группы, левым нижним углом в центр расширенной модели. Можно помещать в центр любой элемент квадрата — от этого конфигурация свободных мест установки для последующих элементов не изменится. При этом только сдвигаются ограничивающие единицы в матрице и изменяются координаты запрещенных мест на расширенной модели.

Допустим, что после оптимизации элементы первой группы расположились по квадрату. Проверяем по матрице C допустимость размещения в центре модели такой конфигурации в пределах трафарета. Если конфигурация допустима, то помещаем квадрат, полученный при оптимизации в пределах группы, левым нижним углом в центр расширенной модели. Можно помещать в центр любой элемент квадрата — от этого конфигурация свободных мест установки для последующих элементов не изменится. При этом только сдвигаются ограничивающие единицы в матрице и изменяются координаты запрещенных мест на расширенной модели.

На рис. 5.16,а квадратами на позициях показаны установленные элементы группы, а штриховой линией — возможные крайние положения трафарета после установки группы из четырех элементов; на рис. 5.16,б представлена соответствующая расширенная модель.

На следующем шаге устанавливается группа из двух элементов. При установке группы делаются перестановки в пределах группы с определением в каждом случае оценочной функции. Учитывая, что при определении планарности дерева максимально связанных групп следующие друг за другом группы связаны между собой, то перспективными для установки позициями следует считать близлежащие к установленной группе. Если вторая группа займет позиции $c_{2,6}$ и $c_{3,6}$, то второй столбец расширенной матрицы будет состоять из одних запрещающих единиц:

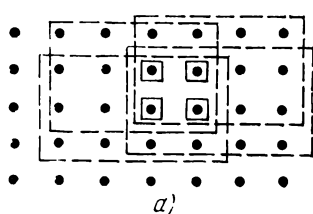
99	99	0	0	0	0	0	0
99	99	0	8	1	5	0	0
99	99	0	4	9	7	0	0
99	99	0	0	0	0	0	0
99	99	99	99	99	99	99	99

При установке группы из трех элементов просматриваются свободные места: вертикальные — слева или справа от ранее установленных, горизонтальные —

выше или ниже относительно установленных элементов; либо просматривается расположение группы G_3 углом.

При горизонтальном расположении строго выше или ниже расположенных элементов трафарет может сдвигаться на позицию влево либо вправо. При всех остальных положениях группы G_3 последующие свободные места для установки последнего элемента определяются однозначно.

Как было показано в § 3.2 и видно из рассмотренного примера, число посадочных мест для установки очередной группы быстро уменьшается. Применение метода блуждающего трафарета при последовательно-групповом размещении позволяет избавиться от сильной зависимости выбора позиции для устанавливаемых эле-



<i>99</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>
<i>99</i>	<i>0</i>	<i>0</i>	<i>8</i>	<i>1</i>	<i>0</i>	<i>0</i>
<i>99</i>	<i>0</i>	<i>0</i>	<i>4</i>	<i>9</i>	<i>0</i>	<i>0</i>
<i>99</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>	<i>0</i>
<i>99</i>	<i>99</i>	<i>99</i>	<i>99</i>	<i>99</i>	<i>99</i>	<i>99</i>

Рис. 5.16

ментов от ранее установленных, т. е. размещение очередной группы в меньшей мере влияет на возможности размещения последующих групп. Метод блуждающего трафарета применим и при размещении блоками, что позволяет избежать перестановки блоков для улучшения размещения.

Предложенный метод имеет преимущество и перед методом последовательного разрезания платы на части, так как в последнем при заранее определенных сечениях возможно разрезание сильно связанных элементов группы и размещение их в разных частях платы. Исследования проводились на платах, содержащих 35—40 микросхем. При ограничении числа элементов в группе $m \leq 4$ получалось около 15 образовавшихся групп. При описанном выше подходе элементы не более двух групп располагались вдали друг от друга, в то время как при размещении дихотомическим методом «разрывалось» 5—6 групп.

Достоинство метода блуждающего трафарета состоит еще и в том, что его применение позволяет в значительной степени устранить один из основных недостатков последовательных алгоритмов — размещение последних устанавливаемых элементов в значительно худших позициях. Расширение модели платы применимо и при размещении разногабаритных элементов. В этом случае увеличиваются размеры самой платы по описанным выше правилам.

Особый интерес представляет собой применение метода блуждающего трафарета при размещении разногабаритных элементов группами, получаемыми в результате описания схемы устройства

графовой моделью (см. § 5.2). Недостатком такого метода (по имеющимся публикациям) является размещение группами без оптимизации размещения элементов в пределах группы.

Описанная в § 5.2 графовая модель позволяет определить порядок расположения групп на плате и желательное расположение их друг относительно друга. После определения места для расположения очередной размещаемой группы элементов целесообразно определить занимаемый ею размер площади и ее конфигурацию. На основе этой части платы необходимо построить расширенную модель этой локальной площади установки. Связи размещаемых элементов с уже размещенными можно учитывать как связи элементов с соединителем и с учетом этих условий на локальной расширенной модели размещать элементы данной группы.

После размещения на локальной части полученное расположение элементов группы переносится на первоначальную модель платы. Если для устанавливаемой группы нет места для размещения в оптимальной конфигурации, то либо деформируют локальную модель, либо осуществляют сжатие ранее установленных элементов, либо группу разрывают и устанавливают в разных несвязных областях платы. Последнее на практических схемах получается при плотной установке и с 2—3 последними устанавливаемыми группами.

Заметим, что изложенный метод может успешно использоваться при размещении на плате и ИС и дискретных элементов одновременно. В этом случае в группу включаются вместе с ИС и связанные с ними дискретные элементы. На локальной расширенной модели легче размещать ЭРИ между ИС на свободных местах, чем после расположения всех ИС, как это делается в известных по публикациям алгоритмах.

5.6. ОСОБЕННОСТИ РАЗМЕЩЕНИЯ ЭЛЕМЕНТОВ ПРИ ПРОЕКТИРОВАНИИ ЧЕРТЕЖА ЭЛЕКТРИЧЕСКОЙ СХЕМЫ

Наиболее сложной задачей при проектировании чертежа принципиальной электрической схемы является размещение элементов (по сравнению с задачами, решаемыми при проектировании печатной платы). Размещение элементов на чертеже имеет свою специфику. Согласно рекомендациям ГОСТа элементы должны располагаться так, чтобы максимально отражать порядок прохождения сигнала слева направо. Условием, облегчающим решение задачи, является возможность пересечения проводников, их жгутование, т. е. сведение в линию групповой связи, а также возможность располагать контакты соединителя не по порядку их нумерации. Входные контакты соединителя должны располагаться слева, а выходные — справа, вне зависимости от их физической принадлежности к тому или иному соединителю на плате.

Так как проблема вычерчивания автоматически принципиальных схем решается после решения основных задач проектирования — компоновки, размещения и трассировки, то необходимо использовать ту же информацию, стараясь

минимально увеличить объем вновь вводимой. В отличие от размещения конструктором, при автоматическом вычерчивании ИС располагаются колонками.

В начальный момент рассчитывают рабочее поле чертежа. При этом из размеров принятого формата вычитается горизонтальная полоса, равная по высоте левому угловому штампу, а также вычитаются вертикальные полосы слева и справа, необходимые для размещения выводов соединителя. Оставшаяся часть предназначается для размещения элементов схемы и прокладки связей. Исходя из графических размеров изображения элементов, учитывая необходимые пространства для прокладки линий групповых связей вместе с необходимой отцифровкой, проводится расчет числа колонок.

Если число возможных колонок больше, чем это необходимо при плотной установке элементов, то делают перерасчет с целью предоставления большей свободы размещения и будущей прокладке связей. Зная число возможных занимаемых колонок и их усредненную высоту, выделяются группы элементов по числу колонок. Выделение групп элементов основано на нахождении порядка прохождения сигнала. При этом за основу берутся цепи, содержащие два вывода. По соотношению входов и выходов в цепях определяется левая и правая части соединителя. Первая группа формируется из элементов, связанных входами с соединителем (здесь и далее, если не оговорено особо, речь идет о двухконтактных связях). Последняя группа формируется из элементов, связанных выходами с соединителем. Вторая группа формируется из элементов, связанных своими входами с выходами элементов первой группы. Предпоследняя группа формируется из элементов, связанных своими выходами со входами последней группы и т. д. Так как число групп ограничено расчетным числом колонок, то в некоторый момент возможна «встреча» групп, тогда распределение по колонкам продолжается в уже начавшие заполняться группы.

При таком заполнении возможны несколько случаев. Рассмотрим некоторые из них. Так как высота колонки ограничена, то возможно, что не все распределенные в нее элементы могут поместиться. Для решения этой конфликтной ситуации при распределении привлекается мера принадлежности, определяемая как функция от числа входов и выходов элемента, от числа задействованных в данных цепях выводов, от их расположения относительно ранее распределенных элементов. При «перегрузке» колонки из нее убирается претендент, имеющий наименьшую меру принадлежности к этой колонке. Он переносится в колонку, на которую он претендует с большей из оставшихся мер.

Если после просмотра схемы по двухконтактным связям по колонкам распределены не все элементы, то проводится распределение по колонкам на основе рассмотрения трех и четырех контактных связей, а затем и по оставшимся связям. Такой порядок позволяет распределить все элементы по колонкам, одновременно сохраняя последовательность прохождения сигнала, которую более вероятно начертить без заведения соответствующих цепей в групповую линию связи.

После распределения всех элементов по колонкам переходим к построению кортежей. Из первой группы берется элемент, и для него находится элемент из второй колонки, который связан с его выходами своими входами, из третьей колонки берется аналогичным образом элемент, связанный по этому же принципу с элементом из второй колонки и т. д. Затем снова берется элемент из первой колонки и определяется последовательность влекомых им элементов. После того, как будет исчерпана первая колонка, переходим к построению кортежей

от последней колонки. Здесь элементы выбираются инверсным способом. Если по двухконтактным цепям не все элементы вошли в кортежи, то цепи просматриваются по возрастанию числа контактов, чтобы найти наиболее благоприятное включение элемента в кортеж. Заметим, что один элемент может повлечь за собой два и более элементов в следующей колонке (рис. 5.17). После построения кортежей можно приступать к размещению элементов в колонках. Для размещения целесообразно расширить модель вдвое только по высоте.

На следующем этапе выбирается наиболее «длинный» кортеж и устанавливается в центр по высоте расширенной модели. Первый элемент при этом устанавливается непосредственно в центр своей колонки, второй — в следующей ко-

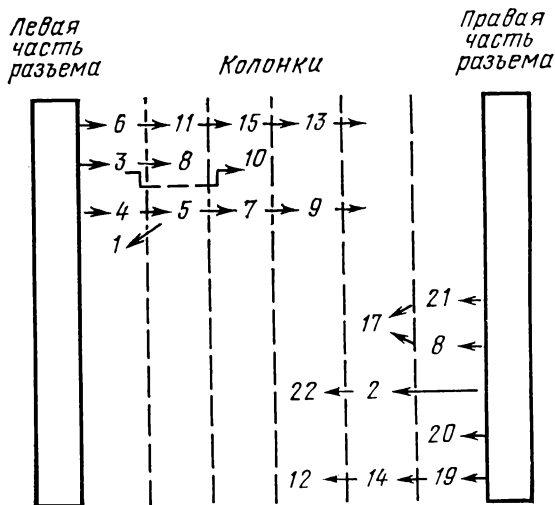


Рис. 5.17

лонке таким образом, чтобы максимальное число двухконтактных связей реализовывалось в виде прямых линий. Затем ставится элемент из того же кортежа в следующую колонку по такому же принципу и т. д.

После размещения всех элементов первого кортежа выбирается второй кортеж для размещения. Критерием выбора второго кортежа является максимальное число вертикальных связей с первым кортежем. Здесь целесообразно учитывать не только двухконтактные, но и трех-четырёхконтактные связи, которые могут быть реализованы между двумя рядами выводов колонок. При последующем размещении элементов по описанным выше принципам необходимо проверять методом блуждающего трафарета возможность установки кортежей в оптимальной конфигурации. Трафарет в данном случае перемещается только по высоте. Последние кортежи, возможно, будут разорваны, т. е. элементы в них не будут следовать один за другим. Вычерчивание контактов соединителя делается напротив задействованных выводов элементов, как это разрешено ГОСТом.

Окончательное расположение элементов на чертеже определяется в процессе прокладки трасс. По анализу загрузки вертикального пространства делается попытка прокладки без включения в групповую линию связи цепей с малым

числом контактов. При этом, вероятно, возможно провести вертикальные связи, используя как свободное пространство из-за различной ширины графического изображения элементов одной колонки, так и раздвижение колонок.

Попытки реализовать максимальное число связей без введения в групповую линию связи оправданы большей наглядностью получаемого чертежа. Более глубокий анализ загрузки поля чертежа и связей позволяет организовывать локальные групповые линии связи.

Вопрос целесообразности и эффективности включения в САПР подсистемы вычерчивания принципиальных электрических схем будет рассмотрен в гл. 7.

Глава 6

ПАРАЛЛЕЛЬНЫЙ МЕТОД ПОФРАГМЕНТНОЙ ПРОКЛАДКИ ТРАСС

61. ОБЩАЯ СХЕМА ПОФРАГМЕНТНОГО ПОДХОДА К ТРАССИРОВКЕ

Результаты, полученные существующими методами и алгоритмами трассировки, имеющимися в эксплуатации, уступают результатам опытного конструктора по таким параметрам, как плотность печатных соединений на слое, число переходных отверстий со слоя на слой при проектировании двухслойных печатных плат и т. п. Трудности алгоритмизации действий конструктора заставляют разработчиков САПР искать новые методы и алгоритмы различных этапов автоматизированного проектирования печатных плат, модифицировать имеющиеся алгоритмы. Основной задачей при этом остается трассировка соединений.

Большинство известных алгоритмов трассировки предполагает последовательную реализацию связей на плате, что отрицательно сказывается на конечных результатах проектирования, так как при этом невозможно учесть все потребности последующих прокладываемых трасс. Частично этот вопрос решается различными модификациями волнового алгоритма, блокировкой контактов модулей, перекладыванием отдельных проводников и другими методами.

Одной из проблем при последовательной трассировке является выбор очередного соединения для реализации, что влияет как на число непроведенных связей, так и на число переходных отверстий со слоя на слой. В связи с этим, а также вследствие интеграции и миниатюризации аппаратуры этот метод, как не отвечающий возмозностям полного использования ресурсов коммутационного поля, отходит на второй план. В последнее время более широко начали разрабатываться и использоваться параллельные методы трассировки.

Для устранения указанных недостатков разработан пофрагментный подход к решению задачи трассировки, который относится к параллельным методам. При таком подходе сначала определяется возможность реализации проектируемой схемы на заданном типоразмере платы после размещения на ней элементов, при этом распределяются по стволам фрагменты связывающих деревьев, оптимизируется порядок их реализации, после чего проводится непосредственно прокладка трасс по фрагментам.

В основе предлагаемого метода лежит построение квазиоптимального связывающего дерева на ортогональной сетке по описанному в § 3.5 алгоритму. Так как при построении леса не учитываются возможные препятствия на пути проводников, то следующим шагом является укладка ветвей связывающих деревьев на коммутационное поле платы. При решении задачи укладки используется модель платы в виде стволов и границ, описанная в § 3.2. Укладка ведется с использованием теории расплывчатых множеств.

При предварительном распределении ветвей деревьев по стволам каждой ветви приписывается некоторая мера принадлежности к какому-нибудь стволу. Исключение составляют ветви, лежащие в граничном пространстве полностью и совпадающие с его ориентацией; при этом должно соблюдаться условие непересечения границ ортогонального направления. При определении меры принадлежности учитываются такие факторы: вершины, инцидентные разносимой ветви, ориентация ветви относительно ориентации ИС на плате, влияние ранее разнесенных ветвей, ветви, смежные разносимой, расстояние от вершин, инцидентных рассматриваемой ветви, до границ, а также другие факторы.

После распределения всех ветвей связывающих деревьев леса проводится анализ загрузки стволов платы для чего строятся матрицы горизонтальных и вертикальных пропускных способностей стволов платы, а также матрицы загрузки горизонтальных и вертикальных стволов.

Анализ может показать, что пропускные способности всех сечений стволов платы позволяют реализовать схему в виде печатных проводников, и тогда перейти непосредственно к трассировке соединений либо к попытке более равномерного распределения ветвей, искусственно понизив предварительную пропускную способность каждого сечения на единицу, а затем перейти к перераспределению ветвей между стволами с учетом новых пропускных способностей.

Если анализ показывает, что значения отдельных элементов матриц пропускных способностей ниже, чем значения соответствующих элементов матриц загрузки, попытаемся корректировать разнесение ветвей.

Корректировка основана на определении минимального односвязного интервала перегрузки, определении множества ветвей, полностью пересекающих интервал перегрузки и имеющих меру принадлежности менее единицы. Из полученного множества вет-

вей выбирают ветвь с минимальной мерой принадлежности и пытаются перенести эту ветвь в соседний ствол. Если соседний ствол перегружен, то необходимо разгрузить его аналогичным образом. Перераспределение делается для всех мест перегрузки. Возможны случаи, когда нет ветвей, полностью пересекающих минимальный интервал перегрузки. Тогда переносится минимальная совокупность ветвей, удовлетворяющая упоминавшимся требованиям.

Альтернативным подходом к корректировке разнесения ветвей во избежание перегрузки стволов может служить деформация сетки посадочных мест элементов без деформации поля трассировки. Такой подход проще реализуется, хотя не всегда дает такие же результаты, как предыдущий. Однако внедрение в практику такого подхода встречает сопротивление со стороны конструкторов, ведущих разработку, а также технологов, особенно если в ряду или колонке сдвигаются только отдельные модули.

Одной из трудностей при решении задачи укладки и коррекции является сохранение связности деревьев. Данную проблему можно решать введением в точках раздвоения ветвей дополнительных вершин и отрезков нулевой длины, раздвоением отдельных ветвей деревьев, что позволяет избежать потери информации при реализации ветвей.

При прокладке трасс, соответствующих ветвям, необходимо ограничивать стволы, чтобы ветви проводились строго в отведенных им стволах. Чтобы минимизировать число пересечений, порядок проведения ветвей в стволе определяют согласно теореме Жордана о разбиении области на две несвязные части [80] — ветви упорядочиваются по возрастанию длины вне зависимости от принадлежности к конкретным цепям.

Общая последовательность прокладки заключается в реализации ветвей сначала одного ствола, например горизонтального направления, затем следующего ствола этой же ориентации и т. д.; после просмотра при первой итерации всех стволов данной ориентации переходим к стволам ортогональной ориентации. В стволе реализуются ветви, соединяющие основные вершины либо основную и дополнительную. Если реализована ветвь, имеющая одну инцидентную дополнительную вершину, то этой вершине присваивается статус псевдоосновной вершины, что необходимо для замыкания алгоритма выбора ветвей. Псевдоосновная вершина при выборе ветвей для реализации считается основной, а ее координаты считаются закрепленными. После просмотра всех горизонтальных и вертикальных стволов повторные просмотры делаются до тех пор, пока либо не будут реализованы все ветви, либо при последней итерации не будет проведен ни один проводник.

Преимущество предлагаемого подхода заключается в более полном использовании коммутационного поля платы, что обеспечивается предварительной укладкой ветвей и последующей коррекцией с учетом меры принадлежности, а также то, что до начала непосредственного проведения трасс определяется возможность реализации всех ветвей на заданном типоразмере платы.

6.2. ОСНОВНЫЕ ПРИНЦИПЫ ПРЕДВАРИТЕЛЬНОЙ УКЛАДКИ ВЕТВЕЙ ДЕРЕВЬЕВ НА КОММУТАЦИОННОЕ ПОЛЕ ПЛАТЫ

При начальном распределении ветвей деревьев по стволам оказывается неудобным жестко закреплять ветвь в каком-либо конкретном стволе, так как в дальнейшем при перегрузке отдельных сечений стволов неясно, какие ветви лучше переносить в другие стволы, чтобы в дальнейшем можно было провести большее число трасс. Для гибкости дальнейшего обращения со связывающими деревьями при окончательном разнесении ветвей деревьев по стволам удобно использовать при предварительной укладке понятие расплывчатости и меру принадлежности (см. § 2.1), что согласуется с тем, что само выдвигаемое требование на каждом шаге принятия решения о проведении того или иного проводника, с условием меньше помешать последующим прокладываемым проводникам, содержит в себе размытые условия прокладки трасс, ибо трассы, проложенные по любому точному критерию, значительно ухудшают, начиная с некоторого момента, условия прокладки последующих проводников. Таким образом, в процессе принятия решения перед нами стоит расплывчатая цель. Определяя для фрагмента меру принадлежности, мы не закрепляем его в стволе окончательно, а только определяем для него «желательный» ствол вместе с мерой «желательности», т. е. на первом этапе мы получаем нечеткое решение и довольствуемся им. Проведенные выше рассуждения являются еще одним примером, показывающим различие между вероятностью и мерой принадлежности.

При определении меры принадлежности для разносимых по стволам ветвей учитывалось: расстояние от вершин, инцидентных разносимой ветви, до границ стволов, что определяет число возможно отсекаемых контактов других цепей; наличие ортогонально подходящих ветвей, что позволяет в дальнейшем уменьшить число переходов со слоя на слой; вершины (основные, дополнительные), инцидентные рассматриваемой ветви, и их сочетания.

Перед распределением ветвей на плате имеются границы стволов и лес квазиоптимальных деревьев на ортогональной сетке.

Как оговаривалось ранее, считаем, что модули ориентированы горизонтально, т. е. ряды выводов модулей расположены параллельно оси абсцисс. Задача заключается в присвоении каждой ветви номера ствола и в определении для каждой ветви меры принадлежности $\mu \in [0, 1]$. Одним из неперемennых условий распределения является сохранение связности. В первую очередь определяются ветви, полностью лежащие в граничном пространстве, и они помечаются особо. Особо помечаются также ветви, полностью лежащие в стволе (для удобства программного обращения с такими ветвями условно принято $\mu_r = 2$ для ветвей, лежащих полностью в граничном пространстве, и $\mu_c = 3$ для ветвей, лежащих полностью в стволе).

Каждой ветви могут быть инцидентны две вершины в следую-

щих сочетаниях: основная — дополнительная (и наоборот); основная — основная и дополнительная — дополнительная. Каждая из вершин может лежать на границе горизонтальной либо вертикальной, может принадлежать одновременно как горизонтальной, так и вертикальной границе, может лежать вне границ.

В зависимости от перечисленных выше сочетаний и условий определяется мера принадлежности.

Рассмотрим следующие случаи.

1. *Ветвь соединяет две основные вершины.*

Горизонтальная ветвь. Если обе вершины лежат на границе и не имеют горизонтальных инцидентных ветвей, кроме рассматриваемой ветви, то мера принадлежности определяется по формулам:

$$\mu_{\text{в}} = 1/2 + \sum_{i, j=1, 2} P_{ij}; \quad \mu_{\text{н}} = 1/2 - \sum_{i, j=1, 2} P_{ij}; \quad (6.1)$$

где $\mu_{\text{в}}$, $\mu_{\text{н}}$ — мера принадлежности соответственно к верхнему и нижнему стволу относительно расположения рассматриваемой ветви; P_{ij} — слагаемое, отражающее наличие ветвей инцидентных i -й вершине и их направленность; если инцидентные ветви расположены выше рассматриваемой ветви, что $P_{ij}=1/4$; если ниже, то $P_{ij}=-1/4$.

Если какой-нибудь из этих вершин инцидентны ранее уже разнесенные горизонтальные ветви, то расчет ведется по формулам

$$\mu_{\text{в}} = \mu_{\text{в}}' + (\mu_{\text{в}}'' - 1/2) \mu_{\text{н}}'; \quad \mu_{\text{н}} = \mu_{\text{н}}' + (\mu_{\text{н}}'' - 1/2) \mu_{\text{в}}'; \quad (6.2)$$

где $\mu_{\text{в}}'$, $\mu_{\text{н}}'$ — мера принадлежности, рассчитанная по формулам (6.1); $\mu_{\text{в}}''$, $\mu_{\text{н}}''$ — мера принадлежности ранее разнесенной инцидентной ветви.

Возможен случай, когда одна из вершин лежит на границе, а другая — в стволе, ограниченном этой же границей. В этом случае ветви приписывается принадлежность к этому стволу с мерой $\mu=1$. Если вершины принадлежат разным горизонтальным стволам, то ветвь разбивается на части, вводятся дополнительные вершины, а разнесение ведется одним из описанных ниже способов. Во введенных таким образом дополнительных вершинах вводится ветвь нулевой длины.

Вертикальная ветвь. Здесь в первую очередь следует выделить ветви, не пересекающие горизонтальных границ, и приписать им меру $\mu_{\text{г}}$ либо $\mu_{\text{с}}$. Если ветвь пересекает хотя бы одну горизонтальную границу, то для ее реализации на плате в виде печатного проводника необходимо эту ветвь вынести в вертикальный ствол за пределы вертикального граничного пространства; при этом существенно изменяется конфигурация первоначально построенного дерева. Если дерево состоит всего из одной ветви, то мера принадлежности определяется по формулам:

$$\mu_{\text{л}} = \frac{b_1 + b_2}{a_1 + a_2 + b_1 + b_2}; \quad \mu_{\text{п}} = \frac{a_1 + a_2}{a_1 + a_2 + b_1 + b_2}; \quad (6.3)$$

где $\mu_{\text{л}}$, $\mu_{\text{п}}$ — мера принадлежности соответственно к левому и правому относительно разносимой ветви стволу; a_1 — расстояние от первой вершины до ближайшего левого ствола; b_1 — расстояние от первой вершины до ближайшего правого ствола; a_2 — расстояние от второй вершины до ближайшего левого ствола; b_2 — расстояние от второй вершины до ближайшего правого ствола.

После определения приоритетной принадлежности к какому-нибудь стволу разносимой вертикальной ветви проводится корректировка дерева по достройке горизонтальных ветвей от вершин к этому вертикальному стволу. (Здесь уместно вспомнить, что на этапе размещения в оптимизируемой функции учитывалось возможное изменение конфигурации трасс.) Горизонтальные ветви получают приоритетные принадлежности к стволам, через которые проходит вертикальная ветвь, т. е. горизонтальная ветвь, отходящая от верхней вершины, будет иметь $\mu_H > \mu_B$ и, если $\mu_H > \mu_L$, то

$$\mu_{H1} = \frac{1}{2} + \frac{1}{2} \cdot \frac{a_1}{a_1 + b_1}; \quad \mu_{B1} = 1 - \mu_{H1};$$

$$\mu_{B2} = \frac{1}{2} + \frac{1}{2} \cdot \frac{a_2}{a_2 + b_2}; \quad \mu_{H2} = 1 - \mu_{B2}; \quad (6.4)$$

где μ_{H1} — мера принадлежности верхней достроенной ветви к нижнему горизонтальному стволу; μ_{B2} — мера принадлежности нижней достроенной ветви к верхнему горизонтальному стволу. В точках перегибов вводятся дополнительные вершины.

Соответствующий пример показан на рис. 6.1, где показана ветвь до разнесения (а) и разнесенная ветвь (б); стрелки направлены в сторону приоритетной принадлежности ветвей.

Примечание. Если степень вершин больше 1, то a_1, a_2, b_1, b_2 вычисляются как расстояние от дерева до соответствующих стволов. Например, ветвь, инцидентная 1-й вершине, расположена слева и доходит до границы ствола, тогда $a_1 = 0$; если же ветвь не доходит до границы ствола, то a_1 будет равно расстоянию от левого конца ветви до границы ближайшего левого ствола. Если у каждой вершины по две инцидентные горизонтальные ветви и каждая из них доходит до границ вертикальных стволов, то полагаем: $\mu_L = \mu_P = 1/2$.

2. Ветвь соединяет основную вершину с дополнительной.

Горизонтальная ветвь. Если горизонтальной ветви в дополнительной вершине ортогональна только одна ветвь и степень вершины равна двум, кроме того, вертикальная ветвь либо не пересекает горизонтальных границ, либо целиком лежит в вертикальном стволе, то в этом случае, полагаем $\mu = 1$ к ближайшему горизонтальному стволу, через который проходит вертикальная ветвь.

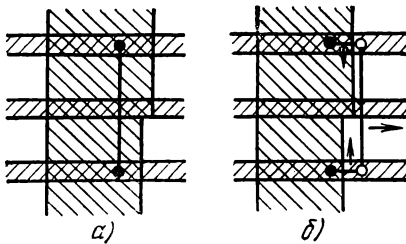


Рис. 6.1

Остановимся на случае, когда дополнительная вершина имеет степень 3 и лежит на горизонтальной границе в вертикальном граничном пространстве, и ей инцидентны две вертикальные ветви, не пересекающие границ. Если в этом случае горизонтальная ветвь лежит в пределах вертикальной границы, то она «раздваивается» на верхнюю и нижнюю, и каждой ветви приписывается $\mu = 1$ к верхнему и нижнему стволу соответственно (на рис. 6.2, где обозначена ветвь до (а) и после (б) разнесения и штриховой линией показано раздвоение ветви). Если горизонтальная ветвь выходит за пределы вертикальной границы, то она

раздваивается таким же образом перед заходом в последнюю вертикальную границу. В месте раздвоения ставится дополнительная вершина и вводится отрезок нулевой длины (на рис. 6.3 обозначено дерево до (а) и после (б) разнесения горизонтальной ветви; горизонтальная ветвь раздвоилась перед заходом в вертикальную границу). Если дополнительная вершина лежит в вертикальном стволе, то принадлежность ветви определяется по формулам (6.1) с учетом инерционности по формулам (6.2).

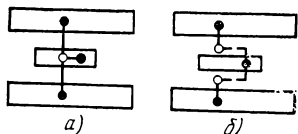


Рис. 6.2

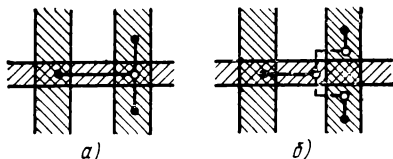


Рис. 6.3

Если степень дополнительной вершины равна 3 и ей инцидентна только одна вертикальная ветвь, то рассматриваемая ветвь получает меру принадлежности $\mu=1$ к ближайшему стволу, через который проходит эта вертикальная ветвь. Подобным же образом делается разнесение при степени дополнительной вершины, равной 4.

Вертикальная ветвь. Вертикальные ветви, полностью лежащие в стволе, получают $\mu=\mu_c$ с указанием номера ствола, в котором лежат. Если вертикальная ветвь лежит в вертикальном граничном пространстве и не пересекает горизонтальных границ, то полагаем для этой ветви $\mu=\mu_r$.

Рассмотрим случай, когда вертикальная ветвь лежит в вертикальном граничном пространстве и пересекает горизонтальные границы. Если, кроме того, дополнительной вершине инцидентна еще и одна горизонтальная ветвь, доходящая до границы, то в числителях формул (6.3) приравниваются нулю соответствующие

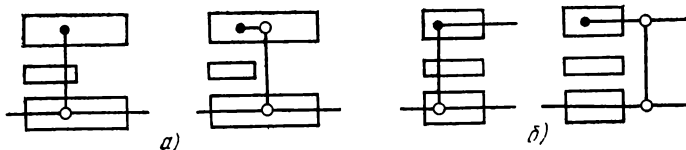


Рис. 6.4

щие слагаемые, относящиеся к дополнительной вершине. Примечание на стр. 116 относится в этом случае только к основной вершине. После определения приоритетной принадлежности дополнительная вершина переносится на границу вертикального ствола; делается также соответствующая корректировка горизонтальных ветвей рассматриваемой части дерева.

Если дополнительная вершина имеет две инцидентные горизонтальные ветви, выходящие за пределы вертикальных границ, то формулы (6.3) упрощаются:

$$\mu_n = b/(a+b); \quad \mu_n = a/(a+b); \quad (6.5)$$

где a , b — расстояния от основной вершины соответственно до ближайшего левого и ближайшего правого ствола.

Если при этих условиях основная вершина имеет степень больше 1 и инцидентные ей ветви горизонтальны, то расстояние вычисляется до ближайшего ствола от ветви. Если ветвь доходит до ствола, соответствующее слагаемое приравнивается нулю. После определения приоритетной принадлежности достраиваются ветви, вводятся дополнительные вершины, ненужные дополнительные вершины снимаются. Примеры приведены на рис. 6.4, где *а*) — пример обхода препятствий при разнесении вертикальной ветви, *б*) — пример сдвига вертикальной ветви по горизонтальным ветвям. В обоих случаях вводятся дополнительные вершины вверху, а внизу происходит перенос дополнительной вершины.

Если ветвь переходит из ствола в ствол, то вопрос решается, как и в случае 1.

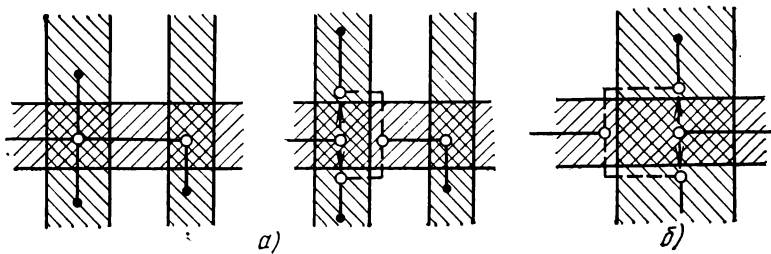


Рис. 6.5

3. Ветвь соединяет две дополнительные вершины.

Горизонтальная ветвь. Если ветвь лежит в горизонтальном стволе, то для нее полагаем $\mu = \mu_c$. Если ветвь не выходит за пределы одного вертикального ствола, то также полагаем $\mu = \mu_c$.

Если одна вершина лежит на пересечении горизонтальной и вертикальной границ и ветвь выходит за пределы одной из границ, то рассматриваются следующие случаи. Указанная вершина имеет одну инцидентную вертикальную ветвь, которая не пересекает горизонтальных границ; тогда полагаем $\mu = 1$ к стволу, в котором лежит вертикальная ветвь. Если при тех же условиях есть еще и горизонтальная ветвь, то она либо уже разнесена к этому же стволу, либо еще не разнесена. Рассматриваемая горизонтальная ветвь в этом случае распределяется по «частям». Сначала от одной вершины до вертикального ствола, затем оставшаяся часть. Если ветви разносятся в разные стволы, то в точке деления вводится дополнительная вершина и отрезок нулевой длины. Если вершине, лежащей на пересечении горизонтальной и вертикальной границ, инцидентны две вертикальные ветви, не пересекающие горизонтальных границ, то часть ветви от вертикального ствола до вершины раздваивается (рис. 6.5,а). При этом подразумевается, что если вторая горизонтальная ветвь и есть, то она еще не разнесена; в противном случае необходимости в раздвоении нет (рис. 6.5,б).

Вертикальная ветвь. Если вертикальная ветвь не пересекает горизонтальных границ и лежит в вертикальном граничном пространстве, то полагаем $\mu = \mu_r$. Если ветвь лежит в вертикальном граничном пространстве и пересекает горизонтальные границы, то мера принадлежности определяется по формулам (6.3) с учетом примечания на стр. 116. Если вершины имеют инцидент-

ные разнесенные ветви (вертикальные), то мера принадлежности вычисляется по формулам, аналогичным (6.2).

Общий порядок разнесения ветвей таков. В первую очередь распределяются все горизонтальные ветви, соединяющие две основные вершины; затем распределяются аналогичные вертикальные ветви. На следующем этапе сначала разносятся горизонтальные ветви, соединяющие основную и дополнительную вершины, затем аналогичные вертикальные ветви. На последнем этапе разносятся ветви, соединяющие дополнительные вершины: сначала горизонтальные, а затем — вертикальные.

6.3. ОКОНЧАТЕЛЬНОЕ РАСПРЕДЕЛЕНИЕ ВЕТВЕЙ ДЕРЕВЬЕВ НА ПЛАТЕ

При предварительном распределении ветвей деревьев по стволам платы и при определении меры принадлежности ветви к одному из стволов во внимание принимались, в основном, факторы, учитывающие «лучшее» расположение рассматриваемой ветви в данный момент. Если некая ветвь γ получает приоритетную принадлежность к i -му стволу, то для этой ветви получалось $\mu_i(\gamma) \geq 1/2$. В альтернативный ствол, номер которого будет $i-1$ либо $i+1$, ветвь получает меру принадлежности $\mu_{i\pm 1}(\gamma) = 1 - \mu_i(\gamma)$. При $\mu_i(\gamma) = 1/2$ считается, что горизонтальная ветвь относится к нижнему стволу, а вертикальная — к правому, т. е. к стволу с большим номером.

Для контроля за укладкой строятся матрицы пропускных способностей сечений стволов рабочего поля платы и матрицы загрузки. Матрицы пропускных способностей строятся на основании модели платы, описанной в § 3.2 (рис. 3.11). Размерность матриц определяется числом стволов в соответствующем направлении и размерами минимального прямоугольника, описывающего поле трассировки.

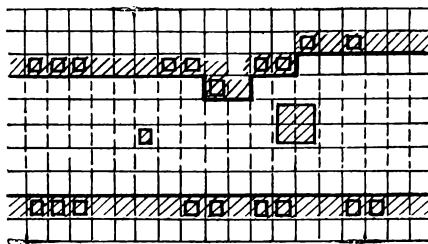


Рис. 6.6

Если размерность матрицы посадочных мест будет $m \times n$, то при принятой ориентации получаем $2n+1$ горизонтальных и $m+1$ вертикальных стволов. Предположив размерность в дискретах минимального описывающего прямоугольника $k \times l$, получаем число сечений любого горизонтального ствола не более чем k , а вертикального — l . Таким образом, размерность матрицы пропускных способностей горизонтальных стволов W будет $(2n+1, k)$:

$$W = \begin{pmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1k} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2k} \\ \vdots & \vdots & & \vdots \\ \omega_{2n+1,1} & \omega_{2n+1,2} & \dots & \omega_{2n+1,k} \end{pmatrix}.$$

Если

$$z_{ij}'' \geq 0, \omega_{lk}'' \geq 0 \quad (6.6)$$

для любых значений i, j, l, k , то это означает, что число каналов при данном разнесении не превышает возможностей проведения ни в одном сечении.

При выполнении этих условий можно перейти к дальнейшей оптимизации равномерной загрузки стволов. Одним из необходимых условий перераспределения трасс между стволами, кроме выполнения условия (6.6), является дополнительно выполнение следующих неравенств:

$$\sum_{l=1}^{2n+1} \omega_{lk}'' > 0 \text{ для } \forall k, \sum_{j=1}^{m+1} z_{ij}'' > 0 \text{ для } \forall i, \quad (6.7)$$

где $2n+1$ — число горизонтальных, а $m+1$ — вертикальных стволов.

Анализ ряда плат показал, что оптимизацию равномерности распределения целесообразно проводить при выполнении следующих усилений требований (6.6) и (6.7):

$$\sum_{j=1}^{m+1} z_{ij}'' \geq m+1 \text{ для } \forall i, \sum_{l=1}^{2n+1} \omega_{lk}'' \geq 2n+1 \text{ для } \forall k, \quad (6.8)$$

т. е. когда в среднем в каждом сечении платы число свободных каналов не меньше числа стволов.

Оптимизация заключается в выделении элементов матриц Z'' и W'' , равных нулю: $z_{ij}''=0$ и $\omega_{lk}''=0$; после этого понижаем пропускную способность соответствующих сечений путем вычитания единицы из соответствующих элементов матриц Z и W : z_{ij} , ω_{lk} , и переходим к перераспределению ветвей в стволах по описанному ниже алгоритму, учитывающему перегрузку стволов. Перераспределение ветвей деревьев по стволам, т. е. окончательная укладка, заключается в следующем.

Рассмотрим горизонтальный перегруженный ствол i . Начиная с первого j , для которого $\omega_{ij}'' < 0$, находим минимальный интервал перегрузки $[j_1, j_2]$, на котором выполнено условие $\omega_{ij}'' < 0$ для $j \in [j_1, j_2]$ и $\omega_{ij}'' > 0$ для $j=j_1-1$ и $j=j_2+1$. Из множества фрагментов, приоритетно принадлежащих i -му стволу, выбираем такие, которые пересекают полностью интервал $[j_1, j_2]$:

$$\gamma(i; [j_1, j_2]) = \{\gamma : \mu_i(\gamma) \geq 1/2, x_1(\gamma) \leq j_1, x_2(\gamma) > j_2\}.$$

Из полученного множества фрагментов $\gamma(i; [j_1, j_2])$ выбираем отрезок $[x_1', x_2']$ с минимальной мерой принадлежности к i -му стволу. В альтернативном стволе, допустим $(i-1)$ -м, определяется возможность проведения этого отрезка, для чего должно выполняться условие

$$\omega_{i-1, j'}' > 0 \text{ для любого } j' \in [x_1', x_2']. \quad (6.9)$$

Если условие (6.9) не выполняется, то из множества отрезков $\gamma(i; [j_1, j_2])$ выбирается следующий и т. д. При невозможности переноса ветвей в стволы $i-1$ и $i+1$ делается попытка разгрузить эти стволы аналогичным переносом ветвей в стволы $i-2$ и $i+2$ и т. д.

Просматриваются также варианты, когда не существует отрезков, полностью пересекающих интервал перегрузки $[j_1, j_2]$. В этом случае выбираются пары перекрывающихся отрезков, каждый с мерой $\mu < 1$: $[x_1^1, x_2^1]$ и $[x_1^2, x_2^2]$ и таких, что $x_1^2 \in [x_1^1, x_2^1]$, $x_2^1 \in [x_1^2, x_2^2]$, $x_1^1 < j_1 < x_1^2$, $x_2^1 < j_2 < x_2^2$.

Возможность переноса определяется по формулам:

$$\begin{aligned} \omega''_{i\pm 1, j} > 0, \text{ для любого } j \in [x_1^1, x_2^2], \\ \omega''_{i\pm 1, j} > 1 \text{ для любого } j \in [x_1^2, x_2^1]. \end{aligned} \quad (6.10)$$

Выполнение условий (6.9) и (6.10) при переносе является необходимым, но не достаточным, так как при наличии ортогонально подходящих смежных ветвей и необходимости их удлинения нужно просматривать на допустимость переноса также и загрузку соответствующих сечений ортогонального направления.

Если по указанным критериям перенести ветвь не удастся, то исключается из рассмотрения минимальный путь в дереве, содержащий эту ветвь и имеющий своим началом и концом либо основные вершины, либо дополнительные со степенью более 2, либо любое сочетание этих вершин. Попытка реализации таких путей делается после трассировки в стволах.

Анализ спроектированных плат показал, что по результатам разнесения и корректировки постоянно недогруженными получаются первый и последний стволы, т. е. стволы, одной из границ которых является контур платы. Наиболее просто переносятся отрезки в горизонтальных стволах. Практически все переносимые отрезки полностью пересекали интервал перегрузки. Затруднения в переносе вертикальных отрезков вызывались трудностями предоставления горизонтальных каналов для перехода ветви в другой ствол.

6.4. ОСОБЕННОСТИ РЕАЛИЗАЦИИ ПОФРАГМЕНТНОГО ПОДХОДА, СОГЛАСОВАННОГО С ЭТАПАМИ КОМПОНОВКИ И РАЗМЕЩЕНИЯ

Предварительное распределение ветвей деревьев по стволам с последующей корректировкой с соблюдением пропускных способностей сечений стволов позволяет с большой вероятностью гарантировать распределенным ветвям место на плате. В связи с тем, что реализация разнесенной ветви должна проходить в строго отведенном ей стволе, при проведении необходимо ограничивать ствол при распространении волны. С этой целью предварительно строится ограничивающий прямоугольник.

Так как всем распределенным ветвям есть место на плате, то теперь на первый план перемещается вопрос о минимизации числа пересечений, так как последние являются основной причиной, порождающей переходы со слоя на слой. Число пересечений проводников зависит от порядка проведения трасс. На вопрос о порядке проведения трасс с целью минимизации числа пересечений дает ответ теорема Жордана, утверждающая, что всякая простая замкнутая кривая C разделяет точки плоскости, не принадлежащие C , на две такие области (не имеющие общих точек), по отношению к которым сама кривая C является общей границей [80]. Оставим понятие простой замкнутой кривой и самой теоремы Жордана интуитивными понятиями. Простота теоремы кажущаяся и ее строгое доказательство занимает несколько страниц.

Из этой теоремы следует, что при пофрагментной реализации необходимо исходить из длины реализуемого фрагмента. В работе [100] показано, что построение всей связывающей цепи не всегда является оптимальным решением, поэтому нами и разработан пофрагментный подход. (Ввиду того, что при реализации ветви в виде печатного проводника на плате возможны повороты проводника, что плохо согласуется с исходным представлением дерева, в дальнейшем будем использовать термин «фрагмент», подразумевая под этим и ветвь дерева, и соответствующий этой ветви печатный проводник, который либо прокладывается, либо уже проложен.)

Покажем на примере роль длины фрагмента. Пусть на плате дана горизонтальная прямая и на ней четыре точки: k_1, k_2, k_3, k_4 . Допустим, что соединять эти точки можно только над прямой. В этом случае, если $[k_1, k_2] \cap [k_3, k_4] = \emptyset$ или $[k_1, k_2] \supset [k_3, k_4]$ либо $[k_1, k_2] \subset [k_3, k_4]$, то точки k_1, k_2 и k_3, k_4 можно соединять между собой попарно проводниками без пересечений. В первом случае соединяемые точки лежат на разных частях прямой, поэтому соединение обеих пар точек может быть сделано по ближайшему к прямой каналу. Во втором и третьем случаях для избежания пересечений необходимо провести на ближайшем канале проводник, соединяющий меньший отрезок, а на следующем канале — большой отрезок (рис. 6.7). При обратном порядке проведения получается два пересечения (рис. 6.8). В противном случае, когда $[k_1, k_2] \not\supset [k_3, k_4]$ и $[k_1, k_2] \not\subset [k_3, k_4]$, и $[k_1, k_2] \cap [k_3, k_4] \neq \emptyset$, то при условии проведения трасс над прямой пересечений не избежать (рис. 6.9). Этим примером подтверждается необходимость проведения сначала коротких, а затем более длинных фрагментов и т. д.

Неизбежность пересечения может возникнуть в ситуациях, подобных следующей. Пусть основная вершина k_1 лежит на нижней границе горизонтального ствола (ветвь горизонтальная), а дополнительной вершине k_2 со степенью 2 инцидентна ветвь, уходящая от нее вверх. В этом случае, если $k_1 \in [k_3, k_4]$, при упомянутых выше условиях пересечение при реализации связывающей сети, содержащей вершину k_1 , обязательно будет, в то время как при условии $k_1 \in [k_3, k_4]$, $k_2 \in [k_3, k_4]$ избежать пересечения можно проведением на ближайшем к границе канале

фрагмента $[k_3, k_4]$, а ветвь $[k_1, k_2]$ реализовывать на одном из последующих каналов.

Учитывая приведенные выше соображения, сформулируем правило определения очередности проведения фрагментов. Множество фрагментов, подлежащих реализации в настоящий момент в данном стволе (допустим горизонтальном), делится на три подмножества: фрагменты, соединяющие вершины нижней границы ствола, у которых степень каждой вершины равна 1, либо имеющие прочие инцидентные ветви, направленные вниз; фрагменты, соединяющие вершины верхней границы ствола со степенью вершины, равной 1, либо имеющие прочие инцидентные ветви, направленные вверх; все остальные фрагменты.

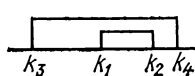


Рис. 6.7

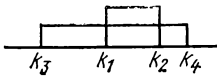


Рис. 6.8

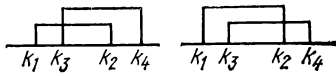


Рис. 6.9

После разбиения множества фрагментов на три подмножества упорядочиваем фрагменты первого и второго подмножества по возрастанию длин фрагментов, а затем упорядочиваем по этому же принципу фрагменты третьего подмножества.

В каждом стволе в первую очередь реализуются в указанном порядке фрагменты с одной или двумя основными вершинами. После проведения фрагмента с одной основной вершиной дополнительная вершина переводится в разряд псевдоосновных, которые при определении порядка проведения имеют статус основной вершины.

После проведения фрагментов во всех горизонтальных стволах в таком же порядке проводятся фрагменты в вертикальных стволах. После окончания первого этапа образуются новые фрагменты с образовавшимися псевдовершинами, поэтому процесс итерационно продолжается до реализации всех разнесенных по стволам фрагментов.

Коренным отличием от общепринятых подходов является реализация на отдельных этапах ветвей, инцидентных точкам Штейнера, когда проводники различных цепей на некоторое время «зависают». Дальнейшим развитием такого подхода следует считать включение в рассмотрение случаев перехода фрагментов в соседние стволы, как, например, показано на рис. 6.10. Данную ситуацию можно обрисовать следующим образом. Если отрезки $[k_1, k_2]$ и $[k_3, k_4]$ пересекаются: $[k_1, k_2] \cap [k_3, k_4] \neq 0$, а отрезок $[k_3, k_4]$ пересекает хотя бы один вертикальный ствол, то возникает возможность реализации этих двух отрезков без пересечения. Читателю может показаться, что мы упускаем из рассмотрения возможность разнесения этих отрезков в разные стволы, но ранее предполагалось, что фрагменты «закреплены» в стволах, а при высокой плотности трассировки вольность в свободном

перенесении может резко ухудшить конечные результаты. Дальнейшим развитием описанного выше подхода является учет при распределении и оптимизации ветвей деревьев по стволам и критерия пересечения фрагментов (или критерия вложенности).

Рассмотренный выше подход к порядку реализации ветвей деревьев на плате хорошо согласуется с критериями, выдвинутыми и учитываемыми на этапах компоновки и размещения. Действительно, на этапе компоновки ставилась задача получить максимальное число петель в гиперграфе, и при этом условии, используя эквивалентность базовых элементов и их входов, достигнуть минимума загрузки каналов и длины проводников в горизонтальном направлении, добиваясь при этом максимального использования вертикального граничного пространства. Так как расстояние между выводами одного модуля, входящими в одну цепь, в большинстве случаев меньше, чем расстояние между выводами различных модулей, также входящих в одну цепь, то соединения между выводами одного модуля будут реализовываться в первую очередь, что соответствует более рациональному использованию рабочего поля трассировки.

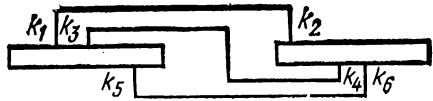


Рис. 6.10

Аналогично при размещении элементов на плате минимизировалась функция (5.1), учитывающая длину подсоединений очередной группы контактов размещаемого модуля, что согласуется с методикой построения квазиоптимального дерева на ортогональной сетке и с порядком разнесения и выбора ветвей для реализации. Отметим, что при определении меры принадлежности и разнесения ветвей проводилась достройка квазиоптимального дерева, т. е. его удлинение, что и учитывалось при размещении элементов.

6.5. ВОПРОСЫ РАССЛОЕНИЯ СОВМЕЩЕННОГО РИСУНКА

Как уже отмечалось ранее, при автоматизированном проектировании двухсторонних печатных плат остро стоит задача минимизации числа переходных отверстий, а также задача определения их оптимального местонахождения и конфигурации. Упоминаемые выше критерии существенно влияют на технологичность изделия, его стоимость, а также на эксплуатационные качества (надежность, ремонтпригодность).

Требования увеличения плотности печатного монтажа, с одной стороны, и минимизации числа переходных отверстий с одновременным увеличением размеров контактных площадок для этих переходных отверстий, с другой стороны, начиная с некоторого момента для любого набора алгоритмов становятся несовместимыми (что наблюдается и при неавтоматизированном про-

ектировании). Учитывая большую стоимость ручной доработки (в том числе и временную), в большинстве систем идут на увеличение числа переходных отверстий с уменьшением размеров контактных площадок, несмотря на усложнение в этом случае технологии и повышение стоимости проектируемого изделия.

Простейшие алгоритмы расслоения основаны на разнесении по слоям с учетом только ориентации отрезков, т. е. горизонтальные отрезки относятся на один слой, а вертикальные — на другой. Такие алгоритмы дают необоснованно большое количество переходов. Одна из модификаций этого алгоритма, позволяющая сократить число переходов, заключается в рассмотрении путей, а не отрезков. В случае конфликта отрезков путей (рис. 6.11,а)

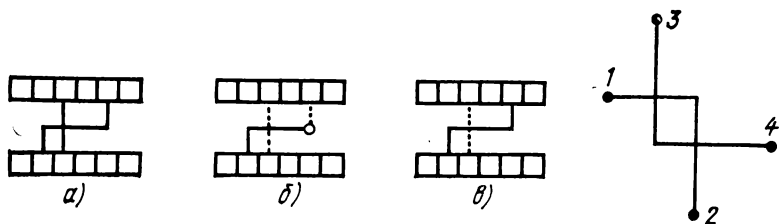


Рис. 6.11

Рис. 6.12

они разносятся на разные слои (рис. 6.11,б). Недостаток этого алгоритма заключается в разнесении на разные слои отрезков путей, конфликтующих всего один раз, в то время как полностью разносить их на разные слои можно без переходов (рис. 6.11,в). Подобные алгоритмы ранее использовались в различных САПР.

Представляет интерес алгоритм расслоения, описанный в работе [101]. При решении задачи расслоения в указанной работе предлагается иметь дело с прямолинейными отрезками. Рассматривается множество всех отрезков, полученных в результате трассировки на совмещенном рисунке. Это множество разбивается на непересекающиеся подмножества пересекающихся отрезков, т. е. выделяются фактор-множества пересекающихся отрезков. Каждый непересекающийся отрезок образует по определению фактор-множество. Окончательное решение принадлежности отрезков к какому-нибудь слою сводится к оптимальному расположению пар подмножеств, что преследует цель минимизировать число переходов со слоя на слой на границах фактор-множеств.

На рис. 6.12 показан простейший пример работы алгоритма и его положительные свойства. Отрезки 1 и 3 образуют одно из фактор-множеств, а отрезки 2 и 4 — другое. Очевидно, что если расположить отрезок 1 на одном слое, то отрезок 3 лучше расположить на другом. Аналогично со вторым множеством отрезков — 2 и 4. Окончательный результат: отрезки 1 и 2 располагаются на одном слое, а отрезки 3 и 4 — на другом. Переходов со слоя на слой в этом случае не будет. В работе предлагается оце-

ночная функция для определения качества взаимного расположения фактор-множеств при просмотре нескольких вариантов, однако методов поиска оптимального решения для большого числа отрезков не предлагается. В работе описывается локально-оптимальный метод поиска решения на основе формирования допустимого множества решений. В развитом в этой работе методе проектирования двухслойных печатных плат вопрос минимизации числа переходных отверстий решается в два этапа.

Первый этап — проектирование совмещенного рисунка трассировки с учетом критерия минимизации числа пересечений трасс; второй — расслоение полученного рисунка с требованием минимизации числа переходов со слоя на слой. Первый этап уже нашел достаточное отражение в изложении работы, поэтому на нем мы останавливаться не будем.

Рассмотрим второй этап. После получения совмещенного рисунка имеется информация о цепях в виде отрезков с координатами их концов на плате, а также с указанием качества вершин-концов: основная или дополнительная. Основная вершина соответствует контактной площадке элемента, установленного на плате, а дополнительная — точкам перегиба проводников или точкам ветвления, если в этой точке нет основной вершины. На совмещенном рисунке отрезки помечены признаками ориентации, а в точках перегиба и пересечения отрезков в дискрете платы указана принадлежность к обоим слоям.

Такой рисунок позволяет практически сразу получить разделенные слои, назначив в точках перегиба с совмещенными признаками переходы со слоя на слой. Однако при этом получается избыток переходных отверстий. Предлагается дальнейшее развитие изложенных выше идей. Прежде всего следует заметить тот факт, что нет необходимости объявлять отдельными фактор-множествами непересекающиеся отрезки. Действительно, если отрезок не пересекает ни одной другой отрезок и соединяет только фиксированные на плате вершины, т. е. контакты, то такой отрезок без потери качества может быть отнесен к любому слою; если отрезок не пересекает ни один другой отрезок, а соединяет либо два конца других отрезков, либо контактную площадку и конец какого-нибудь отрезка, то он может быть присоединен к концу любого отрезка, имеющего с ним общую вершину, и может быть отнесен вместе с последним к одному фактор-множеству и одному слою, что не увеличивает числа переходов, а наоборот, дает дополнительную информацию о числе общих граничных точек непересекающихся множеств. При этом появляется неоднозначность определения отношения такого отрезка к тому или иному слою или к множеству пересекающихся отрезков. Но здесь следует заметить, что, придавая присоединенному отрезку признак принадлежности к ориентации того отрезка, к которому он присоединяется, в любом случае информация о числе точек соприкосновения множеств остается одной и той же, т. е. сохраняются точные данные, необходимые для оптимального расслоения.

Таким образом, размерность задачи, первоначально поставленной в работе [101], понижается без потери информации, что наблюдалось в той же статье на первом этапе перехода от оптимального к локально-оптимальному алгоритму.

Размерность можно понизить также следующим образом. Разбив множество отрезков проводников рисунка трассировки на фактор-множества пересекающихся отрезков, построим граф. Каждое фактор-множество будем считать вершиной $x \in X$ графа $G(X, U)$. Ребро $u \in U$ этого графа отражает наличие общих граничных точек между смежными вершинами. Каждому ребру приписывается вес, соответствующий числу общих граничных точек между этими смежными вершинами.

В полученном графе предлагается рассматривать отдельно каждую компоненту связности. В этом случае число просматриваемых вариантов сокращается до суммы вариантов в каждой компоненте вместо произведения. Расслоение на каждой компоненте не влияет на качество решения. Действительно, так как две компоненты не связаны, то между ними нет пути, т. е. нет общих граничных точек между соответствующими множествами пересекающихся отрезков. Отсюда следует, что при любом варианте взаимного расположения этих компонент они между собой порождать переходы со слоя на слой не будут, что и требовалось доказать.

Решение задачи оптимального взаимного расположения фактор-множеств на каждой компоненте связности можно рассматривать как раскраску графа в два цвета, пользуясь при этом хорошо известным утверждением, что простые четные циклы позволяют раскрашивать граф в два цвета, а простые нечетные — нет. Пути без циклов позволяют окрашивать граф в произвольное число красок. Таким образом, задача сводится к выявлению нечетных циклов и удалению из такого цикла ребер. При этом следует иметь в виду, что вес удаляемого ребра будет соответствовать числу переходов со слоя на слой в граничных точках, соответствующих этому ребру. Отсюда следует, что необходимо удалять ребра, имеющие минимальный вес.

Прежде чем переходить к примеру, поясняющему порядок снятия ребер, оговоримся, что данный подход минимизирует число переходов в углах поворотов трасс, т. е. задача решается с точностью до числа тройников в точках соприкосновения непесекающихся множеств. Из этого не следует, что алгоритм не работает с вершинами связывающей сети, имеющими степень более 2. При необходимости определяются точки перехода и в этих вершинах. Общий ход решения задачи показан на структурной схеме рис. 6.13.

Для выбора ветвей, которые должны быть удалены, предлагается строить дерево по следующим правилам. Сначала выбирается вершина, имеющая наибольшую сумму весов связей: она считается правершиной строящегося дерева. От этой вершины строятся ветви связей к другим вершинам — это второй уровень,

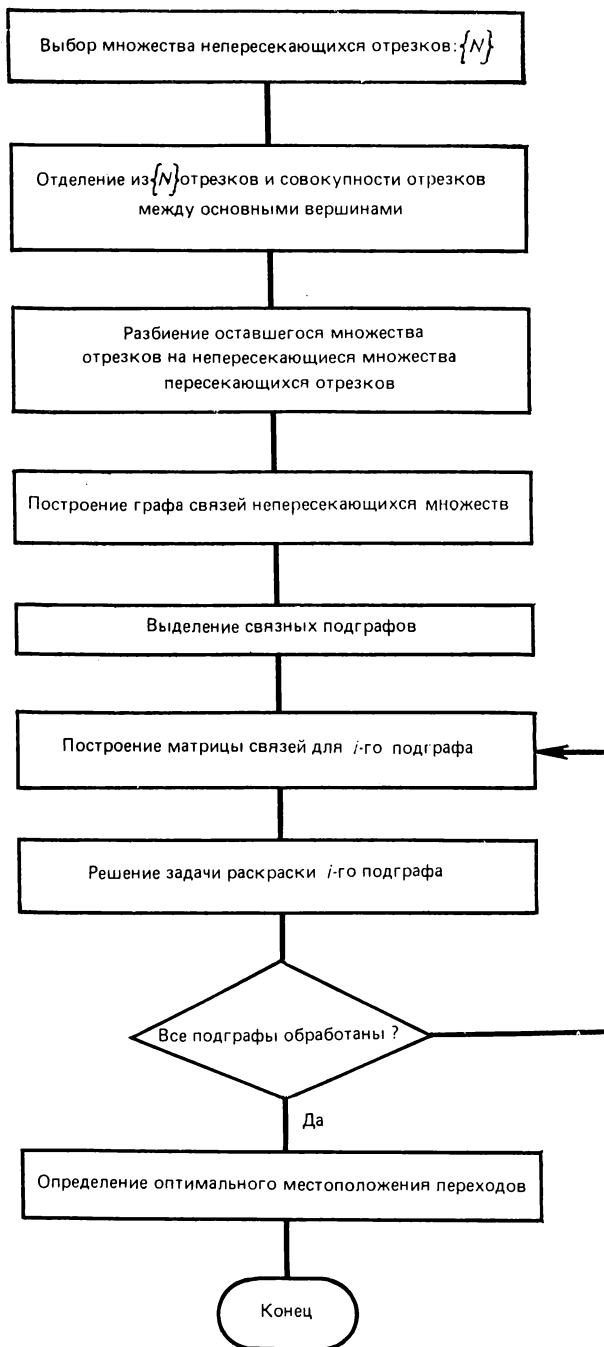


Рис. 6.13

претендующий на окраску вторым цветом. Далее строится следующий уровень по правилу смежности, но без повторения вершин предыдущего уровня (подчеркнем, что предыдущего, а не настоящего и не строящегося; подчеркнем также, что замечание относится к смежным вершинам данной рассматриваемой вершины, в противном случае вершины различных уровней могут и будут повторяться). Если у какой-то вершины на некотором этапе не оказывается смежных вершин, то получаем путь без циклов, т. е. такую ветвь можно раскрашивать. После построения очередного уровня проводится анализ совпадения номеров вершин на четных и нечетных уровнях (достаточно проверять последний построенный уровень с уровнями противоположной четности). При совпадении номеров вершин одна из ветвей, на которых совпали номера, должна быть разрушена. Если вершины принад-

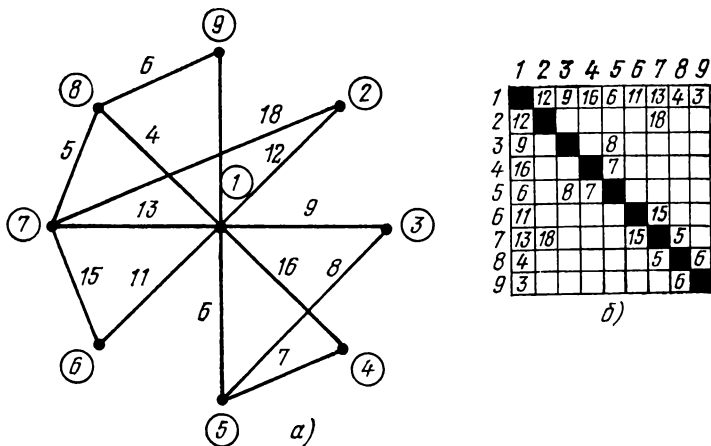


Рис. 6.14

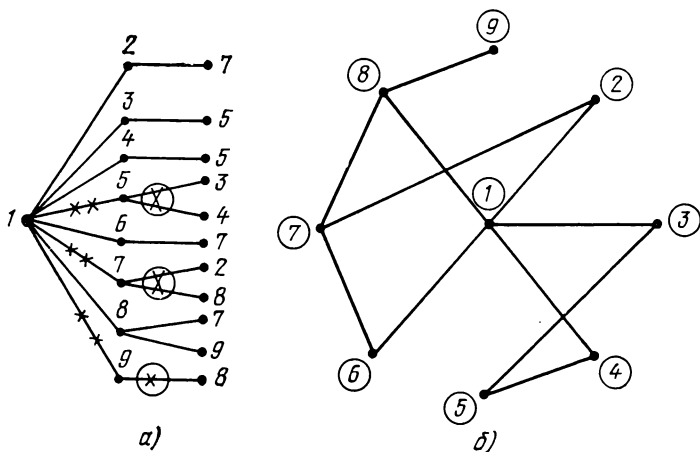


Рис. 6.15

лежат одному пути от правершины и не имеют ветвлений, то можно удалить любую ветвь из дерева, а следовательно, и из графа, выбирая при этом ребро с меньшим весом. Если вершины лежат на разных путях, то лучше удалить ребро с меньшим весом, перебирая при этом несколько путей. Если повторение неоднократное, то возможно, что удаление этого ребра приведет к снятию не одной, а нескольких вершин.

На рис. 6.14,а приведен пример графа G фактор-множеств, а на рис. 6.14,б — соответствующая матрица весов. На рис. 6.15,а показано дерево связей этого графа (удаляемые ребра перечеркнуты). На рис. 6.15,б показан конечный граф, подлежащий раскраске. Процесс заканчивается после заполнения списка раскрашенных без противоречия вершин.

После определения взаимного расположения множеств каждому отрезку приписывается его слоинность. Окончательное определение места расположения каждого конкретного перехода определяется просмотром окрестности точки поворота с попыткой, в случае необходимости, сместить переход по одному из проводников. Данный этап в смысле алгоритмизации существенных трудностей не представляет.

Реализация описанного подхода позволяет сократить (по сравнению с результатами, полученными с помощью алгоритмов разнесения единой конфликтующих отрезков) число переходов на платах с низкой плотностью печатного монтажа более чем в 2 раза, на платах со средней плотностью — в 1,5—2 раза, на платах с высокой плотностью — приблизительно в 1,5 раза. Такой результат объясняется тем, что с повышением плотности монтажа уменьшается число образуемых факторов-множеств.

После расслоения целесообразно проводить автоматическую дотрассировку, распространяя волну по разным слоям платы, а если волна переходит в разрешенных местах на другой слой, то в этих точках образуются новые источники волн на новом слое. Таким образом может быть реализована непроведенная или нераспределенная часть фрагментов.

Глава 7

НЕКОТОРЫЕ ВОПРОСЫ ТЕХНИКО-ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ САПР

7.1. ВОПРОСЫ ВНЕДРЕНИЯ САПР НА ПРЕДПРИЯТИИ

В настоящее время вряд ли найдется предприятие, занимающееся проектированием РЭА, перед которым не стоял или не стоит вопрос выбора или разработки САПР. Причина этого — перегрузка конструкторских отделов, занимаю-

щихся неавтоматизированным проектированием, резкое увеличение сроков разработки конструкторской документации при все возрастающей интеграции и миниатюризации аппаратуры, высокая стоимость переделок при устранении ошибок проектирования, число которых при неавтоматизированном проектировании резко возрастает с увеличением объема обрабатываемой информации, возрастающие требования к точности изготовления. В то же время использование САПР при разработке и изготовлении только малых серий относительно несложных объектов нецелесообразно.

Можно отметить следующие преимущества проектирования с помощью ЭВМ: достоверность, которая обеспечивается строгим соответствием выходной информации входной (благодаря всевозможному контролю на различных этапах), а также логическим контролем, позволяющим выявить ошибки разработчика, трудно обнаруживаемые при неавтоматизированном проектировании;

сокращение сроков проектирования вследствие быстродействия ЭВМ, а также снижение трудоемкости, т. е. повышение производительности труда;

сокращение объема рутинных работ, т. е. социальный фактор;

сокращение сроков производства аппаратуры в результате выпуска с помощью ЭВМ управляющих программ на различных носителях для станков с числовым программным управлением, технологических и различных тестовых автоматов;

возможность включения САПР в общую структуру гибкого автоматизированного производства;

более безболезненная настройка САПР при изменении конструкций благодаря современным методам создания программного обеспечения по сравнению с соответствующей перестройкой у коллектива конструкторов, занимающихся неавтоматизированным проектированием.

Обсудим вопрос выбора метода проектирования с помощью ЭВМ.

Принято различать автоматизированное и автоматическое проектирование с помощью ЭВМ. При этом можно из автоматизированного метода выделить случай, когда автоматизируется небольшое число этапов. Эксплуатация систем с малым числом автоматизируемых этапов ведется, как правило, на малых вычислительных машинах, так как вопросы раскладки схем остаются в таких системах неавтоматизированными (первоначально такие системы базировались только на координатографах и координатоскопах, что резко сужало их возможности). Часто такое проектирование используют на предприятиях как начальную стадию автоматизации проектирования РЭА в качестве переходного этапа, а также при большом тиражировании с одного типоразмера, и технология производства в этом случае, как правило, незначительно отличается от технологии производства аппаратуры, спроектированной неавтоматизированным способом.

Развитие систем с малым числом автоматизируемых этапов на предприятии целесообразно при проектировании около 200—300 типоразмеров плат. Если такие системы выигрывают с технологической точки зрения, то с точки зрения контроля информации они проигрывают. Автоматизированные (впрочем так же, как и автоматические) системы проектирования, охватывающие большее число этапов, требуют значительно больших аппаратных средств, и, что более существенно, нового подхода к технологии изготовления. Эти изменения более остро ставят вопрос изучения возможности и перспективы производства, а также возможности, требования и перспективы развития основных заводов-изготовителей перед принятием решения о выборе САПР.

Автоматическое проектирование, когда человек не вмешивается в процессе конструирования в действия ЭВМ, чаще применяется при проектировании многослойных печатных плат, когда число слоев практически не ограничено, или при проектировании двухслойных печатных плат, если разрешается неразведенные связи реализовывать в виде навесных перемычек. Это направление постепенно изживает себя.

Подводя итог, можно сказать, что наибольшее распространение и развитие получают автоматизированные системы проектирования, в которых на разных уровнях предусмотрен диалог человека и машины. Такие системы оправдывают себя при проектировании более 300 типоразмеров плат в год.

Для принятия решения о выборе эффективного в конкретных условиях метода проектирования с помощью ЭВМ необходимо рассмотреть вопросы, связанные с работой различных служб самого предприятия, предприятий-смежников и заводов-изготовителей. Прежде всего следует решить — разрабатывать свою САПР или взять готовую и адаптировать ее к своим возможностям и потребностям.

Для создания своей системы необходимо иметь достаточный штат специалистов, способных проектировать САПР. Среди них можно выделить следующие основные группы: архитекторы системы, разрабатывающие структуру САПР, удовлетворяющую предъявляемым требованиям; алгоритмисты, способные формализовать поставленные задачи и предложить приемлемые методы их реализации (эти две группы называют также идеологами системы) и, наконец, вычислители-программисты, задача которых — перевод алгоритмов на язык ЭВМ. Создание программ, адекватных сложным алгоритмам, является одной из труднейших проблем. Зачастую идеологи являются в той или иной мере и реализаторами-программистами, что имеет свои положительные и отрицательные стороны. На некоторых вопросах разработки систем остановимся в следующем параграфе.

К квалификации специалистов, внедряющих готовую САПР, могут предъявляться более низкие требования, заключающиеся, в основном, в хорошем знании основ программирования (при этом не имеются в виду представители конструкторских и технологических отделов). Поэтому при недостаточном опыте создания систем (считается, что специалист может стать способным разрабатывать большие системы, если он активно участвовал в разработке трех систем подобного уровня, в более редких случаях бывает достаточно участия в разработке двух систем) целесообразно использовать готовые САПР, что позволяет разработчикам систем быстрее освоиться с задачами проектирования, получить начальный опыт в области, в которой производится разработка САПР, сконцентрировать усилия на лучшей адаптации системы к условиям предприятия. По этим причинам предприятию, ранее не занимавшемуся проектированием РЭА с помощью ЭВМ, целесообразно начинать свою работу в этом направлении с внедрения одной из имеющихся систем. В этом случае встает вопрос адаптации внедряемой системы.

Адаптация заключается, в основном, в приведении программного обеспечения к возможностям предприятия. При этом учитываются технические возможности производства аппаратуры, принятый на предприятии технологический процесс. При несовместимости программного обеспечения не может идти речь о внедрении, так как это наиболее трудоемкая часть разработки САПР, эквивалентная созданию новой системы с учетом имеющихся недостатков.

Итак, при определении политики предприятия по применению ЭВМ для проектирования РЭА должны учитываться следующие факторы:

1. Опыт предприятия по эксплуатации и разработке подобных систем.
2. Технологические возможности данного предприятия, других предприятий-изготовителей проектируемой аппаратуры, перспектива развития технологических возможностей; при этом надо помнить, что усилия громадного коллектива разработчиков САПР могут быть сведены к нулю, если производство не готово внедрить соответствующую технологию.
3. Возможности материально-технического обеспечения как непосредственно для автоматизированного (автоматического) проектирования, так и для производства проектируемой аппаратуры.
4. Объем проектируемой аппаратуры (как по числу типоразмеров, так и по количеству тиражирования с типоразмера). Имеющееся базовое оборудование для САПР (тип ЭВМ и совместимость с другими машинами, периферийное оборудование).

Если принимается решение о внедрении готовой системы, необходимо оценить претендующую систему (относительно альтернативных) со следующих сторон: возможность ее применения при имеющемся техническом оснащении производства; сроки адаптации (изучение системы программистами, освоение конструкторами, подготовка технологии производства); возможности самой системы и предполагаемая трудоемкость внесения изменений в систему с точки зрения возможности расширения ее функций и включения альтернативных алгоритмов; удобство пользования; технико-экономическая эффективность внедрения.

Группа внедрения должна включать в себя программистов, конструкторов и технолога. Численный состав определяется конкретной внедряемой САПР. Обычно это 2—3 программиста, 2—3 конструктора, один технолог. Данный состав необходим только для внедрения системы, при постоянной эксплуатации численный состав зависит от объема проектируемой аппаратуры и поставленных перед этим коллективом задач.

Вне зависимости, внедряется ли взятая на стороне или разработанная на данном предприятии система, остро стоит вопрос организации процесса проектирования. Можно предложить различные решения данной проблемы (выбор конкретного зависит от структуры, традиций и обычаев предприятия).

1. Группа конструкторов-пользователей включается в подразделение вычислительного центра, что обеспечивает простоту связей при прохождении информации на различных этапах автоматизированного проектирования. С другой стороны, при этом они отрываются от основного конструкторского отдела, где проектируется «ручная» часть аппаратуры. Стыковка, увы, безболезненно не проходит.

2. Группа конструкторов-пользователей подчиняется основному конструкторскому отделу. Недостатки и преимущества здесь меняются местами по сравнению с первым вариантом. Ввиду сложности взаимопонимания между представителями ВЦ и КО и комплексного характера ответственности в этом случае подключение руководства на уровне заместителя предприятия необходимо.

3. Выделение в ВЦ ответственных за прохождение конструкторской документации на каждом этапе с последующим вознаграждением по результатам проектирования. Конструкторская группа при этом остается в своем отделе, а ответственные в ВЦ не освобождаются от работ, проходящих по другим темам через ВЦ, однако их внимание концентрируется на задачах автоматизированного про-

ектирования. Такая своеобразная форма «единого наряда» способствует устранению многих мелких недочетов и недоразумений вместе с тем повышает ответственность каждого, участвующего в процессе проектирования, т. е. практически имеет те же преимущества, что и форма «единого наряда».

7.2. МЕТОДОЛОГИЯ РАЗРАБОТКИ САПР

Наиболее перспективным является разработка САПР, способных охватить автоматизацией как можно больше этапов проектирования аппаратуры, а также имеющих возможность варьировать на каждом этапе альтернативным набором алгоритмов, что означает интеграцию по глубине и ширине соответственно.

Современные САПР должны также обеспечивать гибкость программного обеспечения и простоту его приспособляемости к изменениям проектируемой аппаратуры.

Современный уровень развития автоматизации требует либо создания на предприятии большого коллектива разработчиков-автоматизаторов, что обычно не под силу одной организации, либо объединение усилий ряда предприятий, например одной отрасли, либо проведение совместных работ по хозяйственным. В последних двух случаях необходимо обратить внимание на координацию работ.

При любом из этих подходов можно выделить следующие общие основные этапы создания САПР:

- формирование идей, соответствующих функциям будущей системы;

- разработка базовых алгоритмов;

- отработка базовых алгоритмов с целью предварительного определения технико-экономических показателей;

- создание программного обеспечения, соответствующего базовым алгоритмам и вспомогательным функциям системы;

- подготовка системы к опытной эксплуатации и ее проведение;

- подготовка и сдача системы в промышленную эксплуатацию;

- дальнейшее совершенствование САПР в плане расширения ее функций. Этот этап чаще всего начинается параллельно с опытной эксплуатацией.

Каждый из указанных этапов может начинаться до окончания предыдущего. Например, генерация идей будет продолжаться и после внедрения системы в промышленную эксплуатацию. Выработанные идеи могут быть использованы для дальнейшего развития данной системы или при создании системы следующего поколения. Анализ получаемых в процессе эксплуатации результатов плодотворно влияет на генерацию идей.

Рассмотрим каждый этап.

Процесс формирования идей заключается в ознакомлении с ранее разработанными и эксплуатируемыми системами как конкретного класса аппаратуры, так и аналогов для выявления их положительных и отрицательных сторон, а также ознакомления с разрабатываемыми системами.

Теоретические положения находят отражение в публикациях, однако к ним необходимо относиться осторожно, ибо они могут представлять порой только чисто академический интерес.

Для разработки идей необходимо построить и изучить модель проектируемой системы. При этом могут быть обнаружены дополнительные факторы, которые были первоначально упущены случайно или преднамеренно из-за непонимания отдельных целей.

При генерации и разработке идей применяется наблюдение за проектировщиками с целью увидеть, как они принимают решения, связанные с разрабатываемой проблемой. Здесь важно обратить внимание, как проектировщики объясняют свои действия, о чем они говорят в процессе работы. Такие наблюдения могут дать ряд ценных идей. При выработке глобальных идей можно применять и метод «мозгового штурма» [102]. Этап генерации идей заканчивается разработкой структурной схемы вычислительной системы. Напомним, что сюда входит и программное обеспечение и аппаратные средства.

Процесс алгоритмизации предполагает разработку алгоритмов основных этапов. В начальной своей стадии он проходит так же, как и предыдущий этап: изучение действующих аналогов, изучение литературы, в том числе и теоретической. Так, в настоящее время для разработки алгоритмов успешно применяются гиперграфы, ранее не использовавшиеся для этих целей. При разработке и детализации алгоритмов в большей мере, чем на предыдущем этапе, должны учитываться конструктивные особенности объекта проектирования и технологические особенности производства. Некоторые особенности алгоритмизации идей были изложены в гл. 2. При разработке алгоритмов можно использовать и инверсный метод. Так, инверсным подходом к известному алгоритму Ли можно считать периферийную трассировку. Если алгоритм Ли ищет кратчайший путь, то периферийная трассировка прокладывает трассы по окраинам, учитывая при этом потребности других связей. В разработке базовых алгоритмов также большую роль играет наблюдение за действиями проектировщика.

Обработка базовых алгоритмов заключается в их предварительной технико-экономической оценке, что позволяет своевременно определить их эффективность. Может получиться, что потребности при реализации базовых алгоритмов в объеме памяти или машинном времени столь велики, что нет смысла на их основе создавать программное обеспечение, а необходимо искать другие подходы. На этом же этапе проводится ориентировочная оценка качества получаемых решений. Для вновь разрабатываемых систем сравниваются результаты, полученные при неавтоматизированном и автоматизированном проектировании; при наличии аналогов — с результатами, полученными с их помощью. Предварительное технико-экономическое обоснование позволяет провести корректировку алгоритмов, избежать дорогостоящих последующих переделок в программном обеспечении. Создавая ядро системы, необходимо ориентироваться на законченность программного обеспечения, а не рассчитывать на возможность последующих доработок, так как переделки программ обходятся во много раз дороже их составления.

Программное обеспечение, соответствующее базовым алгоритмам, должно создаваться программистами высшей квалификации, так как от адекватности программного обеспечения разработанным алгоритмам зависит эффективность всей системы. Трудности перехода от алгоритмов к программам подобны трудностям строителей Вавилонской башни — идеологи и программисты зачастую мыслят различными категориями и каждому из них привычен свой язык. Большинство ошибок логического характера при разработке систем кроется именно в этой причине. Это обязывает идеологов на всех этапах проверять все результаты, в том числе и промежуточные. Первоначальная реализация ядра системы обычно менее трудоемка, чем написание вспомогательных программ по преобразованию информации между основными этапами, а также по обеспечению удобства пользования.

После получения удовлетворительных результатов ядро системы обрывает

программами обслуживания, перевода информации, контроля информации на всех этапах — от начала процесса проектирования и до его завершения. При распределении сил для реализации основных и вспомогательных функций системы необходимо учитывать, что экономическая эффективность системы в первую очередь зависит от качества работы базовых алгоритмов, поэтому этим вопросом должны заниматься более квалифицированные силы. Вспомогательные программы обеспечивают удобство пользования системой, что предъявляет несколько более низкие требования к исполнению программного обеспечения, но более высокие требования к оформлению.

На этом же этапе решается вопрос о разработке входного языка системы, который должен быть прост в освоении и с достаточной полнотой описывать информацию. Необходимость простоты освоения и его максимальной приближенности к естественному вызывается тем, что эксплуатация системы (здесь мы исключаем сопровождение) осуществляется в основном неспециалистами в области программирования. Использование мнемонических обозначений, принятых ГОСТом, облегчает конструкторам освоение таких языков. Не следует применять при описании языка сложные математические формулы, так как это может только оттолкнуть будущих пользователей.

Подготовка системы к опытной эксплуатации и сама опытная эксплуатация заключается в детальной отработке алгоритмов, проверке работоспособности системы, выявлении и устранении недостатков. Во время опытной эксплуатации определяется объем и формы необходимой сопровождающей документации. Для опытной эксплуатации следует привлекать специалистов, не связанных с созданием системы. Только в этом случае можно оценить ее наиболее полно и объективно. Вместе с тем желательно, чтобы эти специалисты в дальнейшем были пользователями данной системы.

Подготовка системы к промышленной эксплуатации обычно идет параллельно с предыдущим этапом. Основная цель ее — создание документации для эксплуатации системы. Подготовку документации стараются закончить одновременно с окончанием опытной эксплуатации. (На вопросе документирования мы кратко остановимся в следующем параграфе.)

Завершение подготовки к сдаче системы в промышленную эксплуатацию далеко не означает переход разработчиков САПР к спокойной жизни. Как правило, перед началом и в процессе разработки системы с целью скорейшего получения результатов реализуется первостепеннейший круг задач. Так, в большинстве отечественных систем проектирования печатных плат реализованы только задачи размещения и трассировки, т. е. вопрос стоит о расширении функций систем. Кроме того, для повышения качества получаемых результатов и большей эффективности системы необходима разработка и реализация альтернативных алгоритмов. Создание алгоритмов для расширения функций и возможностей САПР у идеологов обычно идет с самого начала проектирования системы, реализация же задерживается из-за отсутствия необходимых трудовых ресурсов и требования сверху быстрее получения «чего-нибудь» работоспособного.

7.3. ДОКУМЕНТИРОВАНИЕ ПРИ РАЗРАБОТКЕ СИСТЕМ

Одной из отличительных черт и трудностей разработки систем по сравнению с решением обычных прикладных задач является разработка документации, которая в процессе создания систем может корректироваться.

Первым документом, появляющимся после получения ТЗ и представляющим собой результат работы архитекторов системы, считается структурная схема вычислительной системы. Современные вычислительные системы предполагают оптимальное сочетание программных и аппаратных средств, поэтому структурная схема должна учитывать и отражать как решение проблемы программным путем, так и взаимодействие аппаратных средств. Наиболее перспективным в настоящий момент является решение проблемы «сверху вниз». При этом основную проблему разбивают на подзадачи, определяя промежуточные цели. Разбиение ведется методами И, ИЛИ, описанными в гл. 2. Построение и анализ структурной схемы позволяют оптимизировать разбиение на подзадачи с учетом минимизации информационных связей между последними. Структурная схема позволяет также решить вопрос о передаче информации, ее форме, взаимодействии между различными частями системы. С помощью этой же схемы определяется ориентировочная трудоемкость реализации, оцениваются требования к аппаратным средствам.

Когда скелет будущей системы готов, встает вопрос о распределении работы между исполнителями (алгоритмистами). На данном этапе важно разработать вопросы передачи информации, что должно быть тщательно задокументировано. Из-за несогласованности вида информации в дальнейшем могут и будут возникать различные неприятности, ведущие к большим переделкам. Информация не должна передаваться через оперативную память!

Результатом работы алгоритмистов являются частные технические задания (ЧТЗ), в которых оговорены формы входной и выходной информации, даются алгоритмы решения локальных задач, сформулированы требования, предъявляемые к программам, даны ограничения. Четкое документирование передающейся от этапа к этапу информации существенно сокращает комплексную отладку системы. Программисту, работающему по ЧТЗ, необходимо помнить, что даже самый ясновидящий алгоритмист может упустить некоторые детали, поэтому одной из целей, реализующих решение поставленной задачи, является предусмотреть возможность внесения изменений и дополнений в программное обеспечение. При работе группы программистов необходимо за каждым исполнителем закрепить отдельные внешние метки.

Вопросы документирования самих программ нашли достаточное отражение в литературе, см., например [103], поэтому на них останавливаться не имеет смысла.

Роль составлявшихся ранее подробных блок-схем на современном этапе программирования понизилась: применение языков высокого уровня делает это излишним. Тем не менее представление путей решения задачи в виде укрупненных блок-схем достаточно наглядно, и графическое отображение мышления для большинства исполнителей позволяет избежать многих ошибок логического характера. Кроме того, укрупненная блок-схема большой программы позволяет выявить слабо зависимые части, сделать удачную разбивку программы. При дальнейшем пользовании программой укрупненные схемы помогают легче и быстрее ориентироваться в листинге как самому разработчику программы через некоторое время, так и лицу, сопровождающему программное обеспечение. Для небольших программ опытные программисты блок-схем не составляют вообще.

Наиболее важным при сдаче системы в промышленную эксплуатацию является вопрос состояния документации. При проектировании с помощью САПР можно выделить следующие группы пользователей: пользователь-проектировщик;

операторы ЭВМ (непосредственно работающие на ЭВМ и операторы группы подготовки информации); сопровождающие программисты. В связи с этим выходную документацию системы можно разделить на документацию общего назначения и документацию для пользователей.

Документация общего назначения содержит рекламные данные: общее назначение системы, требуемые и рекомендуемые характеристики аппаратных средств, необходимый состав группы эксплуатации, общеорганизационные рекомендации, общие характеристики программного обеспечения, схему прохождения при проектировании и пр. Одна из основных целей такой документации — помочь в определении целесообразности внедрения данной системы на предприятии.

Документация, предназначенная для пользователей-проектировщиков, содержит описание языка входных данных, правила пользования системой, описание языка управления проектированием; сюда же должны входить правила ведения баз данных. Для диалоговых систем необходимо привести директивы диалога. В этой документации отражаются и формы выходных данных. Описание входной информации должно сопровождаться хорошо иллюстрированными примерами, так как пользователь-конструктор практически не знаком с программированием. Неотъемлемой частью документации для проектировщиков должно быть описание сообщений об ошибках проектирования и правила их исправления. Все сообщения для таких пользователей должны выдаваться в удобочитаемом и понятном виде. Недостатком многих систем является чрезмерная необоснованная сложность документации для пользователей, что ведет к удлинению процесса освоения системы.

Документация по подготовке данных для операторов составляется в том случае, если перенос информации на носители осуществляется нестандартным образом, что может быть удобно при подготовке данных, но влечет за собой ошибки в исходной информации. Поэтому желательно избегать подобные нововведения.

В документации для операторов ЭВМ необходимо указывать используемые аппаратные средства, и, главное, реакцию на запросы системы и на выдаваемые сообщения об ошибках и предупреждениях. Здесь же должна быть отражена последовательность выполнения программ (в каком порядке они распечатываются при работе системы) и последовательность выдаваемых результатов. Эти данные необходимы, чтобы операторы могли определить правильность хода проектирования.

Наиболее сложным является составление полной документации для группы сопровождения. Задачей этой группы является не только дублирование программного обеспечения системы, регулярная проверка ее работоспособности, консультация пользователей-проектировщиков по работе с системой, но и модификация программ — одна из основных задач группы.

Из перечисленных выше задач сопровождения вытекают требования к необходимой документации. К сдаваемой в промышленную эксплуатацию системе надо прилагать тестовые примеры расчета; описание программ должно соответствовать последней версии (что редко наблюдается в практике), что необходимо как для выяснения сбойных ситуаций, так и при модификации программ. Для модификации программ необходимо иметь также детальное описание форм входной — выходной информации для каждого этапа, а желательно и для всех основных программ системы.

7.4. ЭКОНОМИЧЕСКИЕ АСПЕКТЫ ВНЕДРЕНИЯ САПР

Проектирование с помощью ЭВМ имеет свои явные и неявные экономические преимущества. К явным относятся такие, которые позволяют определить экономический эффект с помощью формул. Неявные преимущества трудно поддаются учету; для их определения необходимы длительные наблюдения, которые не дают четких показателей.

Рассмотрим сначала неявные преимущества. Одним из них является достоверность получаемых результатов: его трудно оценить количественно при внедрении системы, хотя оно и очевидно. Для приблизительной оценки необходимо собрать статистические данные обнаружения брака производства при неавтоматизированном проектировании и проектировании с помощью ЭВМ.

При этом необходимо различать следующие ошибки, являющиеся причинами брака: 1) ошибки разработчика аппаратуры, не выявленные ни при неавтоматизированном проектировании, ни при проектировании с помощью ЭВМ; 2) ошибки разработчика, выявляемые ЭВМ, но не выявляемые конструктором; 3) ошибки разработчика, выявляемые конструктором, но не выявляемые ЭВМ; 4) ошибки неавтоматизированного проектирования; 5) ошибки при проектировании с помощью ЭВМ.

Кроме того, при сравнительном анализе следует различать брак, неизбежно появляющийся в результате неавтоматизированного проектирования, и брак, являющийся следствием проектирования с помощью ЭВМ.

Первый тип ошибок носит обычно схемный характер. Второй — также схемный, но контроль этих ошибок поддается алгоритмизации (например, если в цепи задействованы только входы логических элементов или только выходы, контроль перегрузки и т. п.). Сюда же можно отнести ошибки, выявляемые конструктором не полностью. Так, при составлении перечней элементов разработчик в написании полных наименований допускает до 70 % ошибок в записях. Усилиями конструкторов и нормоконтролеров число ошибок снижается до 5 %, из которых 1 % приходится на технологические ошибки, которые ведут к переконструированию узлов. ЭВМ полностью исключает ошибки такого рода. Ряд ошибок конструкторы не всегда выявляют из-за того, что схема представлена не микросхемами, а базовыми элементами. Третий тип ошибок практически не встречается. Четвертый тип ошибок связан с утомляемостью конструктора, особенно при разводке многослойных печатных плат. Пятый тип ошибок может возникнуть при подготовке информации и при сбое оборудования. Однако за счет избыточности вводимой информации и всестороннего контроля такие ошибки до производства практически не доходят.

Примером брака, являющегося следствием неавтоматизированного проектирования, может служить определение площади металлизации. Точность ее определения влияет на качество металлизации, так как по площади металлизации определяют необходимую концентрацию раствора и время травления или металлизации.

Брак, являющийся следствием проектирования с помощью ЭВМ, связан с недостаточной высокой культурой производства. Такие ошибки характерны в период отработки технологии.

После освоения технологии производства при проектировании с помощью ЭВМ брака получается гораздо меньше, однако количественно оценить эти показатели трудно.

Явные преимущества (экономическая эффективность) рассчитываются по-разному для разных случаев: 1) проектирование РЭА на предприятии с помощью ЭВМ внедряется впервые; 2) ранее уже эксплуатировалась какая-то САПР; 3) расширяются функции эксплуатируемой САПР. Например, на предприятии впервые внедрена САПР, автоматизирующая этапы размещения, трассировки и выпуск управляющих перфолент для координатографов. Для этого случая исследуем сравнительную схему прохождения документации при неавтоматизированном и автоматизированном проектировании.

На вычерчивание электрической схемы устройства как при неавтоматизированном, так и при автоматизированном проектировании оговоренным выше методом затраты времени одинаковые. При неавтоматизированном проектировании конструктор по имеющейся схеме делает размещение и трассировку. Для автоматизированного проектирования необходимо закодировать схему, составить информацию о конструкции платы, отперфорировать информацию или занести ее на другие носители, провести контроль информации на ЭВМ, исправить имеющиеся ошибки, снова провести контроль, затем проектирование на ЭВМ и, при необходимости, ручную доработку схемы (при диалоге или во время коррекции на столе с последующим внесением изменений на носители) — все это необходимо учитывать по времени ручного труда конструктора, оператора ЭВМ и времени машинного счета.

Результаты проектирования обоими способами согласовываются с разработчиками. Корректировку при неавтоматизированном способе делает конструктор за кульманом, а при автоматизированном — на ЭВМ. При проектировании на ЭВМ добавляется этап получения управляющих программ для отрисовки фотошаблона. При этом существенно сокращается время выпуска сопровождающей документации (чертеж деталей платы, узел платы и прочие документы получают на графопостроителях). Существенно сокращается также время проверки конструкторской документации, исправления ее после проверки техническим контролером и нормоконтролером.

При неавтоматизированном проектировании затрачивается время на изготовление фотооригинала, при автоматизированном — время отрисовки фотошаблона.

Выше перечислены практически все основные составляющие для расчета экономической эффективности для первого случая. При численном расчете эффективности учитывается среднедневная ставка конструктора t , оператора по кодированию информации t_1 , оператора ЭВМ t_2 , себестоимость машинного времени C_m , трудоемкость одного часа машинного времени T , усредненное время обработки одной платы на ЭВМ T_m .

В расчет включается стоимость дополнительно приобретенного оборудования K_1 . Если ЭВМ не полностью занята расчетами по проектированию, то стоимость использования ЭВМ K_2 рассчитывается по формуле $K_2 = K_m \eta$, где K_m — стоимость ЭВМ с учетом монтажных работ; η — коэффициент участия. Кроме того, в предпроизводственные затраты входят и затраты на разработку САПР — K_n . Таким образом, дополнительные капитальные затраты $K_d = K_1 + K_2 + K_n$.

Себестоимость конструирования одной платы ручным способом определяется по формуле $C_1 = T_1 t (1 + H_d + H_c + H_n)$, где T_1 — трудоемкость конструирования одной платы ручным способом, чел.-дн; t — дневная тарифная ставка работников, руб.; H_d , H_c , H_n — коэффициенты, учитывающие размер дополнительной заработной платы, размер отчислений по социальному страхованию и накладные расходы.

Годовые затраты при ручном способе конструирования определяются как $\mathcal{E}_1 = C_1 T_{т.п.}$, где $T_{т.п.}$ — число типов плат в год.

Себестоимость конструирования печатной платы с использованием ЭВМ определяется по формуле

$$C_2 = T_{р.т}(1 + H_{д} + H_{с} + H_{п}) + T_{м.р}t_2(1 + H_{д} + H_{с} + H_{п}) + T_{м}C_{м},$$

где $T_{р.т}$ — трудоемкость ручных этапов проектирования; $T_{м.р}$ — трудоемкость оператора по подготовке информации.

Годовые затраты при автоматизированном проектировании $\mathcal{E}_2 = C_2 T_{т.п.}$. Годовой экономический эффект $\mathcal{E} = (\mathcal{E}_1 - \mathcal{E}_2) - E_{п}K_{д}$, где $E_{п}$ — нормативный коэффициент сравнительной эффективности.

По формуле $E = (\mathcal{E}_1 - \mathcal{E}_2) / K_{д}$ определяется коэффициент сравнительной экономической эффективности, а по формуле $T_{ок} = 1/E$ — срок окупаемости.

При расчете экономического эффекта для второго случая (когда ранее САПР уже использовалась на предприятии) применяются формулы как и в предыдущем случае. Разница обычно бывает в скорости проектирования, т. е. в экономии машинного времени и времени сопровождения конструкторами и операторами. В этом случае возможен выигрыш за счет лучших показателей качества проектирования, например сокращение времени коррекции, времени прорисовки фотошаблона и, наконец, времени производства самой платы. В последнем случае экономический эффект увеличивается за счет тиражирования с одного фотошаблона.

Большое значение имеет расширение функций САПР как по глубине, так и по ширине. Объем подготавливаемой при этом дополнительной информации невелик. Например, автоматизация только этапа компоновки выигрыша не дает, в то время как в сквозном цикле, когда информация используется для дальнейшего проектирования, выигрыш в трудоемкости существен.

Рассмотрим вопрос расширения функций САПР в плане вычерчивания принципиальных электрических схем с точки зрения экономической эффективности. Для трассировки и размещения в ЭВМ вводится информация о связях и перечень элементов. Таким образом, для автоматического получения чертежа электрической схемы необходимо еще иметь только статистически устойчивую информацию, которая должна храниться в базе данных — это логическое и геометрическое описание элементов. Проект схемы остается в ЭВМ до конца проектирования, т. е. если в системе предусмотрено и использование соотношений эквивалентности при проектировании, то нет необходимости вести ручную доработку схемы, достаточно дать «указание» машине о внесении изменений, что можно делать и при изменениях по другим причинам.

Приложение 1

ПРИМЕР ПРОГНОЗИРОВАНИЯ НА ЭТАПЕ КОМПОНОВКИ

При проектировании РЭА и ее элементной базы качество конечных результатов оценивается по таким критериям, как сложность монтажа, число конструктивных единиц различного уровня, полнота использования монтажного пространства при минимизации суммарной длины связей и т. п.

Целью прогнозирования в рассматриваемом ниже случае является выработка рекомендаций по динамическому выбору числа элементов в конструктивных единицах высшего уровня и по выбору режима проектирования, т. е. оптимизируется один из вариантов решения задачи компоновки.

Компоновка конструктивно-функциональных узлов (КФУ) является одной из первых задач конструкторского проектирования. Ее цель — назначение элементов в формируемые блоки, а затем определение оптимального числа элементов в блоках. Успешное решение данной проблемы позволяет минимизировать затраты на этапах размещения и трассировки. В процессе компоновки можно получить предварительную информацию о заполнении монтажного пространства (МП) установочными элементами, прогнозировать заполнение МП.

Рассмотрим метод прогнозирования конечного результата монтажно-коммутационного проектирования в зависимости от количества установочных элементов (УЭ), назначаемых в МП с заданными размерами.

Пусть задана модель МП компонуемого КФУ с заданными посадочными местами $\{P_i\}$: a — число мест по горизонтали и b — число мест по вертикали. Пронумеруем посадочные места от 1 до $a \cdot b$.

Построим матрицу $\|\rho_{ij}\|$ расстояний между посадочными местами с учетом соединителя (при проектировании ИС — выводов); здесь $i, j=1, 2, \dots, (a \cdot b + 1)$, где в свою очередь индекс $(a \cdot b + 1)$ соответствует соединителю. Элементы матрицы вычисляются по формуле $\rho_{ij} = |m-l| + |k-n|$, где (m, k) и (l, n) — координаты P_i и P_j на модели, соответственно. Если i либо j равно $(a \cdot b + 1)$, то для вертикального расположения соединителя будет $\rho_{ij} = |m-l|$, а для горизонтального — $\rho_{ij} = |k-n|$. Очевидно, что матрица $\|\rho_{ij}\|$ — симметричная.

Электрическую схему устройства будем описывать гиперграфом $G(X, U; R)$. Задача заключается в выделении из гиперграфа G полнореберного подграфа (подграфов). Для построения методики прогнозирования достаточно описать процесс выделения одного подграфа $G^1(X^1, U; R)$.

Процесс формирования подграфа G^1 — пошаговый, поэтому удобно иметь дело с последовательностью подграфов $G_i^1(X_i^1, U; R)$, где X_i^1 — множество вершин подграфа G_i^1 , т. е. это множество выделенных в подграф вершин после i -го шага. Последовательность множеств выделяемых в подграф вершин обладает свойством $X_i^1 \subset X_{i+1}^1$. В гиперграф G и подграф G^1 целесообразно ввести искусственную вершину x_0 , соответствующую соединителю, через который будет проходить связи при физической реализации. Таким образом, $X_0^1 = x_0$; $X = X_i^1 \cup X_i^2$

и $X_i^1 \cap X_i^2 = \emptyset$. Множество ребер в этом случае представляется в виде $U = U_i^1 \cup U_i^2 \cup U_i^3$, где

$$U_i^1 = \{u : R(x_j \in X_i^2, u) = 1, R(x_k \in X_i^2, u) = 0 \text{ для } \forall j, k\};$$

$$U_i^2 = \{u : R(x_j \in X_i^2, u) = 1, R(x_k \in X_i^2, u) = 1 \text{ для } \forall j, k\};$$

$$U_i^3 = \{u : \exists x_j \in X_i^1, x_k \in X_j^2 \Rightarrow R(x_j, u) \cdot R(x_k, u) = 1\}.$$

В процессе формирования подграфа G_i^1 происходит следующее преобразование ребер.

Пусть на очередном шаге i вершина x_j переносится из X_{i-1}^2 в X_i^1 . Тогда в ребро $u : R(x, u) = 1$ включается вершина x_0 , если выполнено условие $R(x_i \in X_{i-1}^2, u) \cdot R(x_k \in X_{i-1}^2, u) \cdot R(x_m \in X_i^1, u) = 1$ и $R(u_n \in X_{i-1}^1, u) = 0$; из ребра u удаляется вершина x_0 , если выполняется условие $R(x_i \in X_{i-1}^2, u) \cdot R(x_k \in X_{i-1}^2, u) \cdot R(x_m \in X_i^1, u) = 1$ и $R(x_n \in X_i^2, u) = 0$; в остальных случаях изменение в составе ребер не происходит.

Подграф G^1 должен удовлетворять следующим требованиям:

$$|U(x_0)| < T_1, \quad (\text{П.1})$$

$$N < a \cdot b. \quad (\text{П.2})$$

где T_1 — число контактов соединителя; N — число вершин в подграфе G^1 .

Покажем, что в процессе переноса вершин из X_{i-1}^2 в X_i^1 степень ребер в подграфе G^1 не уменьшается. Пусть полнореберный подграф G_i^1 получается переносом в полнореберный подграф G_{i-1}^1 вершины x_j . Рассмотрим в гиперграфе G любое ребро со свойством $R(x_j, u) = 1$. Пусть это будет ребро u_0 . Возможны три случая:

1. $\exists x_i \in X_{i-1}^2, x_i \neq x_k$ и $x_k \in X_{i-1}^1$ такие, что $R(x_i, u_0) \cdot R(x_k, u_0) = 1$, тогда степень ребра u_0 в G_i^1 увеличивается на единицу.

2. $\exists x_i \in X_{i-1}^2, x_i \neq x_j$ такая, что $R(x_i, u_0) = 1$ и $R(x_k, u_0) = 1$ для $\forall x_k \in X_{i-1}^1$, тогда в G_i^1 появляется новое со степенью два.

3. $\exists x_k \in X_{i-1}^1$ такая, что $R(x_k, u_0) = 1$ и для $\forall x_i \in X_{i-1}^2, x_i \neq x_j, R(x_i, u_0) = 0$, в этом случае степень u_0 в G_i^1 не изменяется, так как увеличение степени на единицу компенсируется исключением из ребра вершины x_0 .

Не теряя общности, можно считать, что вершины, переносимые в подграф G^1 , имеют нумерацию x_1, x_2, \dots, x_N .

Построим матрицу C коэффициентов связи для этих вершин, элементы которой вычисляются по формуле

$$c_{ij} = \sum_{\{x_j, u\} : R(x_t, u) \cdot R(x_j, u) = 1} \frac{1}{|U(x_t)|}, \text{ где } t, j = 1, 2, \dots, (a \cdot b + 1).$$

Такая матрица будет симметричной.

Покажем, что коэффициенты связи c_{ij} соответствуют учету в каждом ребре u ($|u| - 1$)-й связи, т. е. соответствуют числу ребер в дереве, построенном на основе рассматриваемого ребра гиперграфа.

Рассмотрим ребро $u_n = \{x_{n1}, x_{n2}, \dots, x_{nk}\}$, здесь $|u_n| = k$. Тогда коэффициент связи

$$c_{ij} = \sum_{\{x_j, u_n\} : R(x_l, u_n) \cdot R(x_j, u_n) = 1} \frac{1}{|v_n(x_{nl})|} = \frac{k-1}{k}.$$

Кроме того, в матрице C ребро u учитывается для каждого $x_{nj} \in u_n$, т. е.

$$\sum_{\{x_{nj}\}} c_{ij} = k \cdot \frac{k-1}{k} = k-1.$$

Для определения порядка формирования подграфа G^1 можно предложить следующую процедуру.

Множество вершин гиперграфа G разбиваем на группы максимально связанных элементов (см. § 3.4). На основе полученных коэффициентов связности строим планарное дерево, в котором упорядочиваем ветви и вершины по величине коэффициента связности, например по убыванию слева направо (§ 5.3).

Начинать формирование подграфа G^1 целесообразно макроэлементами т. е. связными группами. При переносе групп проверяется выполнение условий (П.1) и (П.2). При невыполнении только условия (П.2) либо расформируется последняя перенесенная группа и перенос делается поэлементно, либо вместо последней группы выбирается одна из последующих в упорядоченном дереве групп.

Нарушение условия (П.1) при соблюдении условия (П.2) не является таким же критичным, как в предыдущем случае. Если подсоединяемые группы в дереве входят в ветви, которые соединяются до корня дерева, то дальнейшее формирование соответствует образованию новых связанных групп на более высоком уровне. При этом число внешних связей может уменьшаться. Если подсоединяемые группы образуют в дереве связанных групп ветви, соединяющиеся непосредственно с корнем дерева, то размерность $|U(x_0)|$ не понижается при их подсоединении. В этом случае для выполнения условия (П.1) необходимо из G^1 удалить часть вершин.

Рассмотрим вопрос прогнозирования заполнения МП коммутационными связями. После определения вершин подграфа $G^1: x_1, x_2, \dots, x_N$ строим для этих элементов по описанному выше методу матрицу коэффициентов связей C . На первом шаге по максимальному коэффициенту матрицы выбираем соответствующую пару элементов. По матрице расстояний определяем для этих элементов посадочные места, соответствующие минимальному ρ_{kl} и на эти места размещаем выбранные два элемента. Следующим выбирается элемент из той же группы с наибольшим коэффициентом связности относительно одного из ранее размещенных элементов и устанавливается (условно) на ближайшее посадочное место от этих элементов, т. е. опять выбирается минимальное ρ_{kl} . После переноса всех элементов из очередной группы процедура повторяется со следующей группой. Выбор при этом ведется из числа незанятых посадочных мест.

Для текущего определения заполнения МП из матрицы C строим матрицу C^1 по следующему правилу. Пусть элемент x_n устанавливается на позицию l_k , тогда $c'_{kl} = c_{nj}$ для занятых позиций l_l и $c'_{nl} = 0$ для незанятых позиций l_l . Аналогично определяются элементы c'_{lk} .

Такая матрица позволяет после распределения каждого элемента определять функцию заполненности МП связями:

$$F_l = \frac{1}{2} \sum_{k, l=1}^{a-l+1} c'_{kl} \cdot \rho_{kl}.$$

При задании координат посадочных мест в дискретах МП функция F_i определяет прогнозируемое число дискретов для реализации связей. Разрешенное для реализации связей число дискретов МП F_c по мере формирования подграфа G^1 уменьшается на число дискретов, занимаемых контактами вновь переносимых элементов. Нижняя граница F_c определяется как разность числа дискретов МП S и числа дискретов, занимаемых контактными площадками элементов при их плотной установке: $F_{cн} = S - S_k$. Реально число дискретов для прокладки связей ограничено сверху значением функции F_m , учитывающей «мертвые» зоны, образующиеся в углах МП, а также около элементов за счет блокировки площади прокладываемыми трассами.

Теоретически нижней границей функции F_i является функция $F_{ш}$, отражающая число дискретов, занимаемых соответствующими связывающими деревьями Штейнера. Следующим уточнением нижней границы является функция F_p , соответствующая суммарной длине разнесенных деревьев после укладки. Исследования показывают, что $F_{ш} < F_p < F_i$, причем $F_i - F_p$ возрастает по мере увеличения числа элементов в подграфе G^1 , так как с увеличением занятости МП удлиняются реализуемые связи из-за обхода ранее проложенных трасс.

Использование предложенной методики позволяет для каждого класса проектируемых объектов определить, при каких значениях $F_m - F_p$ и F_i целесообразно использовать автоматические методы проектирования, а при каких — автоматизированные, а также определить минимальное значение $F_m - F_p$, когда схема практически не реализуема, либо стоимость ручной доработки очень велика.

СПИСОК ЛИТЕРАТУРЫ

1. **Радиоэлектроника:** Обзор по материалам зарубежной печати. Вычислительная техника. Т. 1. — М.: НИИЭИР, 1983.—59 с.
2. **Селютин В. А.** Машинное конструирование электронных устройств. — М.: Сов. радио, 1977.—384 с.
3. **Морозов К. К., Одинокоев В. Г., Курейчик В. М.** Автоматизированное проектирование конструкций радиоэлектронной аппаратуры. — М.: Радио и связь, 1983. — 280 с.
4. **Пакет программ машинного проектирования печатного монтажа ТЭЗов и ответных панелей/ А. В. Грекович, М. А. Федоров, А. Ф. Авдеев, М. В. Лебедев// Вопросы радиоэлектроники. Сер. ЭВТ. — 1980. — Вып. 11. — С. 132—135.**
5. **Рычков Л. А., Кузьмин Б. А.** Принципы построения САПР нового поколения// Приборы и системы управления. 1979.—№ 1.—С. 4—7.
6. **Система машинного проектирования топологии изделий электронной техники/ В. Н. Лошаков, Г. Э. Широ, Л. Б. Осипов, Г. Н. Брусникин// Электронная техника. Сер. 10.—1977. — Вып. 6. — С. 73—81.**
7. **Забара С. С.** Интерактивная система автоматизированного проектирования// Управляющие системы и машины. — 1978. — № 6. — С. 100—106.
8. **Odawara G. and oth.** Automatic Location System of Printed Circuit Board Design// J. Fac. Eng. Univ. Tokyo. — 1979. — Vol. XXXV, Ser. B, N 2. — P. 221—243.
9. **Промышленная система автоматизированного проектирования топологии многослойных печатных плат/ С. А. Арустамов, А. Л. Кузнецов, Г. А. Петухов, С. Ю. Яковлева// Управляющие системы и машины. — 1983. — № 6. — С. 28—32.**
10. **Девис У.** Операционные системы: Пер. с англ. — М.: Мир, 1980.—436 с.
11. **Гаврилоз М. А.** Интегрированные системы — современная тенденция в развитии систем автоматизированного проектирования// Приборы и системы управления. — 1979. — № 1. — С. 3—4.
12. **Архитектура систем автоматизированного проектирования радиоэлектронной аппаратуры/ Г. В. Орловский, И. И. Казенова, Ю. Г. Квальвассер и др.// Обмен опытом в радиопромышленности. — 1975. — Вып. 6. — С. 11—13.**
13. **Хананина Н. С., Юлин Б. И.** Автоматизированная информационно-поисковая система по радиоэлементам/ Труды НИИР. — 1980. — Вып. 2. — С. 123—126.
14. **Андреев Г. Д., Петухов Г. А., Скорубский В. И.** Компенсирующий алгоритм поиска малоповоротных путей// Вычислительная техника. Т. III. Каунасский политехн. ин-т, 1972.— С. 380—386.
15. **Петухов Г. А.** Основные направления развития САПР// Машинные методы проектирования ЭВА. — Л.: Знание, 1981. — С. 4—6.
16. **Юлин Б. И., Наймушин Г. А., Смолич Г. Г.** Ввод графической информации в ЭВМ// Управляющие системы и машины. — 1978. — № 1. — С. 83—87.
17. **Об одном перспективном методе ввода в ЭВМ чертежно-графической информации в системах автоматизации проектирования технических средств ЭВМ/ Р. М. Асцатуров, М. А. Бокач, А. П. Грищенко и др// Вопросы радиоэлектроники. Сер. ЭВТ. — 1978. — Вып. 12. — С. 114—125.**
18. **Абрайтис Л. Б., Шейнаукас Р. И., Жилевичюс В. А.** Автоматизация проектирования ЭВМ. — М.: Сов. радио, 1978. — 272 с.
19. **Бункин Б. В.** Основные направления развития отраслевых САПР// Обмен опытом в радиопромышленности. — 1981. — Вып. 2—3. — С. 1—3.
20. **Базилевич Р. П.** Декомпозиционные и топологические методы автоматизированного конструирования электронных устройств. — Львов: Высшая школа, 1981.—168 с.

21. Бершадский А. М. Критерии и алгоритмы схемной компоновки узлов РЭА и ЭВА// Вопросы радиоэлектроники. Сер. ЭВТ. — 1980. — Вып. 8. — С. 108—115.
22. Luccio F., Sami M. On the Decomposition of Networks in Minimally Interconnected Subnetworks// IEEE Trans. — 1969. — Vol. CT-16, № 2. — P. 184—188.
23. Бершадский А. М. Подсистема решения задач компоновки// Вопросы радиоэлектроники. Сер. ЭВТ.—1979.—Вып. 6.—С. 53—57.
24. Тафинцев В. А., Штейнберг В. И. О формировании библиотеки модулей для покрытия заданной логической схемы// Вопросы радиоэлектроники. Сер. ЭВТ. — 1974. — Вып. 3. — С. 23—32.
25. Абрайтис Л. Б. Алгоритм для оптимальной компоновки наиболее связанных подмножества элементов// Вычислительная техника. Т. 1.—Каун. политехн. ин-т, 1970.—С. 251—268.
26. Абрайтис Л. Б. Алгоритм определения максимально связанных наборов элементов// Автоматика и вычислительная техника. — 1970. — № 5. — С. 40—47.
27. Алгоритм компоновки принципиальных схем по микромодулям/ В. М. Бурцев, Л. Ф. Ватченко, А. Н. Наголкин и др.// Вопросы радиоэлектроники. Сер. ЭВТ. — 1972. — Вып. 3. — С. 93—97.
28. Мелихов А. Н., Курейчик В. М., Лисяк В. В. Метод отображения схемы в заданную систему блоков// Управляющие системы и машины. — 1974. — № 3. — С. 123—126.
29. Субботин В. А. Компоновка базовых элементов на типовом элементе замены, улучшающая условия для размещения и трассировки// Вопросы радиоэлектроники. Сер. ЭВТ. — 1977. — Вып. 1. — С. 74—80.
30. Проектирование монтажных плат на ЭВМ/ Под ред. К. К. Морозова. — М.: Сов. радио, 1979.—224 с.
31. Фридман А., Менон П. Теория и практика проектирования переключательных схем: Пер. с англ./ Под ред. В. А. Тафта. — М.: Мир, 1978.—582 с.
32. Применение гиперграфов для компоновки схем в ячейку/ А. Н. Мелихов, Л. С. Берштейн, В. В. Селянкин, М. И. Хиль// Изв. АН СССР. Техническая кибернетика. — 1974. — № 3. — С. 202—207.
33. Bowen F. G., Lowit G. J. Automatic Layout of Printed Boards// Electrical Communication. — 1972. — Vol. 47. — P. 287—292.
34. Гольдин В. В., Крапчин А. И., Лазарева Т. С. Некоторые методы улучшения результатов автоматической трассировки печатных плат// Вычислительная техника. Т. 2. — Каун. политехн. ин-т, 1971. — С. 113—117.
35. Абрайтис Л. Б., Балберюс П. В., Пуоджюс И. И. Система автоматизированного проектирования цифровых устройств// Приборы и системы управления. — 1975. — № 12. — С. 11—13.
36. Медведев А. С., Штейн М. Е. О задачах компоновки и размещения компонентов цифровых устройств// Управляющие системы и машины. 1979.— № 1. — С. 70—73.
37. Горин В. С., Чикин В. А., Юсим Г. В. Подсистема компоновки ТЭЗов// Изв. ЛЭТИ. — 1980. — Вып. 266. — С. 89—92.
38. Duggal R. A Comparison of New and Existing Placement Algorithms for Autoprint Circuit Board Layout// Proc. Nat. Electronics Conf. — 1969. — Vol. XXV. — P. 694—700.
39. Теория и методы автоматизации проектирования вычислительных систем./ Под ред. М. Брейера: Пер. с англ./ Под ред. Л. Д. Райкова. — М.: Мир, 1977. — 286 с.
40. Бугаев Е. С., Юрин О. Н. Размещение разногабаритных элементов с нефиксированными связями// Вопросы радиоэлектроники. Сер. ЭВТ. — 1976. — Вып. 8. — С. 82—93.
41. Закупнева Л. М., Смолич Г. Г. Решение задачи порядка установки элементов на плату с помощью ЭВМ// Труды НИИР. — 1973. — № 2. — С. 150—152.
42. Смолич Г. Г. Установка разногабаритных элементов на плате с помощью ЭВМ// Труды НИИР. — 1973. — Вып. 4. — С. 121—124.
43. Алгоритм размещения по комплексному критерию качества/ А. М. Бершадский, В. Б. Лебедев, Л. Р. Фионова, А. А. Шавро// Вопросы радиоэлектроники. Сер. ЭВТ. — 1978. — Вып. 13. — С. 116—121.

44. Рычков Л. А., Кузьмин В. А., Эйдес А. А. Алгоритмы размещения радиоэлементов разной формы// Приборы и системы управления. — 1979. — № 2. — С. 1—3.
45. Hosking K. H. A New Technique for the Placement of «Solder — Blobs» and Hence Discrete Components// Marconi Rev. — 1975. — Vol. 199, № 4. — P. 169—183.
46. Артемьев В. Б., Рябов Л. П. Алгоритмы размещения модулей различных габаритов на печатной плате// Обмен опытом в радиопромышленности. — 1977. — Вып. 2. — С. 29—31.
47. Hosking K. H. The Use of a Very Fast Routing Algorithm for Printed Circuit Board Design// Marconi Rev. — 1971. — Vol. 182, N 3. — P. 207—226.
48. Штейн М. Е., Медведев А. С. К задаче размещения компонент// Управляющие системы и машины. — 1974. — № 2. — С. 77—80.
49. Мелик-Адамян Ж. Э. Об одной комбинаторной задаче размещения// Управляющие системы и машины. Вопросы радиоэлектроники. Сер. ЭВТ. — 1975. — Вып. 6. — С. 92—96.
50. Гинзбург Б. Д. Алгоритм размещения модулей на плате// Обмен опытом в радиопромышленности. — 1972. — Вып. 4. — С. 31—33.
51. Степанов В. П. Задача размещения модулей при проектировании печатных плат// Изв. АН СССР. Техническая кибернетика. — 1979. — № 2. — С. 212—215.
52. Бахтин Б. И. Алгоритм перестановки конструктивных единиц при проектировании цифровых устройств// Вычислительная техника. Т. 2. — Каун. политехн. ин-т, 1971. — С. 99—105.
53. Айрапетянц Н. С. Метод размещения дискретных навесных элементов в автоматизированной системе проектирования печатных плат// Электронная техника. Сер. 10. — 1977. — Вып. 5. — С. 80—83.
54. Akers S. B. A Modification of Lees Path Connection Algorithm// IEEE Trans. — 1967. — Vol. EC-16, N 1. — P. 97—98.
55. Rubin F. The Lee Path Connection Algorithm// IEEE Trans. — 1974. — Vol. C-25, № 9. — P. 907—915.
56. Рычков Л. А., Алексеева Р. П. Автоматизация трассировки соединений на многослойных печатных платах// Приборы и системы управления. — 1977. — № 9. — С. 7—9.
57. Рычков Л. А., Кузьмин В. А., Дорохин А. А. Алгоритм трассировки соединений// Приборы и системы управления. — 1979. — № 2. — С. 4—6.
58. Широ Г. Э., Осипов Л. Б., Шеменев В. П. Алгоритм параллельной трассировки для двухслойных конструкций// Электронная техника. Сер. 10. — 1978. — Вып. 2. — С. 11—20.
59. Левин А. Ю., Шихаев К. Н. О порядке соединений в задаче плоской трассировки// Управляющие системы и машины. — 1974. — № 3. — С. 117—122.
60. Лазаускас Д. А. Итерационный метод трассировки двухслойных печатных плат// Вопросы радиоэлектроники. Сер. ОТ. — 1978. — Вып. 2. — С. 14—22.
61. Петренко А. И., Тетельбаум А. Я., Заблужев Н. Н. Топологические алгоритмы трассировки многослойных печатных плат. — М.: Радио и связь, 1983. — 152 с.
62. Широ Г. Э. Применение метода топологической трассировки для проектирования соединений в микросхемах// Электронная промышленность. — 1979. — № 4. — С. 69—74.
63. Лазарева Т. С. Алгоритм трассировки печатных соединений на основе представления о каналах// Автоматика и вычислительная техника. — 1969. — № 5. — С. 12—15.
64. Каплан А. В., Казакевич М. В. Метод трассировки печатных соединений в канале// Вычислительная техника. Т. 12. — Каун. политехн. ин-т, 1979. — С. 102—105.
65. Смолинскас И. А. Алгоритм трассировки связей двухслойной печатной платы, соединяющей контакты одного ряда// Вычислительная техника. Т. 12. — Каун. политехн. ин-т, 1979. — С. 98—101.
66. Зайцева Ж. Н., Алферов М. Ф. Метод параллельной трассировки на платах любого конструктивного уровня// Вопросы радиоэлектроники. Сер. ЭВТ. — 1979. — Вып. 6. — С. 40—47.

67. Макаров М. И., Рябов Л. П., Темницкий Ю. Н. Итерационный метод проведения трасс на печатных платах// Обмен опытом в радиопромышленности. — 1977. — № 2. — С. 44—48.
68. Абрайтис Л. Б., Жилевичюс В. А., Каплан А. В. К вопросу автоматической корректировки рисунка межсоединений печатных плат// Управляющие системы и машины. — 1976. — № 2. — С. 130—132.
69. Курейчик В. М., Лебедев Б. К. Система проектирования печатного монтажа/ Изв. ЛЭТИ, 1980. — Вып. 266. — С. 67—73.
70. Лазарев А. И., Наголкин А. Н. Единая система автоматизации проектирования ЭВМ// Вопросы радиоэлектроники. Сер. ЭВТ. — 1972. — Вып. 3. — С. 10—18.
71. Zadeh L. A. Fuzzy Algorithms Information and Control. — 1968. — Vol. 12. — № 2. — P. 94—102.
72. Кофман А. Введение в теорию нечетких множеств: Пер. с франц./ Под ред. С. И. Травкина. — М.: Радио и связь, 1982. — 432 с.
73. Харари Ф. Теория графов: Пер. с англ./ Под ред. Г. П. Гаврилова. — М.: Мир, 1973. — 300 с.
74. Бизам Д., Герцог Я. Многоцветная логика: Пер. с венг. — М.: Мир., 1978. — 440 с.
75. Нильсон Н. Принципы искусственного интеллекта: Пер. с англ./ Под ред. В. Л. Стефанюка. — М.: Радио и связь, 1985. — 376 с.
76. Ботвинник М. М. О решении неточных переборных задач. — М.: Сов. радио, 1979. — 152 с.
77. Аоки М. Введение в теорию оптимизации: Пер. с англ./ Под ред. Б. Г. Поляка. — М.: Наука, 1977. — 344 с.
78. Андреев Г. Д. Анализ и классификация автоматизированных систем конструкторского проектирования электронных устройств// Методы машинного проектирования цифровых устройств и систем. — Л.: Знание, 1974. — С. 4—12.
79. Селютин В. А., Улыбин Б. Н. О приближенных методах решения задачи размещения// Вычислительная техника. Т. 1. — Каун. политехн. ин-т, 1970. — С. 269—274.
80. Курант Р., Роббинс Г. Что такое математика?: Пер. с англ./ Под ред. В. Л. Гончарова. — М.-Л.: Гостехиздат, 1947. — 664 с.
81. Морозов К. К., Мелихов А. Н., Берштейн Л. С. Методы разбиения РЭА на конструктивно законченные части. — М.: Сов. радио, 1978. — 136 с.
82. Мелихов А. Н., Курейчик В. М., Берштейн Л. С. Применение графов для проектирования дискретных устройств. — М.: Наука, 1974. — 304 с.
83. Зыков А. А. Гиперграфы// Успехи математических наук. — 1974. — Вып. 6. — С. 89—154.
84. Смолич Г. Г., Смолич Л. И. Алгоритмы разбиения множества вершин гиперграфа на максимально связанные группы// Изв. АН СССР. Техническая кибернетика. — 1981. — № 4. — С. 216.
85. Kruskal J. B. On the Shortest Spanning Subtree of a Graf and Travelling Salesman Problem// Proc. Amer. Math. Soc., 1956. — N 7. — P. 48—50.
86. Прим Р. К. Кратчайшие связывающие сети и некоторые обобщения: Пер. с англ./ Кибернетический сборник. — М. — 1974. — Вып. 2. — С. 95—107.
87. Napan M. On Steiner Problem with Rectilinear Distance// J. SIAM. — 1966. — Vol. 14, N 2. — P. 255—265.
88. Штейн М. Е., Штейн Б. Е. Методы машинного проектирования цифровых устройств. — М.: Сов. радио, 1973. — 296 с.
89. Основные алгоритмы систем автоматизированного проектирования печатного монтажа/ Б. Д. Гинзбург, Л. А. Балагушкина, Г. М. Додин и др.// Вопросы радиоэлектроники. Сер. ЭВТ. 1972. — Вып. 3. — С. 107—117.
90. Петренко А. И., Тетельбаум А. Я. Формальное конструирование электронно-вычислительного аппарата. — М.: Сов. радио, 1979. — 256 с.
91. Абрайтис Л. Б., Шейнаукас Р. И., Шимайтис А. П. Распределение инвариантных контактов, соблюдая плоскостные ограничения. В кн.: Вычислительная техника. Т. 3. — Каун. политехн. ин-т., 1972. — С. 368—372.
92. Компоновка базовых элементов логической схемы по корпусам микросхем/ Б. И. Юлин, Л. М. Закупнева, Г. Г. Смолич, З. П. Мазова// Радиотехника. — 1979. — Т. 34, № 10. — С. 75—77.

93. Широ Г. Э. Трассировка попарно-многослойных печатных плат с использованием стандартных параллельных моделей// Электронная техника. Сер. 10. — 1980. — Вып. 2. — С. 77—86.
94. Грекович А. В. Опыт применения ЭВМ для разработки и изготовления конструкторской документации// Вопросы радиоэлектроники. Сер. ЭВТ. — 1969. — Вып. 5. — С. 98—104.
95. Дендобренко Б. Н., Малика А. С. Автоматизация конструирования РЭА. — М.: Высшая школа, 1980. — 384 с.
96. Майоров С. А. и др. О задаче размещения конструктивных единиц при проектировании цифровых устройств// Вычислительная техника Т. 1.—Каун. политехн. ин-т, 1970. — С. 275—279.
97. Gupta H., Sharma J. A Two-Dimensional Placement Algorithm for the Layout of Electronics Circuits// Microelectronics and Reliability. — 1980. — Vol. 20, N 3. — P. 323—328.
98. Андреев Г. Д., Экало Ю. В., Скорубский В. И. Алгоритмы размещения// Применение вычислительных машин в проектировании и производстве пленочного монтажа. — Л.: Знание, 1975. — С. 52—56.
99. Goto S., Kuh E. S. An Approach to the Two Dimensional Placement Problem in Circuit Layout// IEEE Trans. — 1978. — Vol. CAS-25, N 4. — P. 208—214.
100. Смолич Г. Г. Повышение качества печатного монтажа// Тр. НИИР. — 1978.—№ 2. — С. 66—71.
101. Андреев Г. Д., Экало Ю. В. Алгоритм оптимального и локально-оптимального разбиения печатных соединений на два слоя платы// Обмен опытом в радиопромышленности. — 1977. — № 2. — С. 33—37.
102. Диксон Дж. Проектирование систем: Пер. с англ.—М.: Мир, 1969.—440 с.
103. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ: Пер. с англ./ Под ред. Э. А. Трахтенгерца.—М.: Мир, 1981.—320 с.
104. Широ Г. Э. Автоматическая система АСП-53 проектирования многослойных печатных плат топологическим методом// Управляющие системы и машины. — 1979. — № 5. — С. 113—119.
105. Cullingford R. E., Joseph L. J. A Heuristically «Optimal» Knowledge Base Organization Technique// Automatica. — 1983. — Vol. 19, № 6, — P. 647—654.
106. Минский М. Фреймы для представления знаний. — М.: Энергия, 1979.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава 1. Краткий обзор состояния автоматизации конструирования радиоэлектронной аппаратуры	5
1.1. Основные тенденции применения ЭВМ при конструировании радиоэлектронной аппаратуры	5
1.2. Развитие алгоритмов компоновки	10
1.3. Некоторые возможности оптимизации алгоритмов размещения	12
1.4. Тенденции развития алгоритмов трассировки	15
1.5. Некоторые возможности расширения функций САПР	18
Глава 2. Математический аппарат оптимизации алгоритмов конструирования	24
2.1. Основные понятия теории расплывчатых множеств	24
2.2. Определения и некоторые понятия из теории гиперграфов	28
2.3. Некоторые вопросы теории принятия решений	32
2.4. Методы сокращения перебора	38
Глава 3. Математические методы и модели задач оптимизации конструирования радиоэлектронной аппаратуры	47
3.1. Анализ процесса конструирования печатных плат	47
3.2. Разработка моделей посадочных мест и рабочего поля трассировки	53
3.3. Разработка согласованных критериев алгоритмов компоновки, размещения и трассировки	57
3.4. Разработка критериев и алгоритмов разбиения множества вершин гиперграфа на группы максимально связанных элементов	59
3.5. Построение квазиоптимального дерева на ортогональной сетке	66
Глава 4. Решение задачи компоновки с использованием соотношений эквивалентности	72
4.1. Распределение базовых элементов логической схемы по корпусам микросхем	72
4.2. Определение эффективности распределения базовых элементов по корпусам	77
4.3. Использование эквивалентности базовых элементов и их входов на этапе компоновки	80
4.4. Особенности реализации этапа компоновки	85
Глава 5. Последовательно-групповой подход к размещению элементов	88
5.1. Общие требования к задаче размещения	88
5.2. Размещение разногабаритных элементов	90
5.3. Оптимизация порядка установки микросхем	96
5.4. Разработка функции оптимизации и ее связь с трассировкой	101
5.5. Применение расширенной модели платы при установке элементов	104
5.6. Особенности размещения элементов при проектировании чертежа электрической схемы	108
Глава 6. Параллельный метод пофрагментной прокладки трасс	111
6.1. Общая схема пофрагментного подхода к трассировке	111
6.2. Основные принципы предварительной укладки ветвей деревьев на коммутационное поле платы	114
6.3. Окончательное распределение ветвей деревьев на плате	119
6.4. Особенности реализации пофрагментного подхода, согласованного с этапами компоновки и размещения	122
6.5. Вопросы расщепления совмещенного рисунка	125
Глава 7. Некоторые вопросы технико-экономической эффективности внедрения САПР	131
7.1. Вопросы внедрения САПР на предприятии	131
7.2. Методология разработки САПР	135
7.3. Документирование при разработке систем	137
7.4. Экономические аспекты внедрения САПР	140
Приложение. Пример прогнозирования на этапе компоновки	143
Список литературы	147