

Министерство образования и науки Российской Федерации

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ
И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

ФАКУЛЬТЕТ ДИСТАНЦИОННОГО ОБУЧЕНИЯ (ФДО)

Е. Ф. Жигалова

**АВТОМАТИЗАЦИЯ КОНСТРУКТОРСКОГО
И ТЕХНОЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ**

Учебное пособие

Томск
2016

УДК 621.396.6.001.2-52 + 658.512.22.011.56

ББК 32.844-06я73

Ж 681

Рецензент:

Т. В. Ганджа, канд. техн. наук, доцент кафедры моделирования и системного анализа Томского государственного университета систем управления и радиоэлектроники

Жигалова Е. Ф.

Ж 681 Автоматизация конструкторского и технологического проектирования : учебное пособие / Е. Ф. Жигалова. – Томск : ФДО, ТУСУР, 2016. – 201 с.

В учебном пособии изложены принципы современных систем автоматизированного проектирования радиоэлектронных средств (РЭС). Рассмотрены постановки типовых задач и алгоритмы, связанные с разработкой математического обеспечения конструкторского и технологического проектирования. Приводится описание математических моделей функциональных элементов устройств РЭС.

Пособие содержит общие сведения о системах автоматизированного проектирования РЭС, описания радиоэлектронных средств, сущность и этапы проектирования РЭС, состав и принципы построения САПР и описание пакетов программ автоматизированного проектирования, математические модели радиоэлектронных объектов проектирования и цифровых устройств, математические модели конструкций РЭС, описание типовых задач конструкторского проектирования и алгоритмы их решения.

Для бакалавров технических университетов, обучающихся по направлению «Информатика и вычислительная техника», профиль – «Системы автоматизированного проектирования».

© Жигалова Е. Ф., 2016

© Оформление.

ФДО, ТУСУР, 2016

Оглавление

Введение	5
1 Методология автоматизированного проектирования. Общие сведения о проектировании электронных средств	8
1.1 Основные понятия и определения.....	8
1.2 Принципы проектирования.....	15
1.2.1 Иерархические уровни описаний проектируемых объектов	16
1.2.2 Унификация проектных решений и процедур.....	18
1.2.3 Нисходящее и восходящее проектирование	19
1.3 Сущность и этапы проектирования радиоэлектронных устройств	21
1.3.1 Описания радиоэлектронных средств	21
1.3.2 Этапы (аспекты) проектирования РЭС.....	24
2 Математическое моделирование технических объектов	28
2.1 Общие сведения о математических моделях	28
2.2 Требования к математическим моделям.....	32
2.3 Классификация математических моделей.....	35
2.4 Методика получения математических моделей	44
2.5 Математические модели для задач конструирования электронных средств	45
2.6 Математическое моделирование цифровых устройств	51
2.6.1 Описание языков моделирования и элементов цифровых устройств в моделях логического уровня.....	53
2.6.2 Синхронное моделирование цифровых устройств двоичными алфавитами	57
2.6.3 Моделирование неисправностей в цифровых устройствах и синтез диагностических тестов.....	66
3 Типовые проектные процедуры	70
3.1 Маршруты проектирования и принципы их построения.....	73
3.2 Место моделирования в проектировании РЭС	77
3.3 Оптимальное проектирование РЭС на основе решения задачи линейного программирования.....	84
4 Типовые задачи конструкторского проектирования электронных средств и алгоритмы их решения	108
4.1 Алгоритмы и модели компоновки.....	110

4.2	Размещение конструктивных элементов в монтажном пространстве	133
4.3	Трассировка соединений	144
4.3.1	Трассировка проводных соединений	144
4.3.2	Трассировка печатных соединений	154
5	Системы автоматизированного проектирования ЭС	160
5.1	Общая характеристика процесса проектирования и базовые определения	160
5.2	Принципы построения САПР	163
5.3	Системы автоматизированного проектирования РЭС и их место среди других автоматизированных систем	166
5.4	Основы построения автоматизированных систем технологической подготовки производства	172
5.4.1	Задачи автоматизации технологической подготовки производства	176
5.4.2	Пакеты программ автоматизации проектирования РЭС	183
	Заключение	194
	Литература	196
	Список условных обозначений и сокращений	198
	Глоссарий	200

Введение

Автоматизация проектирования имеет огромное значение для развития науки, техники и производства в современном обществе. В настоящее время автоматизация проектирования – основной способ повышения производительности и эффективности труда инженерно-технических работников, занимающихся проектированием. Использование автоматизации проектирования позволяет создавать все более сложные технические объекты и гибко реагировать на появление новых решений в той или иной области техники.

Процесс автоматизации проектирования радиоэлектронных средств по содержанию и последовательности решаемых задач весьма разнообразен в зависимости от функциональной и конструктивной сложности разрабатываемых радиотехнических систем, комплексов, устройств и узлов.

Создание микроэлектронных элементов электронно-вычислительной аппаратуры (ЭВА) включает в себя проектирование технологических процессов, приспособлений, оснастки и инструмента при технологической подготовке производства (ТПП). Основной целью при этом является проектирование документации для организации ТПП, контроля качества и промышленной эксплуатации изделий.

Разработка новых классов приборов микроэлектроники, расширение их номенклатуры и объёма выпуска при ограниченных сроках освоения в производстве определили новые производственные тенденции и направления. Главной их особенностью является быстрое освоение в производстве новых типов изделий без остановки производственного процесса выпуска прежних изделий с помощью быстрой перестройки технологического процесса и использования существующего парка технологического оборудования.

Реализация этих тенденций возможна на основе широкой автоматизации производственных процессов с применением адаптивных автоматических устройств на основных и вспомогательных операциях, а также реализации программы использования специальных средств управления и синтеза технологических процессов. Все эти проблемы решаются с помощью гибких автоматизированных систем анализа и обработки информации, содержащейся в базах данных (БД) и базах знаний (БЗ) в виде файлов программной обработки деталей, технологических маршрутов и параметров, справочников, необходимых

для систем автоматизированного проектирования (САПР) на всех стадиях производства.

Практическая реализация методов и идей автоматизированного проектирования происходит в рамках САПР. Однако мало создать высокопроизводительные современные САПР. Надо уметь их эффективно использовать. Для этого нужны квалифицированные инженеры-пользователи САПР. В современном «компьютеризированном» обществе инженер любой специальности, занимающийся разработкой технических устройств, должен уметь использовать средства автоматизированного проектирования. Это позволяет повысить эффективность проектирования, улучшить его качество, снизить материальные затраты и уменьшить число разработчиков.

В учебном пособии представлены основные аспекты математического моделирования и решения типовых задач оптимального конструкторского проектирования радиоэлектронных средств (РЭС).

Соглашения, принятые в учебном пособии

Для улучшения восприятия материала в данном учебном пособии используются пиктограммы и специальное выделение важной информации.



.....
 Эта пиктограмма означает определение или новое понятие.



.....
 Эта пиктограмма означает «Внимание!». Здесь выделена важная информация, требующая акцента на ней. Автор может поделиться с читателем опытом, чтобы помочь избежать некоторых ошибок.



.....
 В блоке «На заметку» автор может указать дополнительные сведения или другой взгляд на изучаемый предмет, чтобы помочь читателю лучше понять основные идеи.



.....
 Эта пиктограмма означает теорему.



Эта пиктограмма означает лемму.



Пример

Эта пиктограмма означает пример. В данном блоке автор может привести практический пример для пояснения и разбора основных моментов, отраженных в теоретическом материале.



Выводы

Эта пиктограмма означает выводы. Здесь автор подводит итоги, обобщает изложенный материал или проводит анализ.



Контрольные вопросы по главе

1 Методология автоматизированного проектирования.

Общие сведения о проектировании электронных средств

1.1 Основные понятия и определения

При построении новых объектов техники по заданному описанию несуществующего объекта выполняется его материализация в работоспособную надежную конструкцию.



.....

Проектирование – это процесс создания описания, необходимого для построения в заданных условиях еще не существующего объекта, на основе первичного описания этого объекта. Процесс создания описания нового объекта может выполняться различными способами. Выделим три основных.

.....



.....

Если весь процесс проектирования осуществляет человек, то проектирование называют *неавтоматизированным*. В настоящее время неавтоматизированное проектирование таких сложных объектов, как электронно-вычислительная аппаратура (ЭВА), практически не применяют.

Наибольшее распространение получило проектирование, при котором происходит взаимодействие человека и ЭВМ. Такое проектирование называют *автоматизированным*. Автоматизированное проектирование, как правило, осуществляется в режиме диалога человека с ЭВМ на основе применения специальных языков общения с ЭВМ.

Проектирование, при котором все преобразования описаний объекта и алгоритма его функционирования осуществляются без участия человека, называют *автоматическим*.

.....

Рассмотрим ряд понятий, которые используются, например, при разработке электронно-вычислительной аппаратуры.



.....

*Первичное описание ЭВА, представленное в заданной форме, называется **заданием на проектирование**. В задании на проектирование электронно-вычислительной аппаратуры должны быть сведения о её назначении, её параметрах, способах функционирования, конструктивной реализации, изготовления и т. п.*

***Проектным решением** называется промежуточное или конечное описание объекта проектирования, необходимое и достаточное для рассмотрения и определения дальнейшего направления или окончания проектирования.*

***Результатом проектирования** является проектное решение или их совокупность, удовлетворяющие заданным требованиям, необходимым для создания нового объекта. В заданные требования должны быть обязательно включены требования к форме представляемого проектного решения.*

*Документ, выполненный по заданной форме, в котором представлено какое-либо проектное решение, полученное при проектировании, называется **проектным**.*

*Совокупность проектных документов в соответствии с установленным перечнем, в котором представлен результат проектирования, называется **проектом**.*

*Под **проектной процедурой** понимают формализованную совокупность действий, выполнение которых оканчивается проектным решением. Например, проектными процедурами являются оптимизация, контроль, поиск решения, корректировка, компоновка, проверка правильности трассировки и т. п.*

*Действие или формализованная совокупность действий, составляющих часть проектной процедуры, алгоритм которых остается неизменным для ряда проектных процедур, называется **проектной операцией**.*

.....



Пример

Примерами проектных операций являются составление таблиц с данными вычисления, вычерчивание топологии, ввод и вывод данных, набивка перфокарт и т. п.



*Проектная процедура, алгоритм которой остается неизменным для различных объектов проектирования или различных стадий проектирования одного и того же объекта, называется **унифицированной проектной процедурой**.*

Выполнение проектных работ можно распределить как во времени, так и по подразделениям проектной организации.



При временном распределении работ по созданию новых объектов процесс проектирования разделяется на *стадии* и *этапы*.

Различают восемь стадий:

- 1) предпроектные исследования;
- 2) техническое задание;
- 3) техническое предложение;
- 4) эскизный проект;
- 5) технический проект;
- 6) рабочий проект;
- 7) изготовление;
- 8) отладка, испытание и ввод в действие.

При создании новых объектов выделяют следующие *этапы*:

- этап научно-исследовательских работ (НИР). Объединяет стадии: предпроектные исследования, техническое задание и часть технического предложения. Здесь проводят исследования по поиску новых принципов функционирования, новых структур, физических процессов, новой элементной базы, технических средств и т. п.;
- этап опытно-конструкторских работ (ОКР). Объединяет стадии: часть технического предложения, эскизный проект, технический проект.

Здесь отражаются вопросы детальной конструкторской проработки проекта;

- этап рабочего проектирования. Объединяет стадии: рабочий проект, изготовление, отладка и испытание, ввод в действие. Здесь прорабатываются схемные, конструкторские и технологические решения, проводят испытания, изготовление [1, 4].

Для этапа НИР в основном используют системы автоматизации научных исследований и экспериментов (АСНИ).



.....

Распределение работ между подразделениями производят с использованием *блочно-иерархического подхода* (БИП) к проектированию. Этот подход основан на структурировании описаний объекта с разделением описаний на ряд иерархических уровней по степени детальности отображения в них свойств объекта и его частей. Каждому иерархическому уровню присущи свои формы документации, математический аппарат для построения моделей и алгоритмов исследования. Совокупность языков, моделей, постановок задач, методов получения описаний некоторого иерархического уровня часто называют уровнем проектирования.

Методология блочно-иерархического подхода базируется на трех концепциях:

- 1) *разбиение и локальная оптимизация;*
 - 2) *абстрагирование;*
 - 3) *повторяемость.*
-

Концепция разбиения позволяет сложную задачу проектирования объекта свести к решению более простых задач с учетом взаимодействий между ними. Локальная оптимизация подразумевает улучшение параметров внутри каждой простой задачи. Абстрагируемость заключается в построении формальных математических моделей, отражающих только значимые в данных условиях свойства объектов. Повторяемость заключается в использовании существующего опыта проектирования.



.....

Основное достоинство блочно-иерархического подхода – это упрощение процесса проектирования и получение возможности решать задачи проектирования доступными средствами.

.....

Использование блочно-иерархического подхода помогает: упростить решение проблемы хранения данных, сократить размерность выполняемых программ и время проектирования, применять системы автоматизированного проектирования (САПР) один раз для объекта (его части) независимо от числа идентичных объектов (его частей).

Весь процесс проектирования можно представить как последовательность этапов, связывающих концептуальное описание объекта и создание этого объекта. Указанную связь реализуют в одном из двух направлений: *восходящем* или *нисходящем*. Восходящее проектирование, т. е. проектирование *снизу вверх*, характеризуется решением сначала задач низких иерархических уровней с последовательным переходом к решению задач более высоких уровней. Нисходящее проектирование, т. е. проектирование *сверху вниз*, является противоположным по отношению к восходящему проектированию.

Отметим, например, что используемая в настоящее время концепция проектирования интегральных микросхем с большой степенью интеграции по модульному принципу – это концепция блочно-иерархического подхода. В системе блочно-иерархического подхода конструктор выполняет функциональные, интуитивные и интеллектуальные преобразования на верхних уровнях, а ЭВМ выполняет проектирование на нижних уровнях.



Уровни проектирования можно выделять не только по степени подробности отражения свойств объекта, но и по характеру отражаемых свойств. Если в первом случае уровни называют горизонтальными, или иерархическими, то во втором – вертикальными, или аспектами.

При выделении горизонтальных уровней проектирования производится разделение объекта на блоки и рассмотрение вместо объекта его отдельных блоков. Если на некотором уровне i_1 имеем объект s , то на соседнем, более низком уровне i_2 происходят разделение s на блоки s_1, s_2, \dots, s_j и рассмотрение каждого блока s_1, s_2, \dots, s_j на уровне i_2 с большей степенью детализации, чем на уровне i_1 .

В общем случае при проектировании технических объектов можно выделить несколько вертикальных уровней, основные из них – функциональный,

конструкторский, технологический. Описание каждого вертикального уровня в свою очередь делят на иерархические уровни.

На рисунке 1.1 приведен пример структурированного описания проектирования ЭВМ:

Вертикальные уровни

Горизонтальные уровни	Функциональный	Алгоритмический	Конструкторский	Технологический
	Системный	Программирование системы	Шкаф, стойки	Принципиальная схема технологического процесса
	Логический		Панель	
	Схемотехнический	Программирование модулей	ТЭЗ	Маршрутная технология
	Компонентный	Программирование микропрограмм	Модуль	Технологические операции
Кристалл				
Ячейка				

Рис. 1.1 – Пример структурированного описания проектирования ЭВМ



Функциональное проектирование включает в себя анализ технического задания (ТЗ) и на его основе выбор с системных позиций методики построения и путей реализации вычислительного процесса в ЭВА; связано с анализом и синтезом блоков ЭВА; заключается в разработке функциональных и принципиальных схем. Здесь определяют принципы функционирования и важнейшие параметры и характеристики ЭВА.



Основные задачи функционального проектирования следующие: разработка структурных схем, определение требований к выходным параметрам; анализ и формирование ТЗ на разработку отдельных блоков ЭВА; синтез функциональных и принципиальных схем полученных блоков; контроль и выработка диагностических тестов; проверка работоспособности синтезируемых блоков; расче-

ты параметров пассивных компонентов и определение требований к параметрам активных компонентов; формулировка ТЗ на проектирование компонентов; выбор физической структуры, топологии компонентов; расчеты параметров диффузионных профилей и полупроводниковых компонентов, электрических параметров, параметров технологических процессов эпитаксии, диффузии, окисления и др.; вероятностные требования к выходным параметрам компонентов.

.....

.....



Алгоритмическое проектирование заключается в разработке алгоритмов функционирования и создании математического обеспечения электронно-вычислительной аппаратуры.

Конструкторское проектирование заключается в реализации принципиальных схем в заданном конструктивном базисе. При этом решаются вопросы выбора форм и материалов, выбора типоразмеров, компоновки, размещения элементов, трассировки соединений, контроля.

.....

.....



Основные задачи конструкторского проектирования: покрытие функциональных схем, т. е. получение принципиальных электрических схем; конструкторский расчет геометрических размеров компонентов и площади размещения; компоновка элементов; размещение элементов с учетом конструкторских схемотехнических и технологических ограничений; трассировка соединений; контроль топологии; проектирование фотошаблонов; выпуск конструкторско-технологической документации.

.....

.....



Технологическое проектирование заключается в решении задач технологической подготовки производства – разработке принципиальной схемы, маршрутов, операций и переходов технологических процессов изготовления деталей, сборки и монтажа узлов, включая выбор оснастки, инструмента, технологического оборудования и т. п.

.....

.....



.....

Функциональное проектирование электронно-вычислительной аппаратуры состоит из четырех основных горизонтальных уровней: системного, логического, схемотехнического, компонентного.

.....

На *системном* уровне определяют общую структурную схему, структурные схемы основных блоков.

На *логическом* уровне создают функциональные и принципиальные схемы ЭВА. Здесь выделяют подуровни – регистровый и вентильный. На регистровом подуровне проектируются устройства из модулей (функциональных узлов) типа регистров, счетчиков, сумматоров, интеграторов и т. п. На вентильном уровне проектируются устройства и модули из отдельных логических вентилей и триггеров.



.....

Алгоритмическое проектирование используется для разработки программного обеспечения электронно-вычислительной аппаратуры. Для больших программных систем обычно используют набор иерархических уровней, два из которых являются основными. На первом планируют всю программную систему и разрабатывают схемы алгоритмов на основе программных модулей. На втором производят программирование модулей на заданном алгоритмическом языке.

Конструкторское проектирование состоит из иерархических уровней проектирования компонентов, БИС, типовых элементов замены, панелей, стоек, шкафов. Здесь в основном используется восходящее проектирование.

Технологическое проектирование состоит из уровней проектирования принципиальной схемы технологического процесса, технологических маршрутов, технологических операций [1, 4].

.....

1.2 Принципы проектирования

Возможность проектирования сложных технических объектов основана на следующих *основных принципах*:

- декомпозиция и иерархичность описаний объектов;
- многоэтапность и итерационность проектирования;

- типизация и унификация проектных решений и средств проектирования [1, 4].

1.2.1 Иерархические уровни описаний проектируемых объектов

Описание технических объектов по сложности должно быть таким, чтобы его мог воспринять человек, а также доступным для существующих средств проектирования. Однако в случае сложных систем это требование не выполняется, поэтому приходится разделять описание объектов (и представление о них) на иерархические уровни и аспекты.

Это позволяет разработчику (коллективу разработчиков) осмыслить задачу проектирования, получить промежуточные описания, а также распределить работу по проектированию сложных объектов между различными подразделениями, что повышает эффективность и производительность проектирования.



Выводы

Таким образом, использование разделения описаний объекта по степени детализации отображаемых свойств и характеристик лежит в основе *блочного-иерархического* подхода к проектированию и приводит к появлению иерархических (горизонтальных) уровней. На каждом иерархическом уровне используются свои понятия системы и элементов (рис. 1.2).

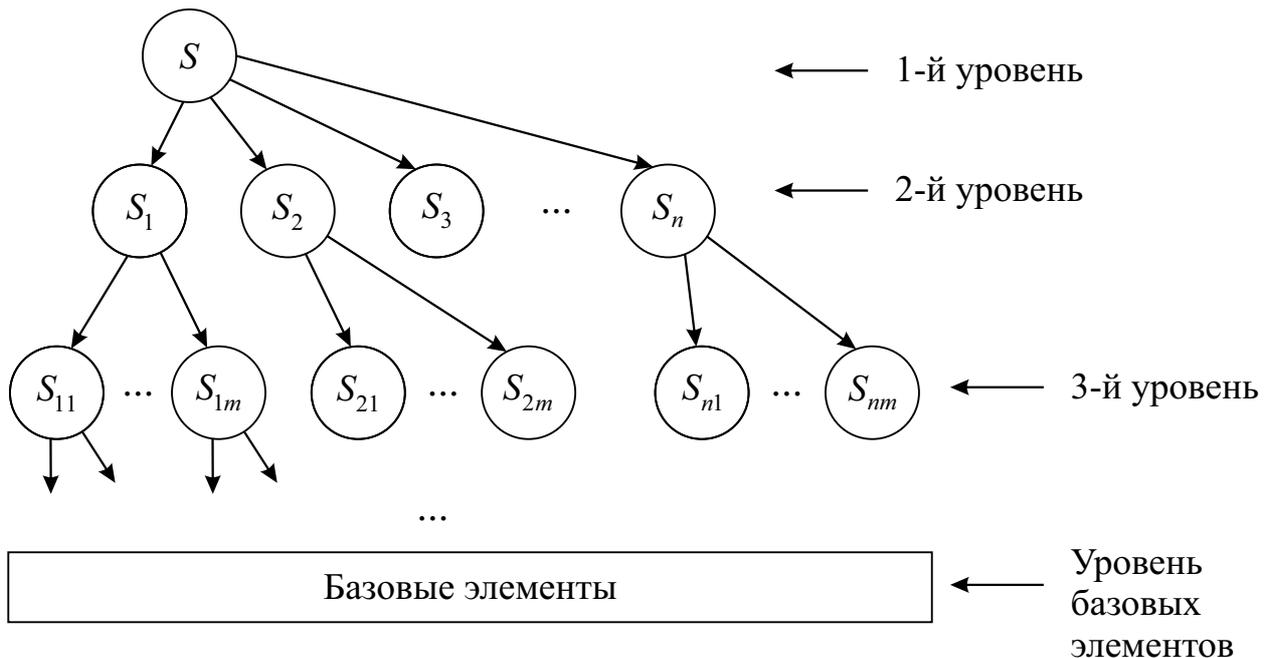


Рис. 1.2 – Иерархические уровни системы S

1-й уровень сложности.

Объект S рассматривается как комплекс систем из взаимосвязанных элементов (блоков) S_i .

2-й уровень сложности.

Каждый из элементов S_i представляет собой также довольно сложный объект, который, в свою очередь, рассматривается как система из элементов S_{ij} , формируемых из элементов j -го уровня ($j = i - 1$) и т. д.

Как правило, выделение элементов S_{ij} на уровне j происходит по функциональному признаку.

Разделение на уровни продолжается, пока на некотором уровне не получат элементы, описание которых дальнейшему делению не подлежит. Эти элементы по отношению к системе S называют *базовыми* элементами. Базовые элементы являются *конструктивно неделимыми*.



Пример

При проектировании радиоэлектронной аппаратуры к базовым элементам относятся элементы принципиальной схемы (резисторы, конденсаторы, транзисторы, микросхемы и т. д.).



Следующий уровень – функциональные узлы. Функциональные узлы входят в состав *блоков* (или устройств).



Пример

Из блоков (устройств) komponуют радиоэлектронную (электронно-вычислительную) систему (рис. 1.3):

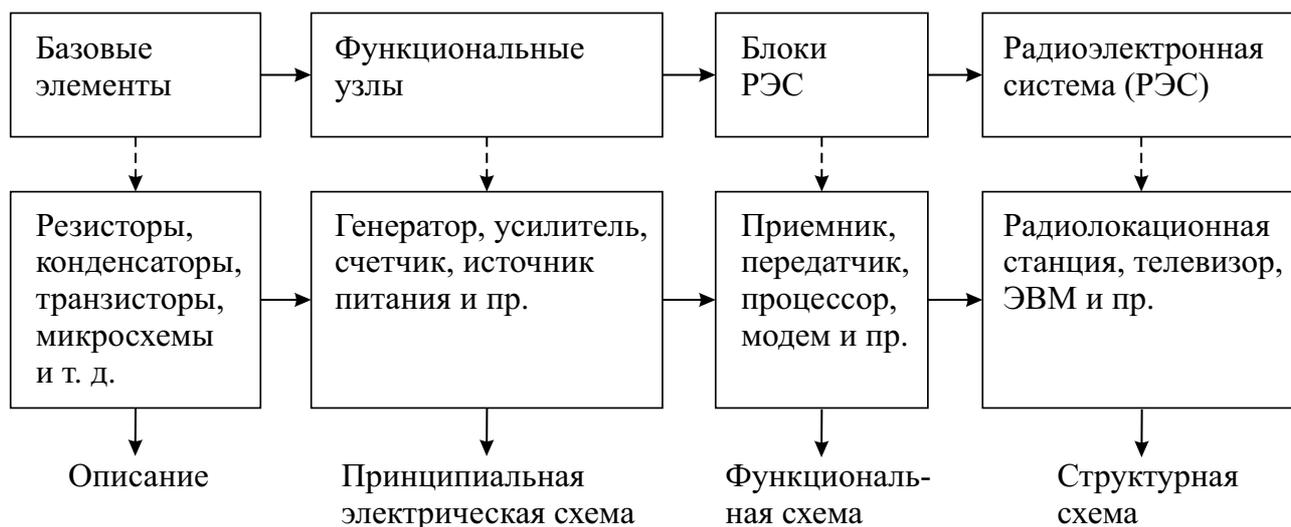


Рис. 1.3 – Структура радиоэлектронной системы

1.2.2 Унификация проектных решений и процедур



Обычно **унификация** объектов имеет целью улучшение технико-экономических показателей производства и эксплуатации изделий. Унификация блоков ведет к снижению стоимости, упрощению изготовления и эксплуатации объекта. Использование типовых и унифицированных проектных решений и процедур приводит также к упрощению и ускорению процесса проектирования (типовые элементы и проектные решения разрабатываются один раз, а затем используются в различных проектах многократно).

Существование типовых проектных решений и процедур для объектов некоторого класса позволяет *автоматизировать их проектирование*, т. е. является предпосылкой для создания системы автоматизированного проектирования. Создание новых унифицированных проектных решений и процедур – постепенный процесс, который происходит по мере накопления информации и опыта.

Однако существуют объекты, где необходим индивидуальный подход к проектированию (космические корабли, новые ЭВМ и другие сложные объекты). В этом случае не удастся применить существующие «унифицированные» решения и приходится выполнять проектирование полностью «с нуля». Следовательно, основной минус унификации в том, что нельзя предусмотреть все возможные случаи.

1.2.3 Нисходящее и восходящее проектирование



Принцип итерационности проектирования связан с декомпозицией описаний объекта. Процесс проектирования технического объекта разбивается на этапы по уровням описания объекта и/или аспектам (рис. 1.4).

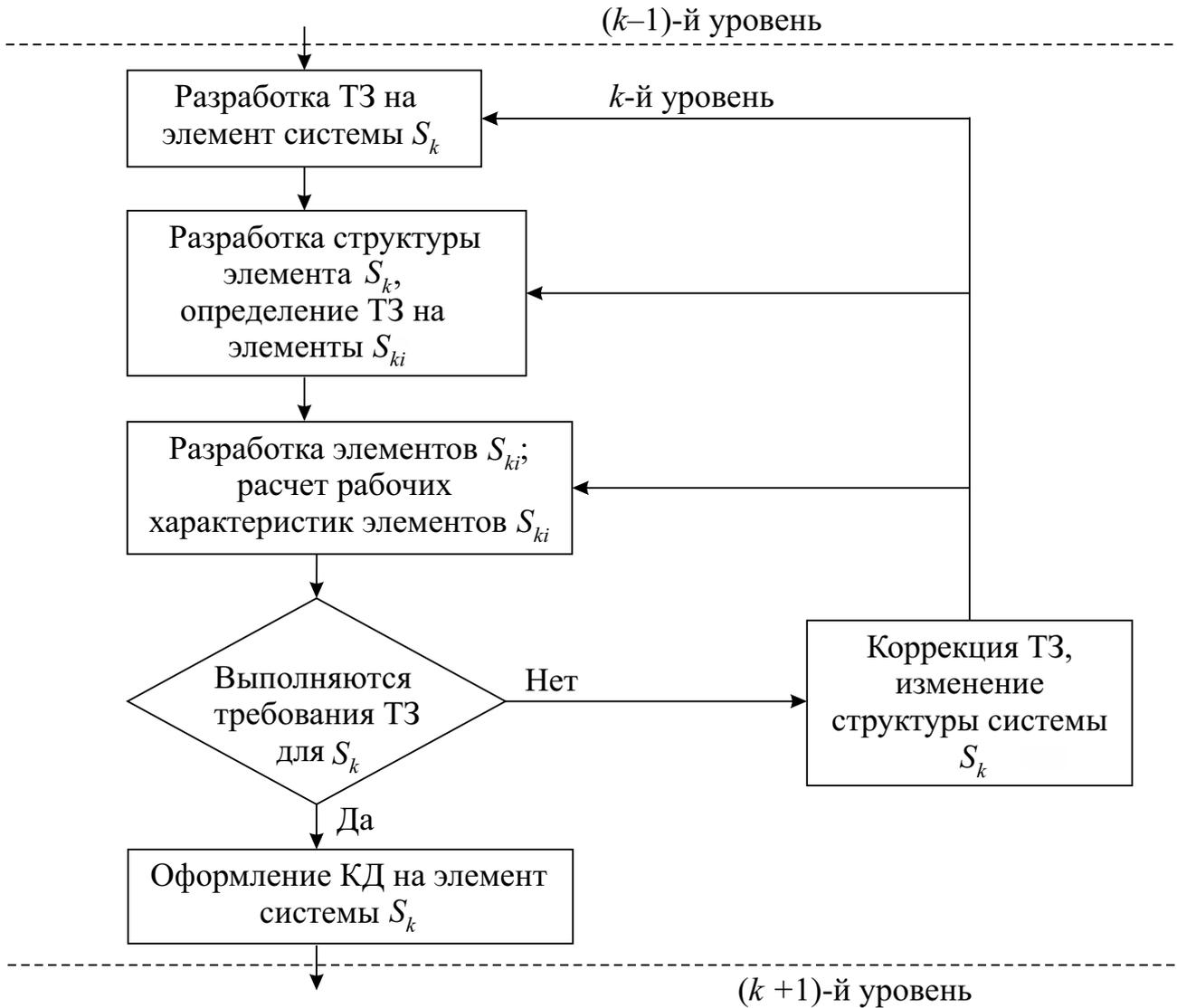


Рис. 1.4 – Итерационный процесс проектирования



Проектирование, которое осуществляется от задач более низких иерархических уровней к задачам более высоких уровней, называется *восходящим* (или *проектированием снизу вверх*). Если же раньше выполняются этапы, связанные с высшими иерархиче-

скими уровнями, то проектирование называют *нисходящим* (*проектирование сверху вниз*).

.....

У каждого из этих двух подходов к проектированию есть свои преимущества и недостатки: при нисходящем проектировании система разрабатывается в условиях, когда сведения об элементах и их возможностях полностью не определены и носят предположительный характер; при восходящем проектировании элементы проектируются раньше системы, поэтому предположительный характер носят требования к самой системе. В обоих случаях из-за отсутствия исчерпывающей исходной информации имеют место отклонения от оптимального технического решения. Однако эти отклонения неизбежны при блочно-иерархическом подходе к проектированию. Другого способа проектирования сложных технологических объектов пока не существует.

Иногда можно пожертвовать оптимальностью характеристик, существенно выиграв во времени и стоимости проектирования (за счет декомпозиции сложности задачи).



.....

Принимаемые предположения относительно характеристик элементов или системы могут не оправдаться, поэтому часто требуется повторное выполнение проектных процедур предыдущих этапов. Такие повторения обеспечивают последовательное приближение к оптимальному результату. Таким образом, проектирование становится итерационным процессом, т. е. один и тот же этап повторяется несколько раз.

.....



.....

На практике процедуры восходящего и нисходящего проектирования сочетаются и часто выполняются поочередно.

.....



Выводы

.....

Таким образом, *проектирование* – сложный многоэтапный процесс, при котором поочередно выполняются процедуры как нисходящего, так и восходящего проектирования с коррекцией ТЗ к системе и ее элементам на каждой итерации.

.....

1.3 Сущность и этапы проектирования радиоэлектронных устройств

1.3.1 Описания радиоэлектронных средств

В основе функционирования радиоэлектронных средств (РЭС) лежат передача, извлечение и обработка информации, а также преобразование и передача электромагнитной энергии. Электромагнитные поля, волны и сигналы – фундаментальные понятия радиоэлектроники, которыми приходится оперировать при создании и использовании РЭС.

Радиоэлектронное средство, как правило, бывает весьма сложным техническим объектом, включающим большое число составных частей с многообразными связями между ними. *Описание* таких объектов в процессе их создания и использования должно быть согласовано с возможностями их восприятия человеком.



.....
 По характеру отображаемых свойств описания РЭС разделяют на функциональные, конструкторские и технологические.

Функциональное описание отображает основные принципы работы и протекающие в РЭС физические и информационные процессы.

Конструкторское описание отображает материальную реализацию РЭС, его геометрические формы, расположение в пространстве, используемые материалы и компоненты и т. п.

Технологическое описание относится к методам и средствам изготовления РЭС [5].

.....



.....

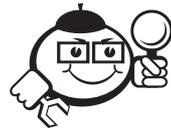
Для упрощения восприятия описаний, их систематизации они разделяются на иерархические уровни. Каждый уровень описания отличается степенью детализации отображаемых свойств РЭС. На всех уровнях описания стремятся представить РЭС в виде совокупности связанных блоков, которые являются обособленными составными частями, причем это, как правило, делается таким образом, чтобы отдельные блоки были самостоятельными материальными объектами. Каждый блок данного иерархического уровня описания представляет собой описание следующего, более низкого иерархи-

ческого уровня, который, в свою очередь, представляется в виде совокупности блоков.



Выводы

Такое разукрупнение (декомпозиция) доводится до уровня, у которого составляющие блоки дальнейшему делению не подлежат.



Пример

По функциональному описанию РЭС уровни разукрупнения включают систему, комплекс, устройство и функциональный узел (рис. 1.5).

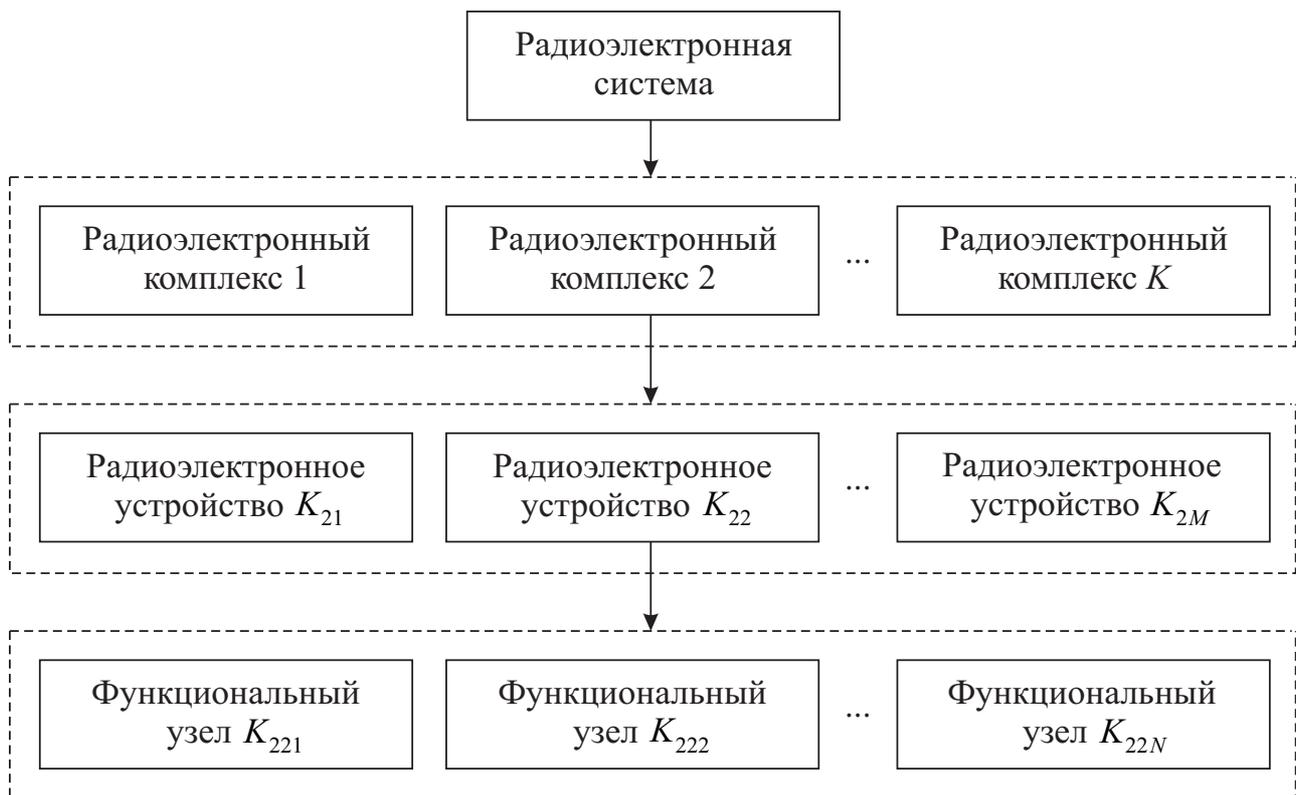


Рис. 1.5 – Иерархическая структура РЭС (функциональный аспект)

Нижний иерархический уровень – радиоэлектронный функциональный узел – представляет собой функционально законченную сборочную единицу, выполненную на несущей конструкции и не имеющую самостоятельного эксплуатационного применения (например, усилитель, модулятор и т. п.). Однако

функциональный узел также может быть представлен в виде составных частей – компонентов, таких как интегральные схемы, транзисторы, резисторы, конденсаторы, печатные платы, подложки и т. п., с определенными связями между ними, т. е. узел может быть описан с различной степенью детализации, так же, как его компоненты.



.....

Конструкторское описание РЭС обычно состоит из следующих уровней: шкаф, блок, модуль, ячейка. Модули также разукрупняются на модули 3, 2, 1-го и нулевого уровня. Все уровни конструкторского описания выполняются на основе своей несущей конструкции, т. е. на совокупности конструктивных элементов, предназначенных для размещения средств, обеспечения их устойчивости и прочности в заданных условиях эксплуатации (например, рама, корпус, кассета, стойка, пульт и т. п.). Радиоэлектронные модули обладают свойствами конструктивной и функциональной взаимозаменяемости.

.....



.....

Окончательное описание радиоэлектронного средства как законченного изделия представляет собой полный комплект технической документации, включающий в себя текстовый материал, схемы, чертежи, рисунки, фотографии, технологические карты и т. п., оформленные в соответствии с требованиями ЕСКД. Этот комплект документации необходим для изготовления и эксплуатации РЭС. Он может быть выполнен как на традиционных бумажных, так и на машинных носителях (магнитных дисках, лентах и др.).

.....

При проектировании также широко используются *промежуточные описания* РЭС и его составных частей. Важнейшее место среди них занимают *модели* различных классов. Например, представления о свойствах РЭС могут быть получены с помощью *натурной* модели (макета, стенда и т. д.) или другой *физической* модели, представляющих собой некий «образ» РЭС. Свойства РЭС на таких моделях изучаются путем эксперимента.

При автоматизированном проектировании для проектирования РЭС доминирующую роль среди промежуточных описаний играют *математические модели*.

1.3.2 Этапы (аспекты) проектирования РЭС



Пример

Рассмотрим более подробно аспекты проектирования РЭС, они имеют свои специфические характеристики и названия. Аспекты проектирования РЭС также часто называют *уровнями* (или *этапами*) проектирования. Их еще именуют вертикальными уровнями, в отличие от горизонтальных иерархических уровней.

Блок-схема процесса проектирования РЭС представлена ниже на рисунке 1.6.



Рис. 1.6 – Этапы проектирования РЭС (блок-схема)

Процесс проектирования РЭС можно разделить на следующие этапы:

1. *Этап системотехнического проектирования.*

Целью проектирования на этом этапе является определение принципов построения и выбор структуры радиоэлектронной системы. При этом формируется структура проектируемой системы, ее составных частей, которыми обычно

являются функционально завершённые блоки, определяются энергетические и информационные связи между составными частями РЭС. В результате формулируются частные ТЗ на проектирование составных частей РЭС. Можно выделить две стадии системотехнического проектирования:

- *Системное проектирование* – на этом этапе проводится анализ ТЗ, принимается решение относительно методов, принципов построения системы в целом.
- *Структурное проектирование* – на этом этапе проводится разработка общей структурной схемы РЭС. Определяются технические требования к выходным параметрам составляющих блоков.

2. *Этап схемотехнического проектирования.* Цель схемотехнического проектирования – разработка принципиальных электрических схем блоков и самой системы в целом. При этом выбирают элементную базу, принципиальные схемы и оптимизируют параметры с точки зрения обеспечения наилучшего функционирования и эффективного производства. При выборе элементной базы и построении схем стремятся учитывать конструкторско-технологические требования.

Задачи, решаемые на этапе схемотехнического проектирования:

- 1) синтез структуры принципиальной схемы блока;
- 2) расчет параметров пассивных элементов, выбор активных элементов;
- 3) анализ принципиальной схемы блока на предмет соответствия требованиям технического задания;
- 4) анализ чувствительности рабочих характеристик блока к разбросу параметров элементов, изменениям внешней среды и пр.

На этапе схемотехнического проектирования часто используют ЭВМ для расчета радиоэлектронных устройств и анализа их рабочих параметров (машинный эксперимент).

3. *Этап конструкторского проектирования.* Данный этап, называемый по-другому *техническим проектированием*, решает задачи компоновки и размещения элементов и узлов, осуществления печатных и проводных соединений для РЭС всех уровней (модулей, ячеек, блоков, шкафов и т. д.), а также задачи теплоотвода, электрической прочности, защиты от внешних воздействий и т. п. При этом стремятся оптимизировать принимаемые решения по конструктивно-технологическим, экономическим и эксплуатационным показателям.



Задачи, решаемые на этапе конструкторского проектирования:

- 1) конструкторско-монтажное проектирование (размещение элементов, трассировка проводников на плате);
- 2) обеспечение тепловых режимов и электромагнитной совместимости;
- 3) конструирование электромеханических узлов внешних устройств, расчеты на прочность;
- 4) изготовление конструкторской документации.

Компьютер на этапе конструкторского проектирования используется как для задач расчетного характера (размещение элементов, разводка печатных плат, расчет тепловых и электромагнитных полей и т. д.), так и для оформления конструкторской документации.

4. Этап технологического проектирования.

На данном этапе осуществляется проектирование технологического процесса изготовления узлов и блоков РЭС, создается технологическая документация на основе предшествующих результатов.

Задачи, решаемые на этапе технологического проектирования:

- 1) разработка схемы технологического процесса;
- 2) разработка маршрутов обработки деталей;
- 3) разработка технологических операций;
- 4) разработка необходимой оснастки и оборудования [5].



В современных «интеллектуальных» производственных линиях ЭВМ управляет всеми технологическими процессами и следит за качеством изготовления устройств. Таким образом, возникает задача разработки алгоритмов и программ для управления подобными автоматизированными линиями. Кроме того, на этапе конструкторского проектирования ЭВМ также используется для оформления технологических документов.



При проектировании новых компонентов задачи функционального, конструкторского и технологического проектирования

взаимосвязаны. Например, при разработке новой интегральной схемы необходимо решать следующие задачи:

- выбор структуры схемы, расчет параметров элементов;
 - выбор топологии, расчет геометрических размеров;
 - расчет электрических характеристик;
 - расчет чувствительности и оценка выхода годных изделий при изготовлении;
 - расчет технологических процессов.
-



Контрольные вопросы по главе 1

.....

1. Опишите концепцию блочно-иерархического проектирования (БИП).
2. Расскажите о горизонтальных и вертикальных уровнях проектирования.
3. Что понимается под математической моделью?
4. Назовите основные направления проектирования ЭС.
5. Назовите базовые элементы РЭС первого уровня.
6. Раскройте принципы проектирования.
7. Раскройте принцип иерархичности.
8. Раскройте принцип декомпозиции.
9. Опишите схему уровней сложности системы ЭВС.
10. Назовите основные принципы проектирования сложных технических объектов.

2 Математическое моделирование технических объектов



Математическое обеспечение автоматизированного проектирования включает в себя математические модели объектов проектирования и алгоритмы выполнения проектных процедур.

2.1 Общие сведения о математических моделях



Проектирование представляет собой процесс преобразования описаний проектируемого объекта, начиная от исходного описания – технического задания – и заканчивая окончательным описанием – комплектом конструкторской документации, предназначенным для использования в процессе изготовления и эксплуатации данного объекта. В процессе проектирования возникают специфические промежуточные решения – проектные решения. В качестве промежуточных описаний в процессе проектирования часто выступают **математические модели объекта (ММ)**.

Математическая модель технического объекта – это система математических объектов (чисел, переменных, матриц, множеств, таблиц и др.) и отношений между ними, отражающая некоторые свойства технического объекта.



Свойств у технических объектов – бесконечное число, поэтому в ММ учитываются только свойства, существенные для решаемой задачи с точки зрения разработчика.



Свойства технических объектов в ММ количественно характеризуются некоторыми числовыми величинами, которые называют **параметрами** (или **характеристиками**). В общей теории математического моделирования ММ любого объекта характеризуют *внут-*

ренными, внешними, выходными параметрами и фазовыми переменными.

.....

Внутренние параметры модели определяются характеристиками компонентов, входящих в состав проектируемого объекта, например номиналы элементов принципиальной схемы и т. д.

Выходные параметры модели характеризуют внешние свойства объекта (его рабочие характеристики), которые зависят от параметров элементов, составляющих объект (*внутренних параметров*), и параметров окружающей среды (*внешних параметров*).

Понятия внутренних и выходных параметров инвариантны, при моделировании на более сложном уровне выходные параметры могут стать внутренними и наоборот. Например, сопротивление резистора является внутренним параметром при моделировании усилительного устройства, компонентом которого он является. Но это же сопротивление будет выходным параметром при моделировании самого резистора, что требуется при его пленочном исполнении.

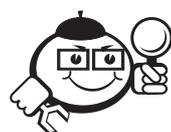


.....

Внешние параметры модели – это характеристики внешней по отношению к проектируемому объекту среды, а также рабочие управляющие воздействия. Вектор внешних параметров в общем случае содержит множество самых различных составляющих. В качестве внешних параметров ММ выступают характеристики внешней среды и воздействия, которые могут быть выражены **количественно**.

Внутренние параметры мы можем менять (хотя бы частично), т. е. геометрические размеры, способ соединения элементов, электрическую схему и т. д. **Внешние параметры** не зависят от разработчика (температура, давление, влажность и пр.) и обычно являются дополнительными требованиями в ТЗ к функционированию объекта [1, 2, 3].

.....



.....

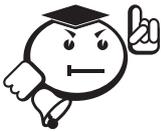
Пример

Для ММ электронного усилителя:

- 1) *выходные параметры* – полоса пропускания, коэффициент усиления, входное сопротивление, потребляемая мощность и т. д.;
- 2) *внутренние параметры* – сопротивления резисторов, емкости конденсаторов, параметры активных элементов (транзисторов) и т. д. и способ соединения этих элементов между собой (т. е. принципиальная схема);
- 3) *внешние параметры* – сопротивления генератора и нагрузки, температура окружающей среды, управляющие сигналы и т. д.

Пусть для некоторой ММ есть m внутренних параметров, l внешних параметров и n выходных параметров. Векторы этих параметров обозначим соответственно $\vec{X} = (x_1, x_2, \dots, x_n)$ – внутренние параметры; $\vec{Q} = (q_1, q_2, \dots, q_l)$ – внешние параметры; $\vec{Y} = (y_1, y_2, \dots, y_m)$ – выходные параметры. Тогда ММ объекта можно записать в общем виде:

$$\vec{Y} = F(\vec{X}, \vec{Q}). \quad (2.1)$$



Наличие такой ММ позволяет легко оценивать выходные параметры по известным значениям векторов \vec{X} и \vec{Q} . Однако существование зависимости (2.1) не означает, что она известна разработчикам и может быть представлена в явном аналитическом виде относительно вектора \vec{Y} . Как правило, ММ в виде (2.1) удастся получить только для достаточно простых объектов. Типичной является ситуация, когда ММ сложных объектов задаются в *алгоритмическом виде*, т. е. известен алгоритм (последовательность) расчета вектора \vec{Y} по заданным внутренним параметрам \vec{X} и внешним параметрам \vec{Q} .



К выходным параметрам (рабочим характеристикам) технических объектов предъявляются определенные требования, которые формулируются в ТЗ на проектирование. Обычно их записывают в виде одно- или двусторонних неравенств:

$$TT_i^- \leq y_i \leq \overline{TT_i^+} \quad (i = \overline{1, m}), \quad (2.2)$$

здесь TT_i^-, TT_i^+ – нижние и верхние допустимые значения величины y_i . Совокупность технических требований TT_i^-, TT_i^+ ($i = \overline{1, m}$), предъявляемых к вектору \vec{Y} , образуют векторы технических требований

$$\vec{TT}^- = (TT_1^-, TT_2^-, \dots, TT_m^-) \text{ и } \vec{TT}^+ = (TT_1^+, TT_2^+, \dots, TT_m^+).$$

Требуемые соотношения между параметрами \vec{Y} и техническими требованиями \vec{TT}^- и \vec{TT}^+ называют **условиями работоспособности**.

.....

Используя соотношение (2.2), можно записать общий вид требований ТЗ к выходным параметрам ММ:

$$\vec{TT}^- \leq \vec{Y} \leq \vec{TT}^+. \quad (2.3)$$

Уравнения математической модели могут связывать некоторые физические характеристики компонентов, которые полностью характеризуют состояние объекта, но не являются выходными или внутренними параметрами модели (например, токи и напряжения в радиоэлектронных устройствах, внутренними параметрами которых являются номиналы элементов электрических схем, а выходными параметрами – выходная мощность, коэффициент передачи и т. п.). Такие характеристики называют *фазовыми переменными*.

Минимальный по размерности вектор фазовых переменных $\vec{V} = (v_1, v_2, \dots, v_s)$, полностью характеризующий работу объекта проектирования, называют *базисным вектором*. Например, при составлении уравнений ММ радиоэлектронных устройств в качестве базисного вектора \vec{V} можно использовать вектор узловых потенциалов либо вектор напряжений на конденсаторах и токов в индуктивностях – так называемые *переменные состояния*. Использование вектора фазовых переменных позволяет упростить алгоритмическую реализацию программ, составляющих уравнения математической модели устройства.

В общем случае выходные параметры \vec{Y} представляются операторами от векторов \vec{X} , \vec{Q} , \vec{V} и могут быть определены из решения системы уравнений

ММ устройства. С учетом вышесказанного, ММ любого технического объекта может быть представлена в виде следующих уравнений:

$$\varphi(\vec{V}, \vec{X}, \vec{Q}) = 0; \quad (2.4)$$

$$\vec{Y} = \psi(\vec{V}), \quad (2.5)$$

где φ и ψ – операторы, определяющие вид системы уравнений ММ.

В результате решения системы (2.4) определяются действующие в устройстве фазовые переменные \vec{V} . Система уравнений (2.5) определяет зависимость выходных параметров объекта \vec{Y} от фазовых переменных \vec{V} . В частных случаях составляющие вектора \vec{V} могут являться внутренними или выходными параметрами ММ, и тогда системы уравнений (2.4) и (2.5) упрощаются.



.....

Отметим, что часто *моделированием* называют лишь получение системы (2.4). Решение уравнений (2.4) и отыскание вектора \vec{Y} с помощью уравнений (2.5) называют *анализом* математической модели.

.....

2.2 Требования к математическим моделям

Математические модели служат для описания свойств объектов в процедурах автоматизированного проектирования.



.....

Если проектная процедура включает в себя создание ММ и оперирование ею с целью получения полезной информации об объекте, то говорят, что процедура выполняется на основе *математического моделирования* или *вычислительного эксперимента*.

К математическим моделям объектов проектирования предъявляются следующие требования:

- универсальность;
 - адекватность;
 - точность;
 - экономичность.
-



.....

Степень *универсальности* ММ характеризует полноту отображения в модели свойств реального объекта. Математическая модель отражает только часть свойств (необходимых для решения задачи проектирования). Так, большинство ММ, используемых при функциональном проектировании, предназначены для отображения протекающих в объекте физических или информационных процессов, при этом не требуется, чтобы ММ описывала такие свойства объекта, как геометрическая форма и вес составляющих его элементов и пр.

.....

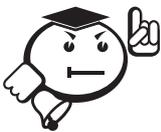


Пример

.....

ММ резистора в виде уравнения закона Ома характеризует свойства резистора пропускать электрический ток, но не отражает габариты резистора, его цвет, стоимость, механическую прочность и т. д.

.....



.....

Естественно, чем больше свойств объекта отображено в ММ, тем более универсальной получается модель. Однако более универсальная модель получается и более сложной. На ее реализацию необходимо затратить больше вычислительных ресурсов компьютера (памяти, времени на расчет).

.....

Универсальность ММ – отличительная черта ММ, используемых в автоматизированном проектировании, в отличие от традиционного «ручного» проектирования, где применяются упрощенные модели. Универсальность математических моделей – это свойство, позволяющее применять модели при описании или функционировании однотипных групп объектов.

Точность ММ оценивается степенью совпадения значений параметров реального объекта и значений этих же параметров, рассчитанных с помощью ММ. Пусть отражаемые в ММ свойства оцениваются вектором выходных параметров $\vec{Y} = (y_1, y_2, \dots, y_m)$. Тогда, обозначив истинное и полученное с помощью ММ

значения j -го выходного параметра через $y_{j \text{ ист}}$ и $y_{j \text{ ММ}}$ соответственно, определим относительную погрешность расчета параметра y_j как

$$\varepsilon_j = \left| y_{j \text{ ист}} - y_{j \text{ ММ}} \right| / y_{j \text{ ист}}, \quad j = \overline{1, m}.$$

Для случая m выходных параметров получим вектор ошибок $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m)$. При необходимости отражения точности в виде одного числа используют какую-либо норму вектора ε . Чаще всего используют единичную (чебышевскую) норму:

$$\varepsilon_1 = \|\varepsilon\|_1 = \max_{j=1, m} \varepsilon_j \quad (2.6)$$

или среднеквадратичную погрешность:

$$\varepsilon_2 = \|\varepsilon\|_2 = \sqrt{\sum_{j=1}^m \varepsilon_j^2}. \quad (2.7)$$



.....

Точность ММ зависит от условий функционирования объекта, т. е. от конкретных значений векторов внешних воздействий \vec{Q} и внутренних параметров \vec{X} . Для разных значений \vec{X} и \vec{Q} получаем разную точность расчета выходных характеристик \vec{Y} . Таким образом, когда мы говорим о точности ММ, необходимо также называть диапазоны изменения значений векторов \vec{X} и \vec{Q} , при которых обеспечивается указываемая точность ММ.

Адекватность ММ – способность отображать свойства реального объекта с заданной точностью в определенном интервале значений внутренних параметров \vec{X} и параметров среды \vec{Q} . Обычно значения внутренних параметров ММ определяют из условия минимизации погрешности ε_1 (или ε_2) в некоторой точке \vec{Q}_0 пространства внешних переменных, а используют ММ с рассчитанным вектором \vec{X} уже при разных значениях \vec{Q} [1, 4].

.....

При этом требуемая точность сохраняется в определенном интервале изменения внешних переменных \vec{Q} – так называемой *области адекватности ММ* (ОА):

$$OA = \{\vec{Q} \mid \varepsilon_1 \leq \delta\},$$

где $\delta > 0$ – заданная константа, равная предельно допустимой погрешности ММ.

2.3 Классификация математических моделей

Основные признаки классификации и типы ММ, применяемых в автоматизированном проектировании, даны в таблице 2.1 [2, 3].

Таблица 2.1 – Классификация математических моделей технических объектов

Признак классификации	ММ
Характер отображаемых свойств	Структурные, функциональные технологические
Принадлежность к иерархическому уровню	Микроуровня, макроуровня, метаяуровня
Способ получения ММ	Теоретические, эмпирические (экспериментальные)
Способ представления свойств объекта	Аналитические, алгоритмические, имитационные
По уровню сложности (детализации)	Полные ММ, макромодели
Используемый в ММ математический аппарат	Линейные, нелинейные, непрерывные, дискретные, статические, динамические и т. д.



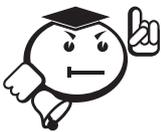
По характеру отображаемых свойств ММ делятся на структурные, функциональные и топологические.

Функциональные математические модели предназначены для отображения физических или информационных процессов, протекающих в технологических системах при их функционировании. Типичная функциональная модель представляет собой систему уравнений, описывающих либо электрические, тепловые, механические процессы, либо процессы преобразования информации. Внутренняя структура системы в таких моделях не отображается и не исследуется. Примером функциональной математической модели является модель «чёрного ящика».

Структурные ММ предназначены для отображения структурных свойств объекта. Структурные ММ, в свою очередь, подразделяются на **топологические** и **геометрические** ММ.

В **топологических** ММ отображаются состав и взаимосвязи элементов объекта. Их чаще всего применяют для описания объектов, состоящих из большого числа элементов, при решении задач привязки конструктивных элементов к определенным пространственным позициям (задачи компоновки оборудования, размещения деталей, трассировки соединений и т. п.) или к относительным моментам времени (например, при разработке расписаний, технологических процессов и т. д.). Топологические ММ могут иметь форму графов, таблиц, матриц, списков и т. п.

В **геометрических** ММ отображаются геометрические свойства объектов, в них дополнительно к сведениям о взаимном расположении элементов содержатся сведения о форме деталей. Геометрические ММ могут выражаться совокупностью уравнений линий и поверхностей; алгебраических соотношений, описывающих области, составляющие тело объекта; графами и списками, отображающими конструкции из типовых конструктивных элементов, и т. п. Геометрические ММ применяют при решении задач конструирования в машиностроении, приборостроении, радиоэлектронике, для оформления конструкторской документации, при задании исходных данных на разработку технологических процессов изготовления деталей. Используют несколько типов геометрических ММ [1].



В машиностроении для отображения геометрических свойств деталей со сравнительно несложными поверхностями применяют ММ, представляемые в аналитической или алгебрологической форме (аналитические, алгебрологические). Аналитические ММ – уравнения поверхностей и линий, например: уравнение плоскости имеет вид $ax + by + cz + d = 0$, а эллипсоида – вид $(x/a)^2 + (y/b)^2 + (z/c)^2 + d = 0$, где x, y, z – пространственные координаты.

ты; a, b, c, d – коэффициенты уравнений. В алгебрологических ММ тела описываются системами логических выражений, отражающих условия принадлежности точек внутренним областям тел.

Для сложных поверхностей аналитические и алгебрологические модели оказываются слишком громоздкими, их трудно получать и неудобно использовать. Область их применения обычно ограничивается поверхностями плоскими и второго порядка.

.....

В машиностроении для отображения геометрических свойств деталей со сложными поверхностями применяют ММ каркасные и кинематические.

.....



Каркасные ММ представляют собой каркасы – конечные множества элементов, например точек или кривых, принадлежащих моделируемой поверхности. В частности, выбор каркаса в виде линий, образующих сетку на описываемой поверхности, приводит к разбиению поверхности на отдельные участки. Кусочно-линейная аппроксимация на этой сетке устраняет главный недостаток аналитических моделей, так как в пределах каждого из участков, имеющих малые размеры, возможна удовлетворительная по точности аппроксимация поверхностями с простыми уравнениями. Коэффициенты этих уравнений рассчитываются исходя из условий плавности сопряжения участков.

В *кинематических* ММ поверхность представляется в параметрическом виде $R(u, v)$, где $R = (x, y, z)$, а u и v – параметры. Такую поверхность можно получить как результат перемещения в трехмерном пространстве кривой $R(u)$, называемой образующей, по некоторой направляющей линии [1, 2].

.....

Коэффициенты уравнений во всех рассмотренных моделях, как правило, не имеют простого геометрического смысла, что затрудняет работу с ними в интерактивном режиме. Этот недостаток устраняется в канонических моделях и в геометрических макромоделях.

.....



Канонические модели используют в тех случаях, когда удастся выделить параметры, однозначно определяющие геометрический

объект и в то же время имеющие простую связь с его формой. Например, для плоского многоугольника такими параметрами являются координаты вершин, для цилиндра – направляющие косинусы и координаты некоторой точки оси, а также радиус цилиндра.

Геометрические макромодели являются описаниями предварительно отобранных типовых геометрических фрагментов. Такими фрагментами могут быть типовые сборочные единицы, а их макромоделями – условные номера, габаритные и стыковочные размеры. При оформлении конструкторской документации макромодели используют для описания типовых графических изображений, например зубчатых колес, винтовых соединений, подшипников и т. п.

Технологические ММ основываются на параметрах технологических процессов изготовления проектируемого объекта. Выходные параметры таких моделей – совокупность физико-топологических или технологических параметров.

Функциональные ММ предназначены для отображения физических или информационных процессов, протекающих в объекте при его функционировании или изготовлении. Обычно функциональные ММ представляют собой системы уравнений, связывающих фазовые переменные, внутренние, внешние и выходные параметры [3, 4].



.....

.....

Деление описаний объектов на аспекты и иерархические уровни непосредственно касается математических моделей. Выделение аспектов описания приводит к выделению моделей электрических, механических, гидравлических, оптических, химических и т. п., причем модели процессов функционирования изделий и модели процессов их изготовления различны, например: модели полупроводниковых элементов интегральных схем, описывающих процессы диффузии и дрейфа подвижных носителей заряда в полупроводниковых областях при функционировании прибора и процессы диффузии примесей в полупроводник при изготовлении прибора [1].

.....



Выводы

Использование принципов блочно-иерархического подхода к проектированию приводит к появлению иерархии математических моделей проектируемых объектов. Количество иерархических уровней при моделировании определяется сложностью проектируемых объектов и возможностями средств проектирования. Однако для большинства предметных областей можно отнести имеющиеся иерархические уровни к одному из трех обобщенных уровней, называемых далее микро-, макро- и метауровнями. Исходя из этого математические модели делятся на ММ, относящиеся к микро-, макро- и метауровням.



Особенностью ММ на *микроуровне* является отражение физических процессов, протекающих в непрерывных пространстве и времени. Типичные ММ на микроуровне – дифференциальные уравнения в частных производных (ДУЧП). В них независимыми переменными являются пространственные координаты и время. С помощью этих уравнений рассчитываются поля механических напряжений и деформаций, электрических потенциалов, давлений, температур и т. п.



Пример

В радиотехнике с помощью ММ *микроуровня* моделируют компоненты СВЧ-устройств (полосковые устройства, волноводы, печатные индуктивности и пр.), а также полупроводниковые приборы и интегральные микросхемы при их проектировании.

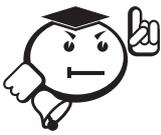


На *макроуровне* используют укрупнённую дискретизацию пространства по функциональному признаку, что приводит к представлению ММ на этом уровне в виде систем обыкновенных дифференциальных уравнений (ОДУ). В этих уравнениях независимой переменной является время t , а вектор зависимых переменных \vec{V} составляют фазовые переменные, характеризующие состояние укрупнённых элементов дискретизированного пространства. Такими пере-

менными являются силы и скорости механических систем, напряжения и силы тока электрических систем, давления и расходы гидравлических и пневматических систем и т. п. Системы ОДУ являются универсальными моделями на макроуровне, пригодными для анализа как динамических, так и установившихся состояний объектов. Модели для установившегося режима также можно представить в виде системы алгебраических уравнений. Порядок системы уравнений зависит от числа выделенных элементов проектируемого объекта.

.....

.....



ММ сложного устройства на макроуровне образуется путём объединения ММ составляющих его элементов. Для описания структуры сложного объекта к *компонентным уравнениям*, описывающим свойства элементов, добавляются *топологические уравнения*, отражающие связи элементов между собой. ММ элементов на макроуровне не отражают пространственную структуру. ММ элементов на макроуровне не отражают пространственную структуру. Они только связывают фазовые элементы с характеристиками самих элементов и временем t . В результате в ММ макроуровня отсутствуют пространственные координаты, а непрерывной независимой переменной остается только время. При моделировании РЭС в качестве фазовых переменных выступают напряжения и токи на элементах. В качестве компонентных уравнений используется закон Ома, а в качестве топологических – законы Кирхгофа.

.....

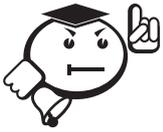
.....



Метауровень характеризуется большим разнообразием типов используемых ММ. Для многих объектов ММ на метауровне по-прежнему представляются системами ОДУ. Однако так как в моделях не описываются внутренние для элементов фазовые переменные, а фигурируют только фазовые переменные, относящиеся к взаимным связям элементов, то укрупнение элементов на метауровне означает получение ММ приемлемой размерности для существенно более сложных объектов, чем на макроуровне.

.....

.....



В ряде предметных областей удастся использовать специфические особенности функционирования объектов для упрощения ММ. Примером являются электронные устройства цифровой автоматики, в которых возможно применять дискретное представление таких фазовых переменных, как напряжения и токи. В результате ММ становится системой логических уравнений, описывающих процессы преобразования сигналов. Такие логические модели существенно более экономичны, чем модели электрические, описывающие изменения напряжений и сил токов как непрерывных функций времени. Важный класс ММ на метауровне составляют модели массового обслуживания, применяемые для описания процессов функционирования информационных и вычислительных систем, производственных участков, линий и цехов.

Структурные модели также делятся на модели различных иерархических уровней. При этом на низших иерархических уровнях преобладает использование геометрических моделей, а на высших уровнях чаще используются топологические ММ.



По степени детализации описания в пределах каждого иерархического уровня выделяют *полные ММ* и *макромодели*.



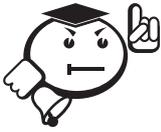
Полная ММ – модель, в которой фигурируют фазовые переменные, характеризующие состояния всех имеющихся межэлементных связей (т. е. состояния всех элементов проектируемого объекта). Полные ММ получают путем объединения всех ММ составляющих элементов в общую систему.

Макромодель – ММ, в которой отображаются состояния значительно меньшего числа межэлементных связей, что соответствует описанию объекта при укрупненном выделении элементов. Обычно макромодель представляет собой некоторую аппроксимацию полной ММ, выходные характеристики которой близки к характеристикам исходной модели. Применение макромоделей вместо

полных ММ обусловлено необходимостью сокращения временных затрат на моделирование достаточно сложных РЭУ [4].

.....

.....



Понятия «полная ММ» и «макромодель» относительно и обычно используются для различения двух моделей, отображающих различную степень детальности описания свойств объекта.

.....

По способу представления свойств объекта функциональные ММ делятся на *аналитические* и *алгоритмические*.

.....



Аналитические ММ представляют собой явные выражения выходных параметров как функций входных и внутренних параметров, т. е. имеют вид $\vec{Y} = \varphi(\vec{X}, \vec{Q})$. Такие ММ характеризуются высокой экономичностью, однако получение явной аналитической формы ММ удастся лишь в отдельных частных случаях, как правило, при принятии существенных допущений и ограничений, снижающих точность и сужающих область адекватности модели.

Алгоритмические ММ выражают связи выходных параметров объекта с параметрами внутренними и внешними в форме алгоритма, т. е. известна последовательность шагов для определения выходных характеристик объекта \vec{Y} по известным значениям векторов \vec{X} и \vec{Q} .

Имитационная ММ – алгоритмическая модель, отражающая поведение исследуемого объекта во времени при задании внешних воздействий на объект. Примерами имитационных ММ могут служить модели динамических объектов в виде систем ОДУ или модели систем массового обслуживания, заданные в алгоритмической форме [4, 5].

.....

Для получения ММ используют методы *неформальные* и *формальные*.

.....



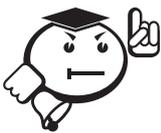
Неформальные методы применяют на различных иерархических уровнях для получения ММ элементов. Эти методы включают изучение закономерностей процессов и явлений, связанных с моделируемым объектом, выделение существенных факторов, принятие

различного рода допущений и их обоснование, математическую интерпретацию имеющихся сведений и т. п. Для выполнения этих операций в общем случае отсутствуют формальные методы, в то же время от результата этих операций существенно зависят показатели эффективности ММ – степень универсальности, точность, экономичность. Поэтому построение ММ элементов, как правило, осуществляется квалифицированными специалистами, получившими подготовку как в соответствующей предметной области, так и в вопросах математического моделирования на ЭВМ.



Пример

Применение неформальных методов возможно для синтеза ММ теоретических и эмпирических (или, по-другому, экспериментальных). *Теоретические* ММ создаются в результате исследования процессов и их закономерностей, присущих рассматриваемому классу объектов и явлений, а *эмпирические* ММ – в результате изучения внешних проявлений свойств объекта с помощью измерений фазовых переменных на внешних входах и выходах и обработки результатов измерений. Эмпирические ММ еще часто называют моделями «черного ящика».



Решение задач моделирования элементов облегчается благодаря тому, что для построения большинства технических объектов используются типовые элементы (количество типов сравнительно невелико). Поэтому разработка ММ элементов производится сравнительно редко. Единожды созданные ММ элементов в дальнейшем многократно применяют при разработке разнообразных систем из этих элементов.



Пример

Примерами таких ММ на микроуровне служат описания конечных элементов для анализа напряженно-деформированного состояния деталей, множество типов конечных элементов включает стержни, плоские элементы в форме

треугольников и четырехугольников, трехмерные элементы типа параллелепипеда, тетраэдра и т. п. Примерами ММ геометрических элементов могут служить уравнения линий прямых, дуг окружностей, плоскостей и поверхностей второго порядка. Примерами ММ элементов на макроуровне являются модели элементов интегральных схем – транзисторов, диодов, резисторов, конденсаторов и т. д.



Формальные методы применяют для получения ММ систем при известных математических моделях элементов.



Выводы

Таким образом, в программах автоматизированного анализа, используемых в САПР, получение ММ проектируемых объектов обеспечивается реализацией моделей элементов и методов формирования ММ системы в целом.

При проектировании РЭС наиболее широкое применение получили электрические модели.

2.4 Методика получения математических моделей

В общем случае процедура получения математических моделей элементов включает в себя следующие операции:

1. *Выбор свойств объекта*, которые подлежат отражению в модели. Этот выбор основан на анализе возможных применений модели и определяет степень универсальности ММ.

2. *Сбор исходной информации* о выбранных свойствах объекта. Источниками сведений могут быть опыт и знания инженера, разрабатывающего модель, научно-техническая литература, прежде всего справочная, описания прототипов – имеющих ММ для элементов, близких по своим свойствам к исследуемому, результаты экспериментального измерения параметров и т. п.

3. *Синтез структуры ММ*. Структура ММ – общий вид математических соотношений модели без конкретизации числовых значений фигурирующих в них параметров. Структура модели может быть представлена также в графической форме, например в виде эквивалентной схемы или графа. Синтез структуры – наиболее ответственная и с наибольшим трудом поддающаяся формализации

операция. При синтезе структура ММ определяется либо из анализа экспериментальных зависимостей выходных характеристик объекта, либо на основе теоретического изучения свойств объекта.

4. *Расчет числовых значений параметров ММ.* Эта задача ставится как задача минимизации погрешности модели заданной структуры, т. е.

$$\min_{x \in \text{ХД}} \varepsilon_{\text{ММ}}(\vec{X}), \quad (2.8)$$

где \vec{X} – вектор параметров модели; ХД – область варьирования (область допустимых значений) параметров \vec{X} ; $\varepsilon_{\text{ММ}}$ – погрешность ММ, которая вычисляется в соответствии с (2.6) и (2.7), где $y_{j\text{ММ}}$ – функция от \vec{X} , а $y_{j\text{ист}}$ определяется по результатам экспериментов либо физических, либо численных (с использованием более точных ММ, если таковые имеются в иерархическом ряду ММ).

5. *Оценка точности и адекватности ММ.* Для оценки *точности* должны использоваться значения $y_{j\text{ист}}$, которые не фигурировали при решении задачи (2.8) [4].



.....

Большую ценность для пользователя представляют не оценки погрешности $\varepsilon_{\text{ММ}}$, выполненные в одной-двух случайных точках пространства внешних переменных, а сведения об области адекватности (ОА). Однако определение ОА требует больших затрат машинного времени. Поэтому расчет ОА выполняется только при тщательной отработке ММ унифицированных элементов, предназначенных для многократного применения.

.....

2.5 Математические модели

для задач конструирования электронных средств



.....

При решении задач в АСКП (автоматизированные системы конструкторского проектирования) возникает необходимость выбора совокупности математических моделей, обеспечивающих необходимой информацией каждый алгоритм решения конкретной задачи.

.....

Для каждой задачи АСКП (компоновка, размещение, трассировка) на этапе, предшествующем постановке задачи, выбирается (разрабатывается) математическая модель. Это позволяет перейти от содержательной постановки задачи к формальной постановке в терминах выбранной модели. Поскольку цели отдельных задач АСКП отличаются, то возможно для решения каждой задачи применять свою модель.

Проблема выбора (построения) моделей является важной и актуальной, так как от модели во многом зависит качество решения задач в АСКП и эффективность алгоритмических методов.



Пример

Требования, предъявляемые к математической модели:

- 1) простота перехода от объекта к модели и обратно;
- 2) удобство представления и обработки модели в ЭВМ;
- 3) небольшой объем памяти компьютера, отводимый для хранения информации для модели;
- 4) разработанность математического аппарата для работы с данной моделью (дает возможность разрабатывать высокоэффективные алгоритмы проектирования);
- 5) информационная простота модели (возможность перехода от описания данной модели к более простой) [4].

Математические модели, используемые при решении задач в АСКП, можно разделить на две группы:

- математические модели схем;
- математические модели монтажного пространства.

Рассмотрим математические модели, построенные в рамках аппарата теории графов.

Введем понятия:

- 1) коммутационная схема;
- 2) электрическая цепь;
- 3) контакты.

Коммутационная схема



Любая схема (функциональная, логическая, принципиальная, структурная) состоит из набора элементов и соединений, о кото-

рых можно сказать, что они находятся в заданном отношении. Поэтому такие схемы можно рассматривать как некоторое множество элементов $X = (X_1, X_2, \dots, X_n)$, $|X| = n$ и некоторое множество соединителей (называемых цепями) $E = (e_1, e_2, \dots, e_m)$ $|E| = m$, находящихся в заданном бинарном отношении.

.....

В схеме имеются внешние и внутренние выводы: C .

$$C = \{c_{0,1}, c_{0,2}, \dots, c_{0,p}, c_{1,1}, c_{1,2}, \dots, c_{n,1}, c_{n,2}, \dots, c_{n,m}\}.$$

Внутренние выводы соединяют элементы схемы между собой.

Внешние выводы соединяют элементы данной схемы с элементами другой схемы.

Множество C будем называть множеством контактов.

Два контакта $C_{i,j}$ и $C_{k,e}$ считаются связанными (соединенными), если они объединяются одной электрической цепью.

.....



Электрическая цепь – это некоторое подмножество контактов C' множества контактов C , т. е. $C' \subseteq C$, принадлежащих к одному эквипотенциалу.

.....

Контакты подмножества C' имеют равный потенциал в принципиальной электрической схеме.

Такое представление схемы называют схемой соединений или коммутационной схемой.

При построении математической модели в коммутационной схеме каждой однопотенциальной цепи ставят в соответствие число из натурального ряда.

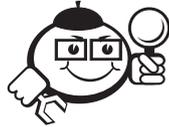
Для решения задач компоновки, размещения и трассировки коммутационную схему можно представлять структурными математическими моделями различного вида.

Математические модели схем РЭС для решения задач конструкторского проектирования разделяют на три вида:

- 1) модели, использующие аппарат теории графов;
- 2) модели, использующие аппарат теории гиперграфов;
- 3) модели, использующие аппарат теории множеств [1, 4].



В настоящее время наиболее широко используются структурные математические модели, построенные на основе аппарата теории графов и в первую очередь для решения оптимизационных задач конструкторского проектирования.



Пример

Рассмотрим структурные математические модели, использующие аппарат теории графов на примере условного фрагмента схемы электрической принципиальной, представленной на рисунке 2.1.

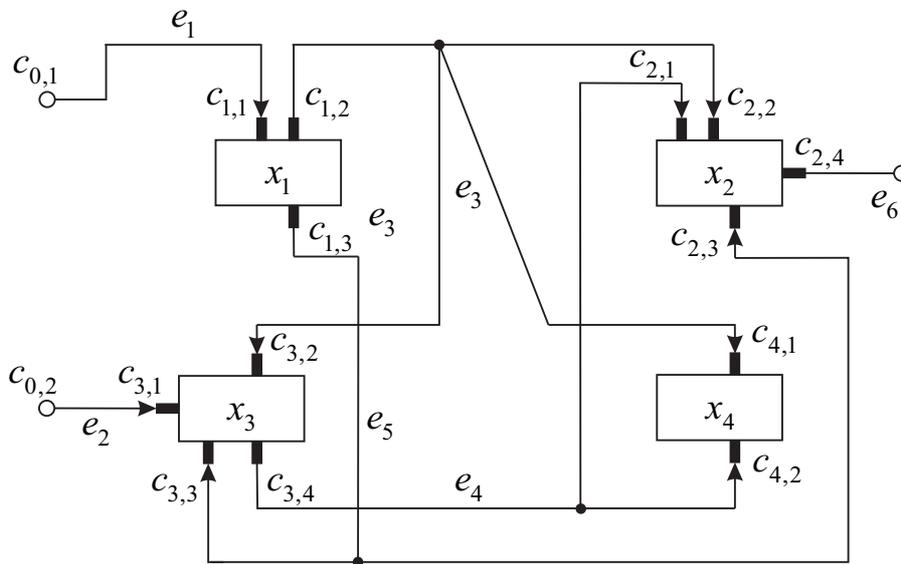


Рис. 2.1 – Фрагмент коммутационной схемы

Одним из способов представления схемы устройства структурной математической моделью является ее интерпретация неориентированным связным графом $G = (X, U)$, где $X = \{x_i\}$ $i = \overline{1, n}$ множество вершин G , U – множество ребер $U = \{u_{ij}\}$, определяющих отношения между элементами множества X .

Для построения графа G необходимо задать между элементами множества $X = \{x_i\}$ и элементами схемы (к элементам схемы относятся сами приборы и внешние контакты схемы) взаимоднозначное соответствие, в том числе

между элементами множества ребер $U = \{u_{ij}\}$ и элементами множества электрических цепей $E = \{e_i\}$.



Пример

Для примера (рис. 2.2) математическая модель M1, интерпретирующая схему (рис. 2.1), представляет неориентированный граф $G_1 = (X, U)$ с мультичислом, равным 3.

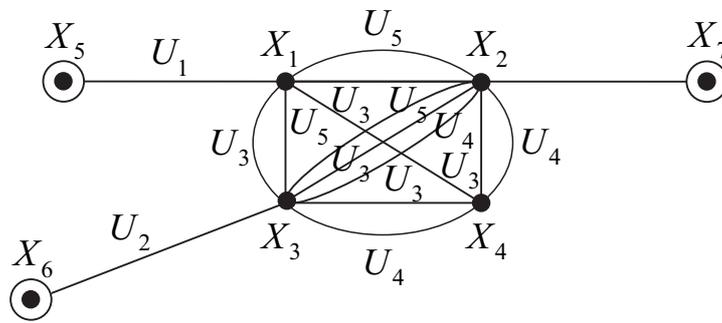


Рис. 2.2 – Математическая модель M1

Модель M1 удобна для решения задач компоновки и размещения.

Для решения задач определения планарности и трассировки математическая модель M1 неудобна из-за больших искажений между моделью и коммутационной схемой.

Модификацией математической модели M1 является математическая модель M2, в которой полные подграфы, моделирующие цепи $e_i \in E$, заменяются покрывающими их деревьями (рис. 2.3).

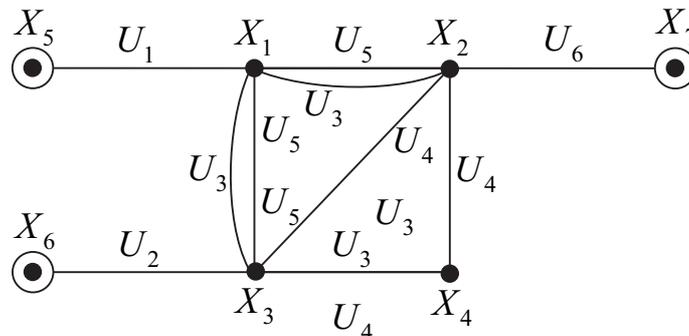


Рис. 2.3 – Математическая модель M2

Разновидностью математической модели М2 является представление коммутационной схемы графом $G = (X, U)$, в котором все покрывающие деревья являются звездными подграфами с центральной вершиной x_i , соответствующей началу цепи e_i (рис. 2.4).

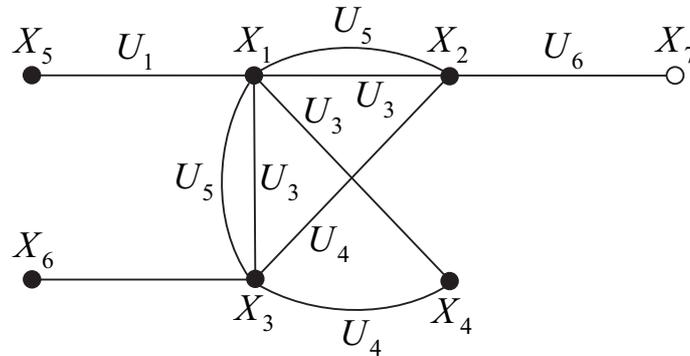


Рис. 2.4 – Математическая модель М3

Общий недостаток приведенных математических моделей: *фиксированные деревья могут ухудшить качество компоновки*. Тем не менее приведенные модели М1, М2, М3 наиболее просты при решении практических задач.

Коммутационная схема также может быть представлена в виде двудольного ориентированного графа $D = (X, C, E, U)$, где X – множество вершин, интерпретирующих элементы схемы, C – множество контактов (входов и выходов, схемы), E – множество вершин, интерпретирующих эквипотенциальные электрические цепи, связывающие группы элементов схемы, U – множество рёбер графа D .

В этом случае граф задают матрицей инцидентности $I = \left\| i_{k,p} \right\|$, элементы $i_{k,p}$ которой равны либо -1 , либо 0 , либо 1 .

$i_{k,p} = -1$, если цепь e_p – выходная для элемента x_k ; $i_{k,p} = 0$, если цепь e_p не связана с элементом x_k ; $i_{k,p} = 1$, если цепь e_p – входная для элемента x_k .

Для коммутационной схемы (рис. 2.1) матрица инцидентности I запишется:

$$I = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \left\| \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ 1 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{array} \right\|.$$

Номера столбцов данной матрицы соответствуют электрическим цепям (e_1, e_2, \dots, e_6) , а номера строк соответствуют элементам схемы (x_1, x_2, x_3, x_4) . Соответствующая этой интерпретации коммутационной схемы математическая модель М4 представлена на рисунке 2.5.

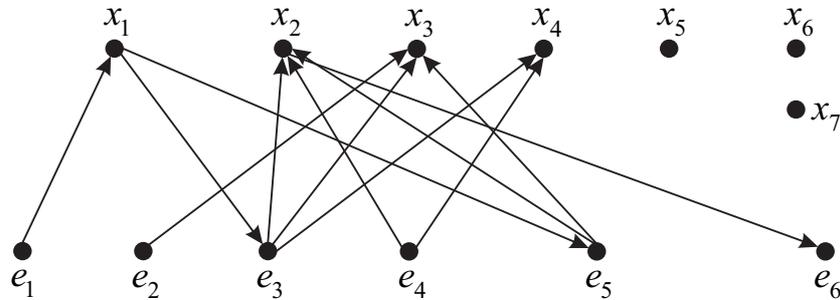


Рис. 2.5 – Математическая модель М4



Математическая модель М4 применяется для разработки алгоритмов трассировки соединений и когда требуется различать входы и выходы логических элементов.

Когда матрицы, представляющие граф, сильно разрежены, т. е. имеют мало ненулевых элементов, хранение информации в матричной форме не эффективно. В этом случае удобной формой представления матриц являются списки соединений.

2.6 Математическое моделирование цифровых устройств



Целью моделирования цифровых устройств (ЦУ) является получение картины логико-временного поведения ЦУ при различных входных воздействиях. Модели ЦУ, используемые при проектировании радиоустройств, можно разделить на две основные группы: физические и логические.

В первом случае отдельные элементы, из которых состоит ЦУ, представляются их электрическими макромоделями, состоящими из базовых элементов теории цепей, на основании которых формируется полная электрическая модель ЦУ. Система уравнений во временной области, соответствующая полной

электрической модели ЦУ, обычно представляет собой систему дифференциальных уравнений высокого порядка, решение которой требует больших затрат машинного времени.



Выводы

Физические модели ЦУ позволяют наиболее полно представить работу устройства во времени с учетом реальных задержек срабатывания элементов, но их целесообразно использовать на заключительном этапе проектирования из-за больших затрат времени на моделирование, когда необходимо иметь данные об устройстве, которые нельзя получить с помощью более простых моделей.



В моделях логического уровня каждый элемент ЦУ представляется упрощенной формальной моделью в виде логического соотношения, описывающего логику функционирования элемента. При меньшей детализации работы по сравнению с физическими моделями логические модели обладают во много раз большим быстродействием и позволяют на начальных этапах проектирования ЦУ решить ряд важных для практики задач:

- 1) проверить правильность логического функционирования ЦУ;
- 2) сравнить характеристики различных вариантов схемных решений;
- 3) разработать процедуры тестового контроля ЦУ и проверить их правильность и полноту;
- 4) проверить работу цепей установки ЦУ в начальное состояние.



В настоящее время разработано много алгоритмов, пакетов прикладных программ и даже специализированных языков для моделирования ЦУ на логическом уровне. Рассмотрим некоторые из них применительно к комбинационным цифровым устройствам.

2.6.1 Описание языков моделирования и элементов цифровых устройств в моделях логического уровня



При моделировании на логическом уровне физическая природа сигналов (ток или напряжение), распространяющихся в ЦУ, не конкретизируется. Сигналы задаются символами, совокупность которых описывает состояние и работу элементов, из которых состоит ЦУ. Совокупность различных символов, используемых при моделировании, называют **алфавитом логического моделирования**. Между реальными сигналами и символами алфавита всегда можно установить однозначное соответствие.

Простейший из используемых алфавитов для моделирования ЦУ на логическом уровне – **двоичный**, включающий всего два символа: 0 и 1. Он обеспечивает максимальную скорость моделирования, но не позволяет выявить неоднозначность работы ЦУ и характер переходных процессов.

Моделирование многозначными алфавитами позволяет получить больше информации о ЦУ, но и требует больших затрат машинного времени. Для анализа состязаний сигналов на элементах ЦУ используют троичный алфавит, содержащий кроме 0 и 1 еще символ неопределенного состояния X, которому ставится в соответствие процесс перехода из 0 в 1 и обратно либо безразличное состояние сигнала (рис. 2.6).

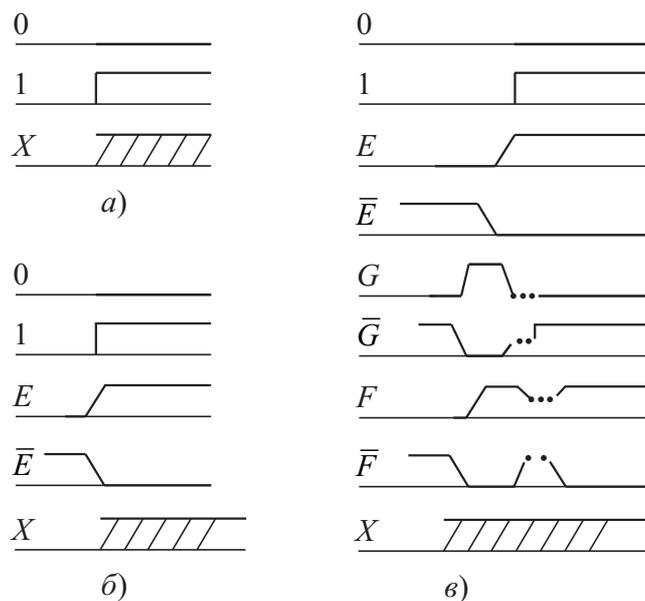


Рис. 2.6 – Алфавиты моделирования: а) – трехзначный; б) – пятизначный; в) – девятизначный



Пример

Для примера на рисунке 2.7, *а* приведены таблицы истинности для наиболее распространенных базовых элементов цифровых схем И, ИЛИ, НЕ при моделировании трехзначным алфавитом. При этом полагаем, что $\overline{\overline{X}} = X$.

Для уточнения характера процесса смены состояний элементов ЦУ используется пятизначный алфавит, показанный на рисунке 2.7, *б*, в котором символы 0, 1 и X имеют тот же смысл, что и в трехзначном алфавите, а символ E описывает гладкую смену сигнала из 0 в 1. Соответственно символ \overline{E} будет описывать обратное изменение сигнала из 1 в 0. Таблицы истинности для элементов И, ИЛИ, НЕ при моделировании их пятизначным алфавитом приведены на рисунке 2.7, *б*.

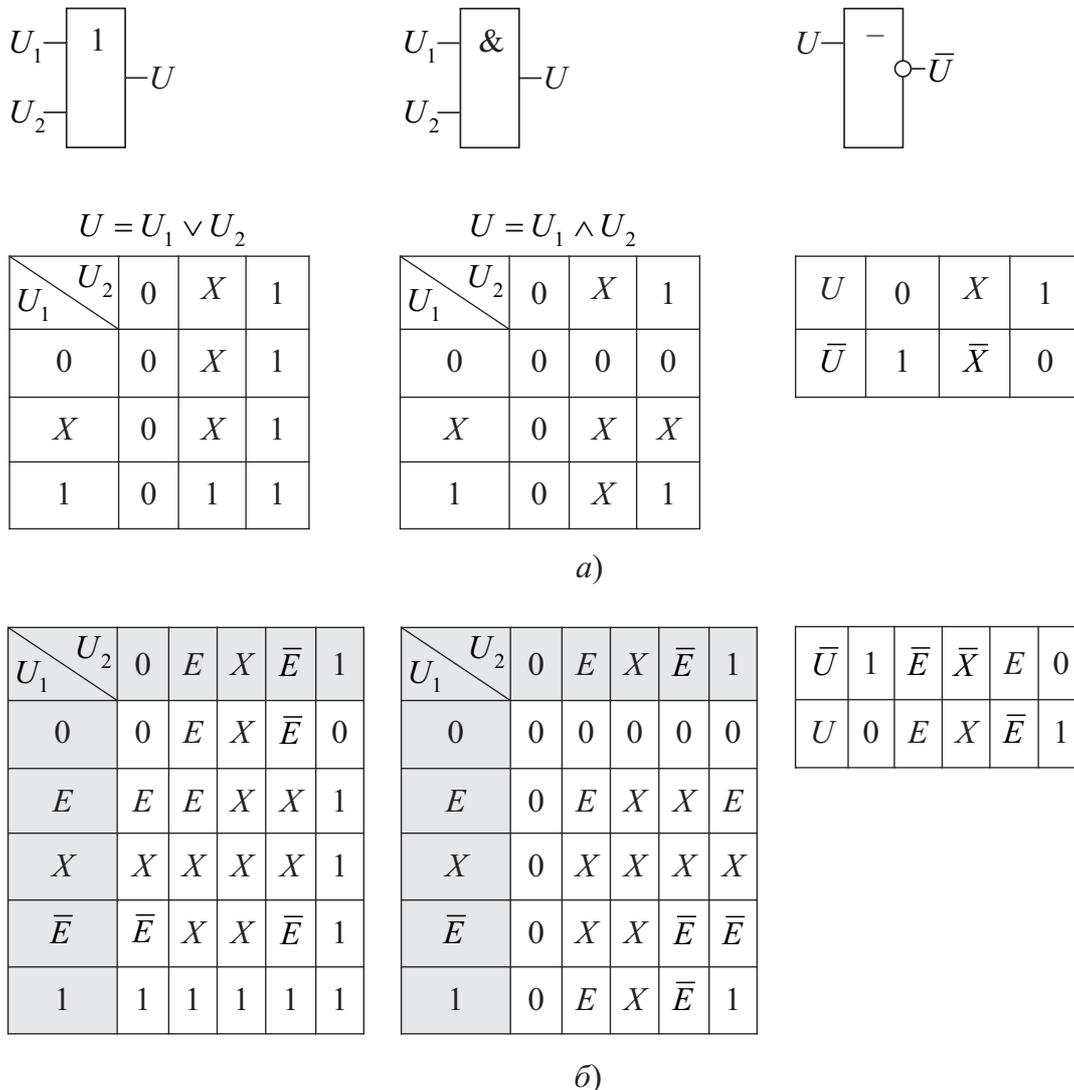
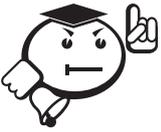


Рис. 2.7 – Таблицы истинности простых логических элементов для трехзначного (*а*) и пятизначного (*б*) алфавитов



.....

Более подробно характер переходных процессов в ЦУ описывается при использовании девятизначного алфавита, включающего символы $0, 1, X, E, \neg E, G, \neg G, F, \neg F$. По сравнению с пятизначным алфавитом здесь добавлены символы G и F , соответствующие статическим и динамическим сбоям при переключении из 0 в 1 и из 1 в 0 [1].

При моделировании ЦУ с помощью многозначных алфавитов отдельные элементы моделируются многозначными таблицами истинности, что приводит к увеличению времени моделирования и требуемого объема памяти. В настоящее время разработаны и другие, в том числе бесконечнозначные, алфавиты.

Как указывалось выше, в качестве моделей элементов цифровых устройств логическом уровне используются формальные модели «черного ящика», в которых связь между входными и выходными сигналами элемента задается с помощью булевых уравнений либо таблиц истинности. Это прежде всего относится к простейшим базовым элементам цифровых схем И, ИЛИ, НЕ – сложению по модулю два и др., в виде комбинаций которых может быть реализовано любое цифровое устройство. Более сложные элементы цифровых устройств – триггеры, регистры, устройства памяти и т. п. – в одних случаях представляются комбинациями простейших базовых компонентов, в других, на уровне регистровых передач, их описывают в терминах выполняемых ими операций, не прибегая к разложению на составляющие их простейшие элементы. Такой подход оказывается наиболее эффективным при моделировании цифровых устройств на основе микросхем средней или большой степени интеграции, позволяя существенно сократить время моделирования. Возможна и комбинация обоих подходов, когда в одних и тех же программах моделирования используются модели простейших базовых элементов цифровых схем и модели, более сложные на уровне регистровых передач. Описание соединений элементов цифровых устройств в ЭВМ может быть выполнено в форме списков, логической сети либо таблиц в зависимости от конкретного языка описания цифровых устройств, используемого в программе.

.....

Как следует из вышесказанного, математическая модель цифровых устройств будет представлять собой систему булевых уравнений, каждое из которых описывает один элемент цифровых устройств либо многомерных таблиц истинности. Такие математические модели могут быть организованы в ЭВМ различным образом.

В зависимости от способа организации модели цифровых устройств можно разделить на компилирующие и интерпретивные.



В моделях компилирующего типа исходное описание цифровых устройств в виде логической сети или таблицы транслируется на язык машинных кодов и оформляется в виде объектного модуля, который затем и выполняется в ЭВМ в процессе моделирования. Для такого перевода описания ЦУ на язык машинных кодов используется специальный компилятор, являющийся частью моделирующей программы. Достоинством такого подхода является большая скорость работы программы моделирования, недостатком – необходимость разработки компилятора, что представляет собой сложную задачу, а также необходимость повторных компиляций при внесении изменений в цифровые устройства.

В моделях цифровых устройств интерпретивного типа связи между отдельными элементами ЦУ представляются в виде таблиц, описание ЦУ в машинные коды не переводится, и каждое логическое уравнение, представляющее математическую модель того или иного элемента ЦУ, решается с помощью специальной подпрограммы.

Выбор очередной подпрограммы производится специальной интерпретирующей программой, которая использует записанные в таблицу адреса перехода от одного элемента цифрового устройства к другому. Обращение к подпрограмме, моделирующей элемент, будет происходить всякий раз, когда интерпретатору понадобится выполнить данную логическую операцию. Модели интерпретивного типа оказываются более простыми и менее трудоемкими в разработке, однако они имеют меньшее быстродействие по сравнению с компилятивными.

В зависимости от учета или неучета задержек в срабатывании ЦУ при моделировании модели ЦУ подразделяются на синхронные и асинхронные, а по

способу организации процесса решения системы уравнений – на сквозные и событийные.

2.6.2 Синхронное моделирование цифровых устройств двоичными алфавитами

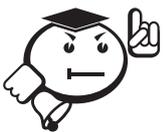


.....

При синхронном моделировании ЦУ на логическом уровне не учитываются задержки срабатывания отдельных элементов, из которых состоит ЦУ.

.....

Это позволяет свести моделирование к последовательному вычислению сигналов на выходах элементов ЦУ по значениям сигналов на их входах с помощью булевых уравнений или таблиц истинности, моделирующих каждый элемент. При этом считается, что независимыми переменными, синхронизирующими работу модели устройства, являются события – изменение сигналов на входах цифрового устройства. После определения состояния, в которое перейдет ЦУ при очередной смене входных сигналов, осуществляется сдвиг модельного времени до момента наступления очередного события. В промежутке между двумя событиями смены сигналов на входах цифрового устройства не происходит, а если на входах устройства появятся новые сигналы, то они будут отнесены к началу следующего такта моделирования.



.....

Контроль временного интервала между событиями необходим для построения временных диаграмм, наглядно представляющих работу цифровых устройств в предположении нулевой длительности переходных процессов.

.....

По принципу работы такая модель соответствует работе синхронного цифрового автомата, в котором сигналы на входы ЦУ поступают только в моменты подачи синхросигналов, в остальное время входные сигналы не могут изменять состояние ЦУ. Переходные процессы в устройстве обязательно заканчиваются к моменту прихода следующего синхросигнала. Синхронные модели можно использовать для моделирования синхронных и асинхронных цифровых автоматов и термин «синхронное моделирование», общепринятый в литературе, означает только неучет задержек в работе отдельных элементов цифрового устройства.



.....

Математическая модель цифрового устройства без учета задержек представляет собой систему булевых уравнений, связывающих между собой сигналы на входах и выходах элементов ЦУ. В общем виде такая система может быть записана следующим образом:

$$\begin{aligned}\bar{U} &= L(\bar{U}_Q, \bar{U}), \\ \bar{F} &= \bar{F}(\bar{U}),\end{aligned}\tag{2.9}$$

где \bar{U}_Q – вектор сигналов на входах ЦУ; \bar{U} – вектор сигналов на внутренних узлах схемы ЦУ (входах и выходах элементов); L – логический функционал; \bar{F} – вектор выходных сигналов, интересующих разработчика.

.....

Система логических уравнений (2.9) при каждом событии – изменение входных сигналов, решается итерационно относительно вектора \bar{U} , по координатам которого определяются требуемые выходные сигналы устройства \bar{F} .



.....

Алгоритмы решения системы логических уравнений делят на две группы: сквозные и событийные. В первом случае при итерационном решении системы логических уравнений математической модели ЦУ на каждой итерации решаются все логические уравнения и процесс решения системы $\bar{U} = L(\bar{U}_Q, \bar{U})$ во многом напоминает решение итерационными методами системы линейных алгебраических уравнений. Сквозное моделирование может выполняться методом простой итерации, когда для определения сигналов в ЦУ на последующей итерации \bar{U}^k используются сигналы на предыдущей итерации \bar{U}^{k-1} . Итерационный процесс решения системы логических уравнений по *методу простой итерации* может быть записан в виде

$$U^k = L(\bar{U}_Q, \bar{U}^{k-1}),\tag{2.10}$$

или в скалярном виде

$$u_i^k = L_i(\bar{U}_Q, u_1^{k-1}, \dots, u_n^{k-1}), \text{ где } i = 1, 2, \dots, n.$$

.....

Если решение системы логических уравнений ведется по методу итерации Зейделя, то сигналы, определенные при решении предыдущих уравнений системы, используются для нахождения сигналов, определяемых последующими уравнениями. В этом случае итерационная формула в скалярной форме записи имеет вид:

$$\begin{aligned} u_i^k &= L_i \left(\bar{U}_Q u_1^k, \dots, u_{i-1}^k, u_{i+1}^{k-1} \dots u_n^{k-1} \right), \\ u_{i+1}^k &= L_{i+1} \left(\bar{U}_Q u_1^k, \dots, u_i^k, u_{i+2}^{k-1} \dots u_n^{k-1} \right). \end{aligned} \quad (2.11)$$

Итерационный процесс решения системы логических уравнений продолжается до тех пор, пока сигналы на всех узлах цифрового устройства на двух последних итерациях не совпадут, что свидетельствует о получении установившегося состояния цифрового устройства. Для ускорения итерационного процесса расположение уравнений в системе логических уравнений математической модели ЦУ необходимо ранжировать в порядке распространения сигналов через элементы устройства.

При программной реализации алгоритмов значения сигналов на узлах цифрового устройства на каждой итерации U^k хранятся в специальном массиве – рабочем поле РП (k), причем в рабочем поле, соответствующем нулевой итерации РП (0), хранятся сигналы, соответствующие начальному состоянию цифрового устройства перед изменением входных сигналов.

Изменения входных сигналов записываются в рабочее поле на первой итерации РП (1). На последующих итерациях до получения решения сигналы на входных узлах цифрового устройства при синхронном моделировании не изменяются. После получения установившегося решения сдвигается модельное время и начинается моделирование следующего события, а в качестве начального состояния ЦУ для него используется решение, полученное на предыдущем шаге.



Пример

Рассмотрим процесс сквозного синхронного моделирования цифрового устройства двоичным алфавитом на примере устройства, приведенного на рисунке 2.8, а. Математическая модель цифрового устройства представляет собой систему логических уравнений:

$$\begin{aligned}
 u_5 &= u_1 \wedge u_3, & u_8 &= \overline{u_4 \wedge u_3}, \\
 u_6 &= \overline{u_5 \vee u_7}, & u_9 &= \overline{u_6 \wedge u_{10}}, \\
 u_7 &= u_2 \wedge u_3, & u_{10} &= \overline{u_8 \wedge u_9},
 \end{aligned}$$

где u_i – сигналы на узлах устройства.

$\bar{U}_Q = |U_1, U_2, U_3, U_4|$ – вектор входных сигналов.

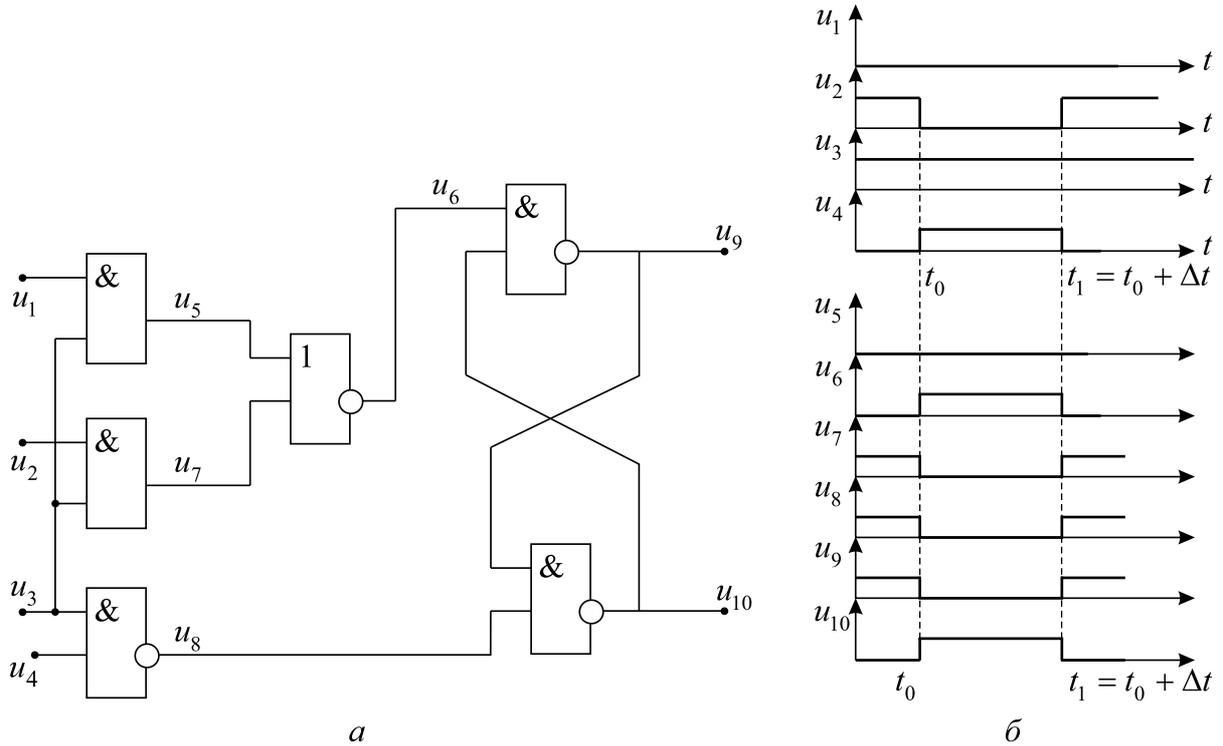


Рис. 2.8 – Схема ЦУ (а) и временные диаграммы ее работы (б)

При сквозном моделировании по методу простой итерации в соответствии с выражением (2.10) итерационные формулы запишутся следующим образом:

$$\begin{aligned}
 u_5^k &= u_1^{k-1} \wedge u_3^{k-1}, & u_8^k &= \overline{u_3^{k-1} \wedge u_4^{k-1}}, \\
 u_6^k &= \overline{u_5^{k-1} \vee u_7^{k-1}}, & u_9^k &= \overline{u_6^{k-1} \wedge u_{10}^{k-1}}, \\
 u_7^k &= u_2^{k-1} \wedge u_3^{k-1}, & u_{10}^k &= \overline{u_8^{k-1} \wedge u_9^{k-1}}.
 \end{aligned}$$

Процесс сквозного моделирования ЦУ по методу простой итерации при изменении входных сигналов U_1, U_2, U_3, U_4 , показанном на рисунке 2.8, б, иллюстрируется таблицей 2.2.

Временные диаграммы, соответствующие приведенным результатам моделирования, показаны на рисунке 2.8, б.

Таблица 2.2 – Процесс сквозного моделирования ЦУ по методу итерации Зейделя

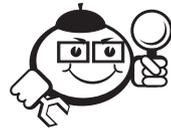
Изменение входных сигналов	Номер итерации	Сигналы на узлах цифрового устройства (ЦУ)									
		Входные				Внутренние и выходные					
		u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
$t = t_0$	0	0	1	1	0	0	0	1	1	1	0
Изменение сигналов	1	0	0	1	1	0	0	1	1	1	0
	2	0	0	1	1	0	0	0	0	1	0
	3	0	0	1	1	0	1	0	0	1	1
	4	0	0	1	1	0	1	0	0	0	1
Решение	5	0	0	1	1	0	1	0	0	0	1
$t = t_0 + \Delta t$	0	0	0	1	1	0	1	0	0	0	1
Изменение сигналов	1	0	1	1	0	0	1	0	0	0	1
	2	0	1	1	0	0	1	1	1	0	1
	3	0	1	1	0	0	0	1	1	0	1
	4	0	1	1	0	0	0	1	1	1	1
	5	0	1	1	0	0	0	1	1	1	0
Решение	6	0	1	1	0	0	0	1	1	1	0



Выводы

Сквозное моделирование по методу простой итерации может использоваться для синхронных и асинхронных моделей ЦУ, недостатком его является большое время, затрачиваемое на решение системы логических уравнений, представляющих математическую модель цифровых устройств на логическом уровне из-за большого числа итераций. Существенно сократить время моделирования можно за счет выполнения итерационного процесса по методу итерации Зейделя. В соответствии с (2.11) для нашего примера итерационные формулы запишутся следующим образом:

$$\begin{aligned}
 u_5^k &= u_1^k \wedge u_3^k, & u_8^k &= \overline{u_3^k \wedge u_4^k}, \\
 u_6^k &= \overline{u_5^k \vee u_7^k}, & u_9^k &= \overline{u_6^k \wedge u_{10}^k}, \\
 u_7^k &= \overline{u_2^k \wedge u_3^k}, & u_{10}^k &= \overline{u_8^k \wedge u_9^k}.
 \end{aligned}$$



Пример

Процесс сквозного моделирования цифровых устройств (ЦУ), показанного на рисунке 2.8, *a*, по методу итерации Зейделя при таком же изменении входных сигналов, что и в предыдущем случае, приведен в таблице 2.3.

Таблица 2.3 – Сквозное моделирование ЦУ по методу итерации Зейделя

Изменение входных сигналов	Номер итерации	Сигналы на узлах цифрового устройства (ЦУ)									
		Входные				Внутренние и выходные					
		u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
$t = t_0$	0	0	1	1	0	0	0	1	1	1	0
Изменение сигналов	1	0	0	1	1	0	0	0	0	1	1
	2	0	0	1	1	0	1	0	0	0	1
Решение	3	0	0	1	1	0	1	0	0	0	1
$t = t_0 + \Delta t$	0	0	0	1	1	0	1	0	0	0	1
Изменение сигналов	1	0	1	1	0	0	1	1	1	0	1
	2	0	1	1	0	0	0	1	1	1	0
Решение	3	0	1	1	0	0	0	1	1	1	0

Как видно из таблицы 2.3, решение системы логических уравнений по методу итерации Зейделя позволяет существенно сократить число итераций, однако этот метод неприменим при асинхронном моделировании цифровых устройств с учетом задержек срабатывания элементов.



Пример

Приведем пошаговое описание алгоритма для разового изменения входных сигналов.

Алгоритм 1. Сквозное моделирование цифровых устройств (ЦУ) двоичным алфавитом по методу простой итерации.

Шаг 1. Формируется рабочее поле на нулевой итерации РП (0), соответствующее начальному состоянию устройства.

Шаг 2. Фиксируется изменение входных сигналов, формируется рабочее поле на первой итерации РП (1).

Шаг 3. Номер итерации k полагается равным 2.

Шаг 4. Начиная с первого логического уравнения системы уравнений математической модели ЦУ определяется новое значение сигналов на выходе элемента ЦУ, описываемого данным уравнением.

Шаг 5. Если сигнал на выходе элемента изменился, то в рабочее поле для k -й итерации РП (k) вносится новое значение сигнала, иначе переписывается сигнал из рабочего поля РП ($k - 1$), соответствующего предыдущей, ($k - 1$)-й итерации.

Шаг 6. Если вычислены все сигналы на k -й итерации, то переходим к шагу 7, иначе к шагу 4.

Шаг 7. Сравниваются рабочие поля на k -й и ($k - 1$)-й итерациях; если они совпадают, то моделирование отклика ЦУ на разовое изменение входных сигналов окончено, иначе увеличиваем номер шага итерации на 1 и переходим к шагу 4.



Значение сигнала на выходе элемента ЦУ может измениться только в том случае, если изменился сигнал по крайней мере на одном из его входов. Это обстоятельство положено в основу работы событийных алгоритмов синхронного моделирования ЦУ, в которых в отличие от алгоритмов сквозного моделирования на каждой итерации решаются не все логические уравнения системы, а только уравнения для тех элементов, у которых на предыдущей итерации изменились входные сигналы. Для организации такого вычислительного процесса в конце каждой итерации решения системы ло-

гических уравнений после сравнения рабочих полей на соседних итерациях формируется список изменившихся сигналов, по которому можно определить активизированные элементы, т. е. элементы, на входах которых изменился сигнал и для которых на следующей итерации необходимо решать логические уравнения. Итерационный процесс прекращается, когда в модели ЦУ не будет изменившихся сигналов и активизированных элементов.



Пример

Процесс моделирования ЦУ событийным алгоритмом, схема которого представлена на рисунке 2.8, *a*, приведен в таблице 2.4.

Таблица 2.4 – Моделирование ЦУ событийным алгоритмом

Изменение входных сигналов	Номер итерации	Сигналы на узлах цифрового устройства (ЦУ)											
		Входные				Внутренние и выходные						Изменившиеся сигналы	Активизированные элементы
		u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}		
$t = t_0$	0	0	1	1	0	0	0	1	1	1	0	–	–
Изменение сигналов	1	0	0	1	1	0	0	1	1	1	0	u_2, u_4	u_8, u_7
	2	0	0	1	1	0	0	0	0	1	0	u_7, u_8	u_6, u_{10}
	3	0	0	1	1	0	1	0	0	1	1	u_6, u_{10}	u_9
	4	0	0	1	1	0	1	0	0	0	1	u_9	u_{10}
Решение	5	0	0	1	1	0	1	0	0	0	1	–	–
$t = t_0 + \Delta t$	0	0	0	1	1	0	1	0	0	0	1	–	–
Изменение сигналов	1	0	1	1	0	0	1	0	0	0	1	u_2, u_4	u_8, u_7
	2	0	1	1	0	0	1	1	1	0	1	u_7, u_8	u_6, u_{10}
	3	0	1	1	0	0	0	1	1	0	1	u_6	u_9
	4	0	1	1	0	0	0	1	1	1	1	u_9	u_{10}
	5	0	1	1	0	0	0	1	1	1	0	u_{10}	u_9

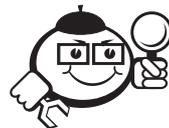
Решение	6	0	1	1	0	0	0	1	1	1	0	–	–
---------	---	---	---	---	---	---	---	---	---	---	---	---	---

Начальные условия, изменения входных сигналов и временные диаграммы совпадают с предыдущими примерами.



Выводы

Как видно из работы событийного алгоритма, здесь наиболее удобной представляется интерпретивная организация математической модели цифрового устройства, однако событийное моделирование может быть и компилятивным. Как и сквозное моделирование, событийное может выполняться методами простой итерации и итерации Зейделя. Событийное моделирование целесообразно применять в тех случаях, когда выигрыш от уменьшения числа решаемых уравнений на всех итерациях превышает затраты на анализ активизации элементов цифрового устройства.



Пример

Приведем пошаговое описание работы алгоритма для разового изменения входных сигналов.

Алгоритм 2. Синхронное событийное моделирование цифровых устройств двоичным алфавитом по методу простой итерации.

Шаг 1. Формируется рабочее поле на нулевой итерации РП (0), соответствующее начальному состоянию ЦУ, списки изменившихся сигналов и активизированных элементов полагаются пустыми:

$$I_k = 0, A_k = 0.$$

Шаг 2. Фиксируется изменение входных сигналов, формируется рабочее поле первой итерации РП (1). Номер итерации полагается равным $k - 2$.

Шаг 3. Сравниваются рабочие поля на $(k - 1)$ -й и $(k - 2)$ -й итерациях РП $(k - 1)$ и РП $(k - 2)$, формируется список изменившихся сигналов I_k , если $I_k = 0$, то переходим к шагу 8.

Шаг 4. На основании I_k и описания ЦУ определяется список активизированных элементов A_k .

Шаг 5. Начиная с первого элемента списка A_k решается логическое уравнение, моделирующее активизированный элемент ЦУ, и определяется новое значение сигнала на его выходе.

Шаг 6. Если сигнал на выходе элемента изменился, то в рабочее поле для k -й итерации РП (k) вносится новое значение сигнала, иначе переписывается выходной сигнал из рабочего поля РП ($k - 1$).

Шаг 7. Если промоделированы все элементы из списка A_k , то полагаем $k = k + 1$ и переходим к шагу 3, иначе к шагу 5.

Шаг 8. Фиксируем состояние ЦУ после изменения входных сигналов на шаге 2.

.....



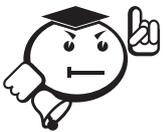
Рассмотренные выше алгоритмы синхронного моделирования цифровых устройств двоичным алфавитом обладают высоким быстродействием, однако результаты моделирования ими не всегда могут выявить ошибки в работе ЦУ, которые могут возникнуть из-за задержек в срабатывании элементов ЦУ, кроме того, синхронное моделирование не позволяет выявить характер переходных процессов в устройстве.

.....

2.6.3 Моделирование неисправностей в цифровых устройствах и синтез диагностических тестов

В процессе разработки, производства и эксплуатации ЦУ в составе различных радиоэлектронных средств возникают задачи проверки правильности функционирования цифровых устройств и отыскания неисправностей в них. Фактически эти вопросы можно свести к решению двух задач: построению набора входных сигналов, проверяющих правильность работы устройства, и разработке входных наборов, позволяющих локализовать тип и место неисправности в ЦУ. При этом контролирующие и диагностические тесты должны выявлять как устойчивые, так и перемежающиеся неисправности. Разработка тестов является трудоемкой задачей, решение которой в большинстве случаев возможно лишь с применением двоичного, реже троичного алфавитов.

Поведение неисправного элемента в составе ЦУ можно имитировать, изменив соответствующим образом математическую модель устройства. Логическое уравнение, отражающее правильную работу элемента, заменяется другим, которое отражает поведение элемента в данном неисправном состоянии. Недостатком такого подхода к моделированию неисправностей в ЦУ являются большие затраты машинного времени на модификацию моделей при внесении или исключении неисправностей. В последнее время чаще отображают неисправность элемента ЦУ логическим сложением или умножением его входных сигналов с соответствующей маской. Например, при моделировании в схеме ИЛИ неисправности типа «постоянная 1 на выходе» достаточно каждый раз при моделировании этого элемента в набор его входных сигналов вводить логическую единицу сложением набора выходных сигналов с маской, содержащей символ логической единицы. Используя логическое умножение, можно задать неисправность типа «постоянный 0 на выходе», для чего следует произвести умножение входных сигналов на маску, содержащую только символы логического нуля. В этом случае оказывается ненужной отдельная программа, управляющая моделированием дефектов, а устанавливать и отменять неисправности можно в любом месте, не затрагивая структуру математической модели ЦУ.



.....

Для комбинационных схем ЦУ существует ряд хорошо разработанных методов построения контролирующих и диагностических тестов. По способу формирования их можно разделить на две группы: алгоритмы случайного поиска и направленные алгоритмы.

.....

В первом случае, задав в ЦУ некоторую неисправность, с помощью датчиков случайных чисел формируют наборы входных сигналов и для каждого из них анализируются диагностические способности. Способность набора входных сигналов выявлять неисправности в ЦУ проверяют сравнением отклика ЦУ при отсутствии неисправностей и при их наличии. Различие откликов свидетельствует о возможности диагностирования неисправности. После моделирования ЦУ неэффективные наборы входных сигналов отбрасывают исходя из следующих правил: *набор входных сигналов должен обнаруживать максимальное число неисправностей, а число наборов, проверяющих работу всех элементов ЦУ, должно быть минимальным.*

Направленные алгоритмы, наоборот, формируют наборы входных сигналов, предназначенные для выявления либо определенной неисправности, либо

проверки данного участка ЦУ. По результатам моделирования определяется эффективность той или иной комбинации входных сигналов и решается вопрос о включении ее в набор контролирующих и диагностических тестов. Все направленные алгоритмы связаны с большим перебором вариантов наборов входных сигналов, поэтому время поиска оптимального входного набора обычно ограничивается программно.

Построение тестов для ЦУ с памятью является более сложной задачей, чем для комбинационных устройств. Это объясняется тем, что для проверки ЦУ с памятью обычно необходимо использовать не один набор входных сигналов, а последовательность таких наборов, кроме того, дополнительные трудности вносит возможная неоднозначность поведения устройства и необходимость учета состязаний сигналов в исправном и неисправном устройствах. По этим причинам объем моделирования при подборе тестов для ЦУ с памятью в несколько раз больше, чем для комбинационных. Алгоритмы, используемые для отыскания тестов для ЦУ с памятью, как правило, представляют собой развитие соответствующих алгоритмов для комбинационных ЦУ.



В заключение качественно сравним эффективность алгоритмов случайного и направленного поиска диагностических тестов. При использовании алгоритмов случайного поиска время генерации случайного набора входных сигналов много меньше времени моделирования ЦУ и за короткое время можно получить тестовые входные наборы, проверяющие неисправности, для которых число проверяющих наборов велико. Затем эффективность работы алгоритма резко снижается, так как остаются неисправности, проверяющиеся малым числом входных наборов, вероятность генерации которых мала. Это иллюстрируется пунктирной кривой на рисунке 2.9.

Для направленных алгоритмов время получения каждого набора входных сигналов сравнимо со временем моделирования либо значительно больше его. Направленные алгоритмы поиска диагностических тестов, таким образом, работают медленнее, чем алгоритмы случайного поиска, но обеспечивают большую полноту тестов (сплошная кривая на рисунке 2.9). Наиболее целесообразно совместное использование алгоритмов: на начальных этапах используются алгоритмы случайного поиска, а затем используются направленные алгоритмы.

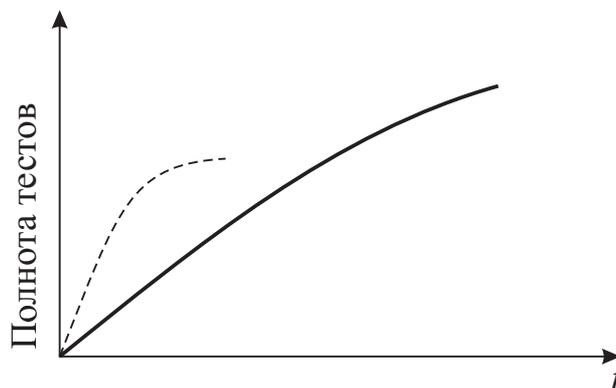


Рис. 2.9 – Эффективность алгоритмов случайного и направленного поиска диагностических тестов

При производстве сложных ЦУ и БИС обычно разработанные тесты с помощью систем АСНИ под управлением ЭВМ через интерфейсные устройства подаются на входные узлы ЦУ и используются для поиска и устранения неисправностей. Реакция ЦУ на поданные входные сигналы вводится в ЭВМ, где и производится анализ работы ЦУ сравнением отклика реального устройства с теоретически полученными значениями.



Контрольные вопросы по главе 2

1. Каким требованиям должны отвечать математические модели, описывающие задачи конструирования РЭС?
2. Что понимают под коммутационной схемой?
3. Какие виды математических моделей используются для решения задач компоновки, размещения, трассировки?
4. Какие ЭС относятся к цифровым устройствам?
5. Что понимают под алфавитом логического моделирования?
6. Представьте таблицу истинности для базовых элементов И, ИЛИ, НЕ в трехзначном алфавите.
7. Назовите основные положения метода Зейделя для решения системы логических уравнений.
8. В чем состоит отличие метода сквозного моделирования от событийного?
9. Каким способом можно выявлять ошибки в функционировании цифровых устройств?

3 Типовые проектные процедуры



.....

Проектная процедура называется **типовой**, если она предназначена для многократного применения при проектировании различных объектов. Различают две основные типовые проектные процедуры – **анализ** и **синтез** (рис. 3.1).

.....



Рис. 3.1 – Классификация типовых проектных процедур



.....

Синтез заключается в создании «проекта» (описания) объекта. **Задача синтеза** – определение структуры и параметров элементов объекта по заданным техническим требованиям, т. е. при известных векторах \vec{T} и \vec{Q} определить вектор внутренних параметров \vec{X} (структуру системы уравнений ММ и значения ее коэффициентов). Под **структурой объекта** понимают состав его элементов и способы связи элементов друг с другом. **Параметр объекта** – величина, характеризующая некоторое свойство объекта или режим его функционирования. Примерами процедур структурного синтеза служат синтез структурной схемы с корректирующими устройствами (структура которой выражается перечнем входящих в нее звеньев и их соединений) или синтез алгоритма (его структура определяется составом и последовательностью операторов). Процедура параметрического синтеза заключается в расчете значений параметров элементов при заданной

структуре объекта, например коэффициентов корректирующих устройств.

Анализ состоит в оценке свойств «проекта» и исследовании его работоспособности или, иначе, определении свойств объекта по его описанию. **Задача анализа** – определение рабочих характеристик объекта при известной структуре и параметрах элементов, т. е. при заданных векторах \vec{X} и \vec{Q} определить вектор выходных параметров \vec{Y} . **Цель анализа** – проверка работоспособности объекта. Часто задача анализа формулируется как задача установления соответствия двух различных описаний одного и того же объекта. При этом одно из описаний считается первичным и его корректность предполагается установленной. Другое описание относится к более подробному уровню иерархии или к другому аспекту, и его правильность нужно установить сопоставлением с первичным описанием. Такое сопоставление называется **верификацией**.

.....

Процедуры анализа подразделяются на процедуры одно- и многовариантного анализа.

При *одновариантном анализе* заданы значения внутренних и внешних параметров, требуется определить значения выходных параметров объекта. При этом используется только один вариант (один вектор) значений внутренних параметров.

Полезно использовать геометрическую интерпретацию этой задачи, связанную с понятием пространства внутренних параметров. Это n -мерное пространство, в котором для каждого из n внутренних параметров x_i ($i = \overline{1, n}$) выделена координатная ось. При одновариантном анализе задается некоторая точка в пространстве внутренних параметров \vec{X} и требуется в этой точке определить значения выходных параметров \vec{Y} при заданных внешних воздействиях \vec{Q} (рис. 3.2). Эта задача сводится к однократному решению системы уравнений, составляющих ММ объекта.

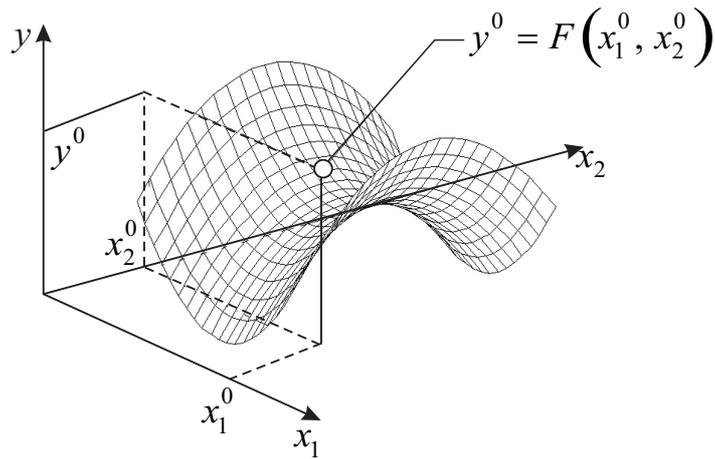


Рис. 3.2 – Одновариантный анализ ММ вида $y = F(x_1, x_2) \rightarrow y^0 = F(x_1^0, x_2^0)$



.....

Многовариантный анализ состоит в исследовании свойств объекта в некоторой области пространства внутренних параметров \vec{X} . Такой анализ требует многократного применения процедуры одновариантного анализа и, следовательно, многократного решения системы уравнения ММ.

.....



.....

Целью *структурного синтеза* является определение структуры объекта $S(\vec{X})$, т. е. типа составляющих его элементов и связей между ними в составе объекта по заданным техническим требованиям к выходным характеристикам.

Параметрический синтез заключается в определении числовых значений параметров элементов (т. е. значений вектора \vec{X}) при заданной структуре $S(\vec{X})$ и условиях работоспособности (векторах

\vec{Q} , \vec{T}^- и \vec{T}^+), т. е. при параметрическом синтезе нужно найти точку или область в пространстве внутренних параметров, в которых выполняются те или иные требования (обычно условия работоспособности). Таким образом, в результате параметрического синтеза необходимо определить область допустимых значений выходных характеристик \vec{Y} в пространстве внутренних параметров \vec{X} (рис. 3.3).

.....

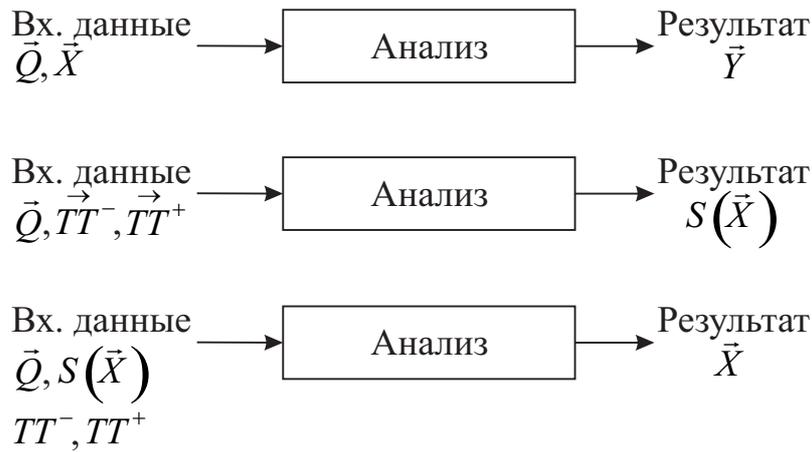


Рис. 3.3 – Задачи анализа и синтеза

3.1 Маршруты проектирования и принципы их построения



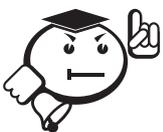
Маршрутом проектирования называется последовательность проектных процедур, ведущая к получению требуемых проектных решений.



Основные принципы построения маршрутов проектирования:

- расчленение сложной задачи синтеза полного комплекта конструкторско-технологической документации на более простые задачи синтеза промежуточных проектных решений;
- чередование процедур синтеза и верификации;
- итерационность проектирования;
- усиление тщательности анализа (многовариантность, усложнение моделей) по мере приближения к окончательному проектному решению [4].

Расчленение сложной задачи синтеза на ряд простых выполняется в соответствии с блочно-иерархическим подходом к проектированию.



Расчленение позволяет организовать параллельно-последовательное выполнение проектных процедур коллективом разработчиков.

Чередование процедур синтеза и верификации обусловлено тем, что для большинства задач структурного синтеза отсутствуют методы, обеспечивающие безошибочное получение проектных решений, удовлетворяющих требованиям ТЗ. Это связано с трудностями формализации задач синтеза, поэтому основные решения принимает человек на основе эвристических приемов. При этом невозможно учесть все многообразие качественных и количественных требований и избежать ошибок. Поэтому результаты предложенных при синтезе проектных решений контролируются выполнением верификации.

Итерационность проектирования обусловлена двумя факторами. Во-первых, она вытекает из особенностей блочно-иерархического подхода. Действительно, при нисходящем проектировании на n -м иерархическом уровне можно лишь предположительно судить о свойствах неспроектированных элементов, которые будут разрабатываться на следующем $(n + 1)$ -м уровне. При восходящем проектировании неопределенность связана с требованиями ТЗ, корректность которых может быть установлена только при выполнении процедур самого верхнего иерархического уровня. Поэтому ошибочность или неоптимальность решений, полученных на предыдущих этапах, выявляется в последующем, что требует возврата к предыдущим этапам для перепроектирования. Во-вторых, итерационность связана с чередованием синтеза и верификации, представляющим собой последовательное приближение к приемлемому проектному решению. Очевидно, что на первых итерациях синтезируемые варианты хуже с точки зрения выполнения ТЗ, чем последующие. Поэтому на первых итерациях с помощью довольно приближенных моделей полученные варианты оцениваются быстро и просто. Чем ближе очередной вариант к окончательному решению, тем более точное и всестороннее исследование требуется для его оценки. Следовательно, в процедурах верификации нужно использовать не одну модель объекта, а иерархический ряд моделей, различающихся сложностью и точностью.

Типичная последовательность проектных процедур

При проектировании технических объектов на любом из этапов проектирования можно выделить *типичную последовательность проектных процедур* (рис. 3.4).

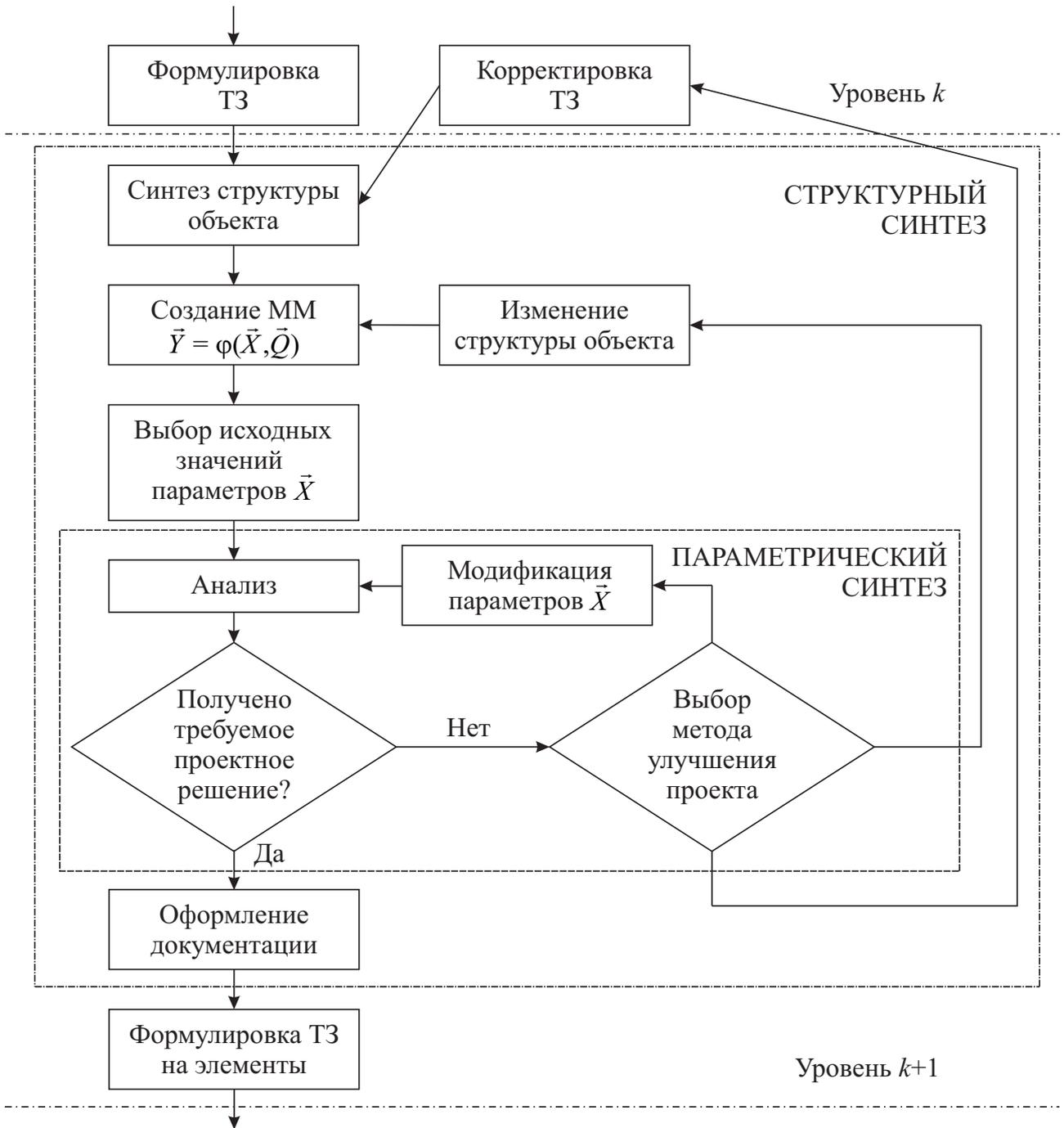


Рис. 3.4 – Схема процесса проектирования

Если на предыдущем этапе решались задачи k -го иерархического уровня, одним из результатов решения этих задач при нисходящем проектировании является формулировка ТЗ на проектирование систем $(k+1)$ -го рассматриваемого уровня. Проектирование системы на $(k+1)$ -м уровне начинается с синтеза исходного варианта её структуры, т. е. определения составляющих блоков и связей между ними. Для оценки этого варианта создается её модель: математическая – при автоматизированном проектировании, экспериментальная или стендовая – при традиционном. После выбора исходных значений параметров

элементов выполняется анализ варианта, по результатам которого становится возможной его оценка. Обычно оценка реализованного варианта заключается в проверке выполнения условий работоспособности, сформулированных в ТЗ. Если условия работоспособности выполняются в должной мере, то полученное проектное решение принимается, система $(k + 1)$ -го уровня описывается в принятой форме и далее формулируется ТЗ на проектирование элементов (блоков) данного уровня. Если же полученное проектное решение неудовлетворительное, то выбирается один из возможных путей улучшения проекта.

Обычно проще всего изменить числовые значения параметров элементов, составляющих вектор \vec{X} . Совокупность процедур модификации \vec{X} , анализа и оценки результатов анализа представляет собой процедуру *параметрического синтеза*. Если модификации составляющих вектора \vec{X} целенаправленны и подчинены стратегии поиска наилучшего значения некоторого показателя качества, то процедура параметрического синтеза является процедурой *оптимизации*. Возможно, что путем параметрического синтеза не удастся добиться приемлемой степени выполнения условий работоспособности. Тогда используют другой путь, связанный с модификацией структуры проектируемой системы.

Новый вариант структуры синтезируется, и для него повторяются процедуры формирования модели и параметрического синтеза. Если не удастся решить поставленную задачу и на этом пути, то ставится вопрос о корректировке ТЗ («ослаблении» требований), сформулированного на предыдущем k -м этапе. Такая корректировка может потребовать повторного выполнения ряда процедур k -го иерархического уровня, что и обуславливает итерационный характер процесса проектирования.



Рисунок 3.4 позволяет установить характерную особенность взаимосвязи проектных процедур анализа и синтеза. Взаимосвязь процедур анализа и синтеза носит характер *вложенности* (рис. 3.5). Анализ входит как составная часть параметрического синтеза (оптимизации), а параметрический синтез – в процедуру структурного синтеза, т. е. однократное выполнение процедуры оптимизации требует многократного выполнения процедуры анализа, в свою очередь, однократное решение задачи структурного синтеза – многократного решения задачи оптимизации. Очевидно, что такой же характер взаимодействия имеют процедуры анализа – многовари-

антный анализ основан на многократном применении процедуры одновариантного анализа.

.....

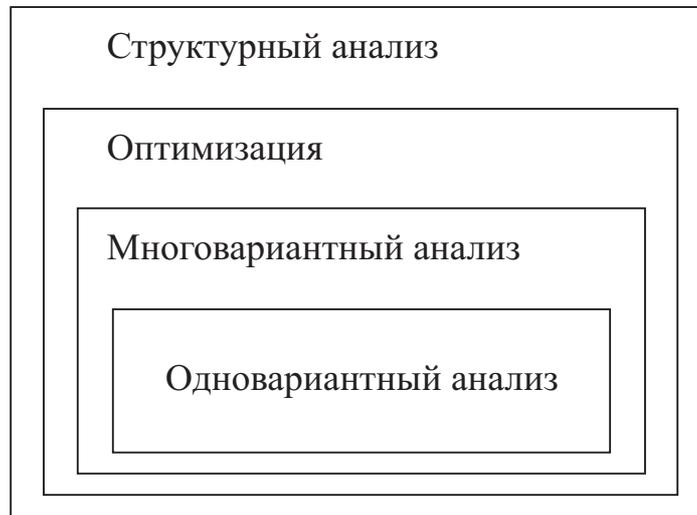


Рис. 3.5 – Вложенный характер взаимосвязи типовых проектных процедур



Выводы

.....

Таким образом, проектирование – трудоемкий процесс, требующий больших затрат вычислительных ресурсов и времени. Основные затраты при этом идут на решение задачи анализа. Чтобы уменьшить время проектирования, необходимо уменьшить время на анализ.

Одним из путей сокращения затрат вычислительных ресурсов является применение так называемых *адаптивных моделей*. При этом точные и сложные математические модели и алгоритмы используются только на завершающих итерациях процедуры синтеза. Первоначальная оценка вариантов проекта производится на основе упрощенных моделей и алгоритмов. Это позволяет отсеять большое количество бесперспективных вариантов на первых шагах синтеза и избежать больших затрат машинного времени на выполнение анализа.

.....

3.2 Место моделирования в проектировании РЭС

Процесс разработки РЭС включает в себя два вида основных работ: *научно-исследовательские* (НИР) и *проектно-конструкторские*. Эти работы различаются целями, методами решения и получаемыми результатами. При выполнении НИР задача ставится следующим образом: известны требования к

характеристикам разрабатываемого устройства (объекта). Необходимо обосновать электрическую схему и показать, что устройство, соответствующее этой схеме, можно сделать с применением известных способов изготовления. Специфика задачи состоит в следующем:

- хотя и известно, что устройство с заданными требованиями можно сделать, до сих пор его не делали;
- на данное устройство нет документации;
- неизвестно поведение устройства при изменении внешних факторов.

Основным методом работы при проведении НИР является моделирование. Различают физическое (натурное) и математическое моделирование.

При *физическом* моделировании создается макет исследуемого устройства и подготавливается измерительная установка, включающая в себя контрольно-измерительную аппаратуру и источники внешних воздействий. Моделирование представляет собой процедуру, включающую проверку принципа действия устройства, снятие экспериментальных зависимостей выходных параметров при различных внешних воздействиях и наборах значений внутренних параметров устройства, анализ экспериментальных результатов и принятие решения о направлении дальнейших исследований.

Достоинством физического моделирования является высокая достоверность, так как используются реальные компоненты и имеется возможность проводить исследования в реальных условиях эксплуатации РЭС.

Однако физическое моделирование требует больших затрат и не позволяет получить полную информацию о поведении объекта из-за ограниченного доступа к элементам моделируемого объекта. Например, нельзя измерить потенциалы внутри интегральной схемы и пр.

В настоящее время широкое распространение получило *математическое* моделирование, при котором исследование проводится на математической модели устройства, представляющей собой систему дифференциальных и алгебраических уравнений, таблицу или какое-либо другое формализованное описание объекта. Параметрам моделируемого объекта и внешним воздействиям соответствует определенный набор численных значений коэффициентов математической модели. Таким образом, задача сводится к заданию конкретного набора значений коэффициентов и получению решения системы уравнений математической модели.



При *математическом* моделировании отпадает необходимость подготовки экспериментального макета и измерительной установки. Однако точность математического моделирования ограничена, так как используются приближенные модели объектов и компонентов, из которых строится модель, и учитываются не все свойства объекта. Особенно эффективным является математическое моделирование на ЭВМ (машинный эксперимент) с помощью САПР. Применение САПР дает возможность отказаться от разработки оригинальных программ для моделирования конкретных устройств и значительно сократить затраты на постановку машинного эксперимента. САПР позволяет проектировать РЭС на уровне структурных, функциональных, принципиальных схем, а также выпускать конструкторскую документацию на РЭС [4].

Независимо от формы представления объекта (структурное, функциональное или какое-либо другое описание) моделирование в самой САПР проводится путем решения системы уровней математической модели, которая автоматически строится на описании объекта на входном языке САПР. Средства САПР позволяют задавать произвольные внешние воздействия и наблюдать выходные параметры в любых точках объекта, в том числе и в тех, в которые нет доступа при физическом моделировании.

Структурный синтез технического объекта в конструкторском проектировании электронных средств

Задача синтеза технического объекта включает в себя создание структуры проектируемого объекта и расчет его параметров. Эти две части синтеза соответственно называются структурным и параметрическим синтезом. *Задача структурного синтеза заключается в поиске оптимальной или рациональной структуры (схемы) технического объекта для реализации заданных функций в рамках выбранного принципа действия.*

Принцип получения математических моделей объектов проектирования электронно-вычислительной и радиоэлектронной аппаратуры для решения задач структурного синтеза рассмотрим на примерах компоновки, размещения, трассировки.

Задача компоновки

Под задачами компоновки понимают задачи разбиения множества $D = (d_1, d_2, \dots, d_n)$, содержащего n элементов, на ряд непересекающихся подмножеств D_k , $k = \overline{1, N}$ так, чтобы при этом выполнялись заданные ограничения и достигался экстремум некоторой функции качества $F(x)$.

При заданном числе N подмножеств разбиения задача компоновки формулируется следующим образом:

$$F(x) \rightarrow \min \quad (3.1)$$

и для $\forall k, l \in \{1, 2, \dots, N\}$ выполняются условия:

$$D_k \cap D_l = \emptyset, \quad (3.2)$$

$$\bigcup_{k=1}^N D_k = D, \quad (3.3)$$

где $D_k \subset D$ – множество элементов, принадлежащих k -му подмножеству разбиения при условии, что мощность $|D_k|$ каждого подмножества из разбиения задана, т. е.

$$|D_k| = n_k; \quad \sum_{k=1}^N n_k = n. \quad (3.4)$$

Просмотреть все варианты разбиения уже для числа $n > 100$ практически нереально! Если же сформулировать задачу компоновки в терминах математического программирования, то можно уменьшить число просматриваемых вариантов компоновки.

Итак, пусть требуется распределить n компонентов электронной схемы между N блоками таким образом, чтобы суммарное число связей между блоками было минимально.

Введем вектор $X = \{x_{i,k}\}$ переменных проектирования, где $x_{i,k}$ – элементы вектора X , $i = \overline{1, n}$; $k = \overline{1, N}$;

$$x_{i,k} = \begin{cases} 1, & \text{если компонент } d_i \text{ включается в подмножество } D_k; \\ 0, & \text{в противном случае.} \end{cases}$$

Пусть функция качества $F(x)$ характеризует общее число связей между подмножествами разбиения множества D :

$$D = \{D_k\}, \text{ для } k = \overline{1, N}.$$

Тогда задача компоновки запишется:

$$F(x) = \sum_{j=1}^{n-1} \sum_{i=j+1}^n \sum_{k=1}^{N-1} \sum_{l=k+1}^N x_{j,k} \cdot x_{i,l} \cdot \pi_{i,j} \rightarrow \min \quad (3.5)$$

при условиях:

$$\sum_{k=1}^N x_{i,k} = 1, \quad i = \overline{1, n}; \quad (3.6)$$

$$\sum_{i=1}^n x_{i,k} = n_k, \quad k = \overline{1, N}; \quad (3.7)$$

$$\sum_{i=1}^n V_i^{(S)} \cdot x_{i,k} \leq V_S^{(k)}; \quad (3.8)$$

$$x_{i,k} = (0;1), \quad (3.9)$$

где π_{ij} – число связей между компонентами d_i и d_j ;

$V_i^{(S)}$ – значение параметра S для компонента d_i ;

$V_S^{(k)}$ – ограничение по параметру S , накладываемое на подмножество D_k ;

S – может быть любой параметр, подчиняющийся свойству аддитивности: объем, масса, энергоемкость, стоимость и т. п.

Таким образом, условие (3.8) означает, что суммарное значение параметра S не должно превышать некоторого критического значения.

В задаче (3.5)–(3.9) условия (3.6), (3.7) соответственно означают, что каждый компонент может быть отнесен только к одному из подмножеств D_k и в каждом подмножестве D_k может содержаться компонентов не более чем заданное число n_k .

Задача размещения

Высокая плотность размещения элементов ЭВА создает большие трудности при реализации соединений между ними. В этой связи задача размещения элементов на плоскости определяет быстроту и качество трассировки. Оптимальное размещение элементов обеспечивает повышение надежности проектируемого устройства, минимизацию наводок, задержек сигналов, уменьшение общей длины соединений и т. п.

Формально задача размещения заключается в определении оптимального варианта расположения элементов на плоскости в соответствии с введенным критерием. Например, с минимальной взвешенной длиной соединений.

В общем случае требуется найти размещение компонентов d_1, \dots, d_n на множестве q_1, q_2, \dots, q_m ($m \leq n$) позиций монтажного пространства, при котором суммарная длина соединений между компонентами была бы минимальной.

Введем булевы переменные:

$$x_{i,k} = \begin{cases} 1, & \text{если компонент } d_i \text{ назначается на позицию } q_k; \\ 0, & \text{в противном случае.} \end{cases}$$

Тогда математическая модель задачи размещения может быть записана:

$$F(x) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^m \sum_{s=1}^m x_{i,k} \cdot x_{j,s} \cdot l_{k,s} \cdot p_{i,j} \rightarrow \min \quad (3.10)$$

при условиях:

$$\sum_{k=1}^m x_{i,k} = 1, \quad i = \overline{1, n}; \quad (3.11)$$

$$\sum_{i=1}^n x_{i,k} \leq 1, \quad k = \overline{1, m}; \quad (3.12)$$

$$x_{i,k} = (0; 1), \quad (3.13)$$

где $l_{k,s}$ – расстояние между позициями q_k, q_s ;

$p_{i,j}$ – число связей между компонентами $d_i; d_j$.

Условия (3.11), (3.12) соответственно означают, что каждый компонент может быть размещен только на одно посадочное место и каждое посадочное место может быть закреплено только за одним компонентом.

Задача трассировки

Задача трассировки встречается при конструировании печатных плат; разработке систем водоснабжения, электроснабжения и т. д.

Трассировка соединений является, как правило, заключительным этапом конструкторского проектирования ЭВА и состоит в определении линий, соединяющих эквипотенциальные контакты элементов и компонентов, составляющих проектируемое устройство.

Задача трассировки – одна из наиболее трудоемких в общей проблеме автоматизации проектирования ЭВА. С математической точки зрения трассировка – наисложнейшая задача выбора из огромного числа вариантов оптимального решения.

Основная задача трассировки формулируется следующим образом: по заданной схеме соединений проложить необходимые проводники на плоскости (плате, типовом элементе замены, кристалле и т. п.), чтобы реализовать заданные электрические соединения с учетом заранее заданных ограничений. Основными являются ограничения на ширину проводников и минимальное расстояние между ними.

Исходной информацией для решения задачи трассировки соединений обычно являются список цепей, параметры конструкции элементов и коммутационного поля, а также данные по размещению элементов.

Критериями трассировки, наиболее часто используемыми для оценки качества решения задачи трассировки, могут быть процент реализованных соединений, суммарная длина проводников, число монтажных слоев, число межслойных переходов, минимальная область трассировки и др.

Задача трассировки всегда имеет топологический и метрический аспекты. Топологический аспект связан с выбором допустимого пространства расположения отдельных фрагментов соединений без фиксации их конкретного месторасположения при ограничениях на число пересечений и слоев. Метрический аспект предполагает учет конструктивных размеров элементов, соединений и коммутационного поля, а также метрических ограничений на трассировку.

Рассмотрим одну из постановок задачи трассировки – задачу построения связывающих сетей минимальной длины для цепей α_k .

Пусть U_k – множество точек, соединяемых по электрической цепи α_k ; $|U_k| = n_k$, где каждому элементу U_k соответствует одна точка в монтажном пространстве.

Введем понятие трассы.

Трасса – множество связанных отрезков, соединяющих точки электрической цепи.

Определим переменную проектирования x_{ij} :

$$x_{ij} = \begin{cases} 1, & \text{если ребро } (i, j) \text{ длиной } l \text{ включается в связывающую сеть;} \\ 0, & \text{в противном случае.} \end{cases}$$

Здесь x_{ij} – булева переменная.

Тогда математическая модель задачи трассировки запишется:

$$F(x) = \sum_{i=1}^{n_{k-1}} \sum_{j=i+1}^{n_k} l_{i,j} \cdot x_{i,j} \rightarrow \min \quad (3.14)$$

при условиях:

$$\sum_{j=1}^{n_k} x_{i,j} \leq K_0, \quad i = \overline{1, n_k}; \quad (3.15)$$

$$x_{i,j} = (0;1), \quad (3.16)$$

где K_0 – максимально допустимое число соединений в одной точке.

Условие (3.15) означает, что в одной точке не могут соединяться ребра в количестве более заданного числа K_0 .

3.3 Оптимальное проектирование РЭС на основе решения задачи линейного программирования

Основные идеи линейного программирования возникли во время Второй мировой войны в связи с поиском оптимальных стратегий при ведении военных операций и в дальнейшем были использованы для решения многих задач из области управления, торговли и техники. В частности, линейное программирование широко используется для проектирования радиоэлектронных средств и систем, конструкций и технологических процессов производства радиоаппаратуры.

Задачи, в которых отыскивается минимум или максимум некоторой функции, зависящей от многих переменных при наличии ограничений на эти переменные, объединяются под общим названием *задачи математического программирования*. Частным случаем этих задач, когда минимизируемая (максимизируемая) функция и ограничения линейно зависят от переменных, являются задачи линейного программирования. В дальнейшем функцию, максимум или минимум которой необходимо найти, будем называть *целевой функцией*. При проектировании радиоэлектронных средств целевая функция будет характеризовать качество работы, стоимость аппаратуры либо иные характеристики, зависящие от параметров составляющих компонентов, оптимальные значения которых в результате решения задачи необходимо найти. Ограничения же, присутствующие в задаче, представляют систему соотношений, сужающих допу-

стимулю область изменения параметров компонентов при решении задачи оптимизации.

Таким образом, задача линейного программирования заключается в нахождении вектора переменных $\bar{V} = |v_1, v_2, \dots, v_n|^T$, минимизирующего линейную целевую функцию

$$F = \sum_{i=1}^n c_i \cdot v_i \rightarrow \min$$

при наличии ограничений-равенств:

$$a_{11}v_1 + a_{12}v_2 + \dots + a_{1n}v_n = A_1,$$

.....

(3.17)

$$a_{k1}v_1 + a_{k2}v_2 + \dots + a_{kn}v_n = A_k,$$

и линейных ограничений-неравенств:

$$b_{11}v_1 + b_{12}v_2 + \dots + b_{1n}v_n \geq B_1,$$

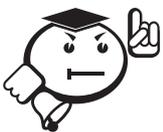
.....

(3.18)

$$b_{r1}v_1 + b_{r2}v_2 + \dots + b_{rn}v_n \geq B_r$$

и общей неотрицательности элементов $v_i > 0$, $i = 1, \dots, n$. Значения A_i , B_i , a_{ij} , b_{ij} , c_j , где $i, j = 1, \dots, n$, полагаются известными.

Это общий вид задачи линейного программирования, часто называемый основной задачей. В математике различают несколько видов формулировки задачи линейного программирования: задача оптимального планирования, транспортная задача, задача о назначениях и т. д. В большинстве случаев они сводятся к основной задаче, поэтому в дальнейшем будем рассматривать только этот вариант задачи линейного программирования.



В отличие от рассмотренной задачи (3.17)–(3.18) в практических задачах могут встречаться ограничения только в виде равенства, либо только в виде неравенства, переменные могут принимать отрицательные значения, а целевая функция может максимизироваться. Такое разнообразие форм записи задачи линейного программирования затрудняет разработку общих алгоритмов ее решения, поэтому при решении задачу сводят к стандартной форме, в которой целевая функция минимизируется, а все ограничения заданы в виде равенств с положительно определенными переменными.

Привести задачу линейного программирования к стандартной форме возможно, если использовать следующие правила:

а) максимизация целевой функции $F = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$ равносильна минимизации функции $F' = -c_1 v_1 - c_2 v_2 - \dots - c_n v_n$;

б) ограничение в виде неравенства $b_{11} v_1 + b_{12} v_2 + \dots + b_{1n} v_n \geq B_1$ может быть приведено к стандартной форме в виде равенства введением новой неотрицательной переменной $v_{n+1} > 0$:

$$b_{11} v_1 + b_{12} v_2 + \dots + b_{1n} v_n - v_{n+1} = B_1;$$

по аналогии ограничение-неравенство вида

$$b_{21} v_1 + b_{22} v_2 + \dots + b_{2n} v_n \leq B_2$$

приводится к стандартной форме введением переменной $v_{n+2} > 0$:

$$b_{11} v_1 + b_{12} v_2 + \dots + b_{1n} v_n + v_{n+2} = B_2,$$

где новая переменная $v_{n+2} > 0$ также неотрицательна. Введенные таким образом дополнительные переменные v_{n+1} и v_{n+2} будут входить в минимизируемую целевую функцию F с нулевыми коэффициентами, чтобы они не влияли на ее оптимальное значение;

в) если некоторая переменная по условиям исходной задачи может принимать любые значения, то ее можно представить в виде $v_i = v_{n+3} - v_{n+4}$ с последующей подстановкой в целевую функцию и ограничением новых неотрицательных переменных v_{n+3} и v_{n+4} .

.....

Таким образом, приведение задачи линейного программирования к стандартной форме может потребовать введения дополнительных неотрицательных переменных, увеличивающих размерность задачи. В окончательном виде стандартная форма задачи линейного программирования в матричной записи может быть представлена следующим образом:

$$F = \bar{C}^T \bar{V} \rightarrow \min; \quad (3.19)$$

$$A : \bar{V} = \bar{A}; \quad (3.20)$$

$$\bar{V} \geq 0;$$

где $\bar{C} = |c_1 \dots c_n|^T$; $\bar{V} = |v_1 \dots v_n|^T$; $\bar{A} = |A_1 \dots A_m|^T$, $A = [a_{ij}]$ – матрица коэффициентов в левой части уравнений-ограничений размера $m \times n$, причем в \bar{V} наряду с исходными переменными будут входить дополнительные, появившиеся в результате приведения задачи к стандартной форме. Коэффициенты при дополнительных переменных будут входить и в матрицу A .



Пример

Рассмотрим примеры постановки задачи линейного программирования, возникающие при проектировании радиоэлектронных устройств.

Пусть проектируется система командной радиосвязи, включающая n различных видов связи (коротковолновые и ультракоротковолновые радиостанции, радиорелейные средства связи, спутниковая связь и т. д.). Число параллельно работающих каналов в каждом виде связи будет равно v_i , $i = 1, \dots, n$. Каждый из видов связи имеет свои преимущества и недостатки при работе в различных вариантах помеховой обстановки, различна и стоимость организации одного канала для каждого вида связи c_i .

Проектируемая система должна устойчиво работать при различных уровнях помех в эфире и в каждом случае должна обеспечивать вероятность правильной передачи сообщения не хуже, чем p_j , $j = 1, \dots, m$.

Вероятность правильной передачи сообщения каналом каждого i -го вида связи при j -м варианте помеховой обстановки в эфире известна и равна p_{ij} . Вероятность того, что сообщение не будет принято абонентом при передаче его одним каналом, соответственно составит $1 - p_{ij}$. Если же это сообщение передается v_i некоррелированными каналами, то вероятность его непередачи составит $(1 - p_{ij})^{v_i}$. Поскольку по условиям задачи одно и то же сообщение должно передаваться не одним, а n видами связи, то полная вероятность, что сообщение не дойдет до абонента, в приближении статистической независимости каналов и видов связи, будет равна $(1 - p_{1j})^{v_1} (1 - p_{2j})^{v_2} \dots (1 - p_{nj})^{v_n}$.

Это позволяет для m вариантов помеховой обстановки в эфире записать систему неравенств-ограничений, исходя из правильной передачи сообщения при каждом j -м уровне помех:

$$1 - (1 - p_{11})^{v_1} (1 - p_{12})^{v_2} \dots (1 - p_{1n})^{v_n} \geq p_1,$$

.....

$$1 - (1 - p_{m1})^{v_1} (1 - p_{m2})^{v_2} \dots (1 - p_{mn})^{v_n} \geq p_m.$$

После логарифмирования и выполнения несложных математических преобразований полученная выше система неравенств-ограничений может быть приведена к следующему виду:

$$b_{11}v_1 + b_{12}v_2 + \dots + b_{1n}v_n \leq B_1,$$

.....

$$b_{m1}v_1 + b_{m2}v_2 + \dots + b_{mn}v_n \leq B_m,$$

где $b_{ij} = \ln(1 - p_{ij})$, $B_j = \ln(1 - p_j)$, $i = 1, \dots, n$, $j = 1, \dots, m$.

Последние соотношения позволяют сформулировать задачу проектирования оптимальной системы радиосвязи как задачу определения количества каналов в каждом виде связи, минимизирующего суммарную стоимость средств связи при обеспечении вероятностей передачи сообщений не хуже заданных для вариантов помеховой обстановки в эфире:

$$F = c_1v_1 + c_2v_2 + \dots + c_nv_n \rightarrow \min;$$

$$b_{11}v_1 + b_{12}v_2 + \dots + b_{1n}v_n \leq B_1;$$

.....

$$b_{m1}v_1 + b_{m2}v_2 + \dots + b_{mn}v_n \leq B_m;$$

$$v_i \geq 0, \quad i = 1, \dots, n.$$

Очевидно, что эта формулировка задачи проектирования оптимальной системы радиосвязи полностью эквивалентна основной задаче линейного программирования, определяемой соотношениями (3.19). Возможны и другие варианты формулировки этой задачи. Например, если в качестве коэффициентов целевой функции c_j выбрать энергетические затраты на организацию одного канала каждого j -го вида связи, то получим задачу линейного программирования, решение которой позволит определить количество каналов в каждом виде связи, при которых при всех уровнях помех сообщение с заданной вероятностью будет передано с минимальными энергетическими затратами.

.....



Пример

В качестве еще одного примера рассмотрим сведение к основной задаче линейного программирования задачи покрытия функциональной схемы цифрового устройства заданным набором (серией) микросхем.

Пусть имеется функциональная схема цифрового устройства, каждый элемент которой выполняет какую-нибудь логическую функцию И, ИЛИ, НЕ-И и т. д. Каждой логической функции φ_i можно в бинарное отношение сопоставить число k_i , $i = 1, \dots, m$, представляющее количество логических элементов φ_i в функциональной схеме цифрового устройства.

Задан также набор микросхем Ψ_j , $j = 1, \dots, n$, на базе которого необходимо реализовать цифровое устройство. Каждая микросхема реализует одну или несколько логических функций φ_i , следовательно, каждой j -й микросхеме можно сопоставить ряд чисел h_{ij} , характеризующих, сколько и каких логических функций φ_i реализуется в микросхеме j -го типа. Для каждой микросхемы h_{ij} образуют строку из n чисел, все h_{ji} образуют матрицу H размером $m \times n$.

Пусть нам известна стоимость каждой микросхемы c_j . При таких условиях задача реализации функциональной схемы цифрового устройства с помощью заданного набора микросхем может быть сформулирована как задача линейного программирования, минимизирующая оптимальным выбором количества микросхем каждого типа v_j стоимость цифрового устройства:

$$F = \sum_{i=1}^n c_i \cdot v_i \rightarrow \min;$$

$$h_{11}v_1 + h_{12}v_2 + \dots + h_{1n}v_n \geq k_1;$$

.....

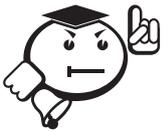
$$h_{m1}v_1 + h_{m2}v_2 + \dots + h_{mn}v_n \geq k_m;$$

$$v_j \geq 0, j = 1, \dots, n.$$

где v_j – количество микросхем j -го типа, используемых для реализации цифрового устройства, заданного функциональной схемой.

Если все $c_j = 1$, то записанная выше задача линейного программирования представляет собой задачу минимизации числа микросхем, с помощью которых реализуется цифровое устройство.

Особенность задачи состоит в том, что искомые переменные представляют собой целые числа. В простейшем случае решение, полученное в непрерывном пространстве, может быть округлено до ближайшего бóльшего целого, обеспечивая некоторую избыточность покрытия.



К подобным задачам могут быть сведены задачи оптимального проектирования частотно-избирательных фильтров, размещения микросхем на печатной плате, трассировки проводников в каналах и ряд других. К задачам линейного программирования сводятся и некоторые алгоритмы решения задачи нелинейного программирования с ограничениями.

Симплекс-метод и основные утверждения линейного программирования

Классический метод отыскания экстремумов функции многих переменных, заключающийся в дифференцировании целевой функции по всем переменным, приравнивании нулю производных и решении полученной системы уравнений относительно оптимальных значений переменных, неприемлем для решения задачи линейного программирования, так как функция F линейно зависит от переменных \bar{V} и её производные по всем переменным v_i постоянны и нигде не обращаются в нуль.

Для решения задач линейного программирования Г. Данцигом была разработана специальная вычислительная процедура, называемая симплекс-методом и основанная на ряде теоретических утверждений линейного программирования. Для лучшего понимания этих утверждений и существа симплекс-метода рассмотрим геометрическую интерпретацию задачи линейного программирования и метода ее решения.

Пусть требуется минимизировать функцию

$$F = -3v_1 - 4v_2 \rightarrow \min$$

при наличии следующих ограничений:

$$v_1 + v_2 \leq 20;$$

$$-v_1 + 4v_2 \leq 20;$$

$$v_1, v_2 \geq 0.$$

Так как целевая функция и ограничения зависят всего лишь от двух переменных, их можно представить в виде прямых на плоскости v_1 O v_2 . Эта графическая интерпретация задачи показана на рисунке 3.6, где определяемые прямыми ограничения показаны сплошными линиями, а линии равного уровня целевой функции – пунктирными. Стрелки на каждой граничной прямой показывают, с какой стороны прямой выполняется ограничение.

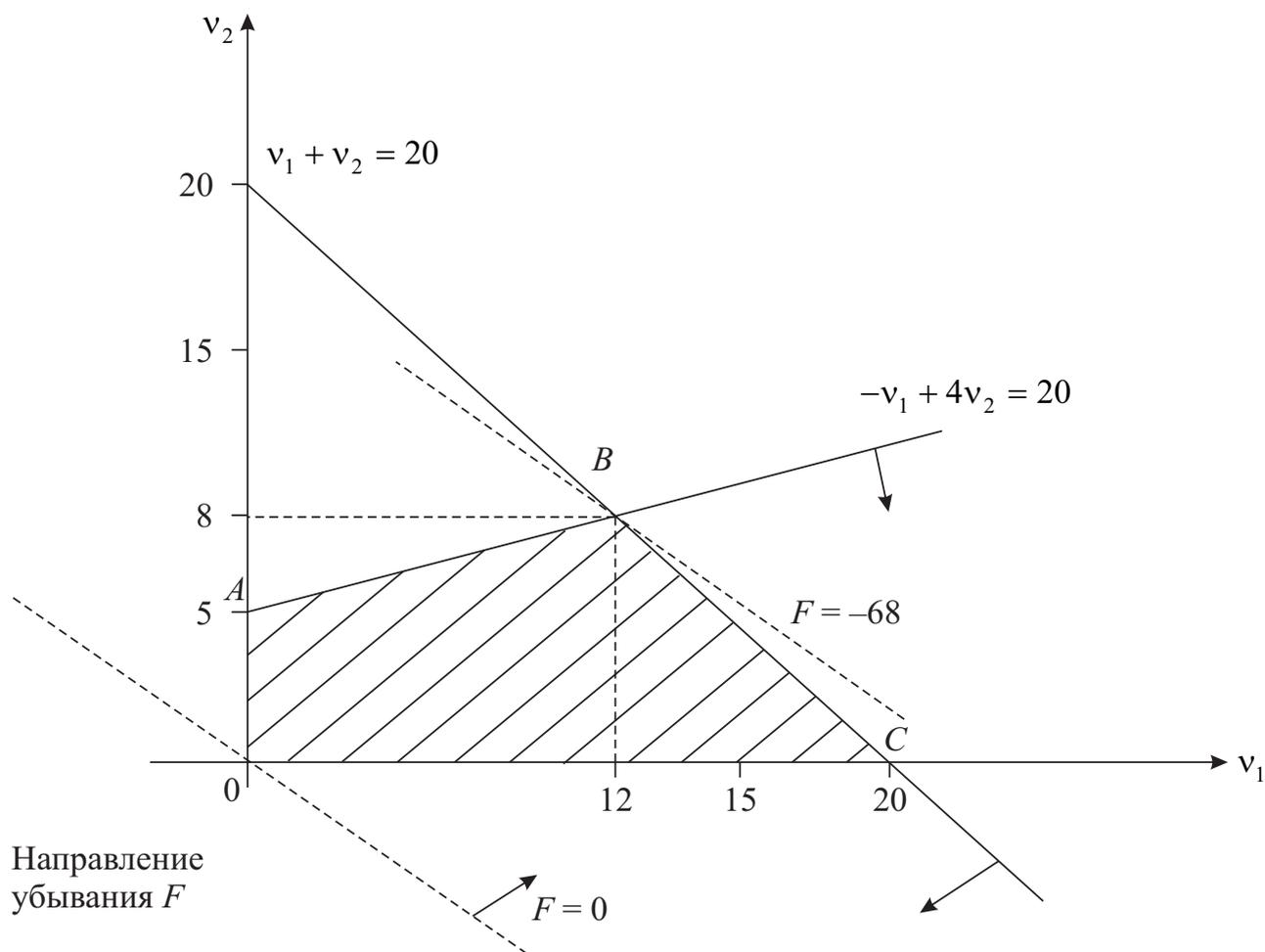


Рис. 3.6 – Геометрическая интерпретация задачи линейного программирования

Для точек внутри заштрихованной области $OABC$ и на ее границах выполняются ограничения задачи. Эта область называется допустимой, а точки внутри области и на её границах представляют собой допустимые решения за-

дачи. Допустимых решений множество, и необходимо найти одно из них, минимизирующее целевую функцию F .

Штриховыми линиями на рисунке 3.6 изображены прямые $F' = -3v_1 - 4v_2 = 0$ и $F'' = -3v_1 - 4v_2 = -68$. Эти прямые параллельны и представляют собой две линии равного уровня F со значениями 0 и -68 соответственно. Целевая функция убывает в направлении вектора, обратного градиенту

$$-\begin{vmatrix} \frac{\partial F}{\partial v_1} \\ \frac{\partial F}{\partial v_2} \end{vmatrix} = \begin{vmatrix} 3 \\ 4 \end{vmatrix},$$

показанному на рисунке 3.6 пунктирной стрелкой.

Линией уровня F с наименьшим значением целевой функции, имеющей хотя бы одну общую точку с областью допустимых значений, является прямая, проходящая через точку B с координатами $(12,8)$. Минимальное значение целевой функции равно -68 . Заметим, что оптимальное решение задачи достигается в вершине допустимой области, а сама область является выпуклой, причем решение соответствует строгому выполнению равенств для ограничений задачи, и число переменных, не равных нулю, равно числу ограничений в задаче. Это эквивалентно равенству нулю двух добавочных переменных, которые необходимо было бы ввести в ограничения задачи для приведения ее к стандартной форме. Такие решения, в которых не равны нулю столько переменных, сколько в стандартной форме задачи ограничений-равенств, а остальные переменные равны нулю, называются *опорными решениями* задачи линейного программирования; они всегда соответствуют одной из вершин многоугольника ограничений. Переменные, не равные нулю, называются допустимым *базисным* решением задачи, или просто *базисом*, остальные переменные – небазисными или нулевыми.

Эти выводы можно распространить и на случай многомерной задачи. В случае n переменных каждое уравнение-ограничение представляет собой плоскость в n -мерном пространстве. Фигура, образованная этими плоскостями, образует область допустимых значений переменных и называется симплексом. Для нашего частного случая двумерной задачи симплекс представляет собой многоугольник на плоскости.

Одно из основных утверждений линейного программирования показывает, что для того, чтобы задача линейного программирования имела оптималь-

ное решение, симплекс должен быть выпуклым, т. е. допустимая область определения переменных является выпуклым множеством.



Опорные решения задачи линейного программирования соответствуют вершинам симплекса ограничений, в которых не равны нулю и положительны m переменных по числу уравнений-ограничений, образующих допустимое базисное решение, и равны нулю остальные $n - m$ переменных, образующих небазисное решение.

Оптимальное решение задачи линейного программирования, если оно существует, является одним из опорных решений.

Эти утверждения подтверждаются рассмотренным ранее графическим примером. Оптимальное решение задачи было получено в одной из вершин симплекса ограничений и представляет собой одно из опорных решений.



Выводы

Приведенные утверждения свидетельствуют, что при поиске оптимального решения достаточно вычислить только все опорные решения и определить значение целевой функции в них. Однако с возрастанием числа переменных и ограничений эти вычисления оказываются очень трудоемкими. Более рациональное решение обеспечивает симплекс-метод, который позволяет, отправляясь от какого-либо допустимого базисного решения, т. е. вершины симплекса, перейти к соседнему базисному решению, в котором значение целевой функции будет меньше, причем остальные варианты не вычисляются. Оптимальное решение задачи линейного программирования симплекс-методом получается за конечное число шагов.



Пример

Рассмотрим процедуру симплекс-метода подробнее.

Работа симплекс-метода начинается с выбора опорного решения \bar{V}_0 , содержащего m базисных ненулевых переменных и $n - m$ небазисных нулевых переменных v_i , $i = 1, \dots, n$. Для упрощения записи положим, что базисные пере-

эту переменную целесообразно включить в число базисных и выбрать ее максимально возможной, чтобы сильнее уменьшить значение целевой функции. При этом одну из базисных переменных необходимо будет перевести в нулевую, чтобы вновь полученное решение соответствовало вершине симплекса ограничений. Это будет переменная v_k , $k \in [1, \dots, m]$, которая быстрее всего достигнет нулевого значения при увеличении v_l от нуля. Другими словами, $v_k = D_k - \alpha_{kl} v_l$ для всех $k = 1, \dots, m$ стремится к нулю быстрее при увеличении значения v_l . Это означает, что $\alpha_{kl} > 0$ и значение $D_k / \alpha_{kl} \geq 0$ минимально для всех базисных переменных: $D_k / \alpha_{kl} = \min \{ D_i / \alpha_{il}, i = 1, \dots, m \}$. Равенство нулю v_k позволяет определить максимально возможное значение вводимой в базис переменной v_l следующим образом:

$$v_k = 0 = D_k - \alpha_{kl} v_l; \quad v_l = D_k / \alpha_{kl}. \quad (3.23)$$

Как следует из (3.21), изменение переменной v_l , ранее относящейся к допустимому базисному решению, приведет к изменению остальных базисных переменных и теперь улучшенный допустимый базис определится как

$$\bar{V}_6 = \begin{cases} D_i - a_{il} (D_k / a_{kl}), & i = 1, \dots, m, i \neq k, i \neq l, \\ v_i = D_k / a_{kl}, & i \neq l. \end{cases}$$

Нулевыми переменными в этом новом опорном решении, соответствующем новой вершине симплекса, будут

$$\bar{V}_n = \begin{cases} v_i, & i = k, i = m + 1, \dots, n, i \neq l, \\ v_i, & i = k, \end{cases}$$

целевая функция F будет равна

$$F = \sum_{i \in V_n} c_i v_i. \quad (3.24)$$



При этой процедуре может понадобиться возврат к исходной точке и выбор новой вводимой в базис переменной из числа входящих в целевую функцию с отрицательными коэффициентами, если значение F окажется больше F_0 . Если же $F < F_0$, то система уравнений-ограничений переформируется относительно новых ба-

численных переменных и описанный процесс продолжается итеративно до тех пор, пока все β_i не станут положительными.



Пример

Рассмотрим в качестве примера использования симплекс-метода решенную ранее графически задачу линейного программирования. В стандартной форме эта задача может быть записана следующим образом:

$$F = -3v_1 - 4v_2 \rightarrow \min$$

при условиях:

$$v_1 + v_2 + v_3 = 20,$$

$$-v_1 + 4v_2 + v_4 = 20,$$

$$v_1 \dots v_4 \geq 0.$$



Здесь v_3, v_4 — дополнительные переменные, введенные в задачу для приведения ее к стандартной форме. Начальное допустимое базисное решение очевидно:

$$\bar{V}_6 = \begin{vmatrix} v_3 = 20 \\ v_4 = 20 \end{vmatrix},$$

небазисные переменные:

$$\bar{V}_H = \begin{vmatrix} v_1 = 0 \\ v_2 = 0 \end{vmatrix}.$$

В соответствии с (3.21) выразим базисные переменные v_3 и v_4 через нулевые:

$$v_3 = 20 - v_1 - v_2,$$

$$v_4 = 20 - v_1 - 4v_2.$$

Целевая функция, записанная подобно (3.22) через нулевые переменные, не изменяет вида:

$$F_0 = -3v_1 - 4v_2.$$



Поскольку коэффициенты в целевой функции F при нулевых переменных v_1 и v_2 не положительны, то начальное допустимое базисное решение не оптимально. Максимальный по модулю отрицательный коэффициент в целевой функции будет у переменной v_2 , поэтому на следующей итерации ее необходимо ввести в базис. С другой стороны, отношение D_4/α_{42} при $\alpha_{42} > 0$ минимально по модулю для базисной переменной v_4 , поэтому её при переходе к новому опорному решению необходимо вывести из базиса. Вводимая в базис переменная определится соотношением (3.23): $v_4 = 0 = 20 + v_1 - 4v_2$, откуда следует $v_2 = 5$.

Определим, используя (3.24), новые базисные переменные v_3 и v_2 и новую целевую функцию:

$$v_3 = 15,$$

$$v_2 = 5,$$

$$F = -20.$$

Так как F уменьшилась по сравнению с F_0 , то новое допустимое базисное решение позволяет сделать еще один шаг симплекс-методом, для чего выразим новые базисные переменные v_2 и v_3 и целевую функцию через нулевые переменные v_1 и v_4 :

$$v_2 = 5 + jv_1 - jv_4,$$

$$v_3 = 15 + 5/4v_1 - jv_4,$$

$$F = -20 - 4v_1 + v_4.$$



Выводы

Из последних соотношений видно, что новое опорное решение также не является оптимальным, так как нулевая переменная v_i входит в целевую функцию с отрицательным коэффициентом и на следующем шаге симплекс-метода переменная v_i должна быть введена в базис. Базисная переменная v_3 , напро-

тив, при переходе к новому опорному решению должна быть из базиса выведена, так как $\alpha_{31} > 0$ и отношение D_3/α_{31} минимально по модулю для базисных переменных. Значение вводимой в базис переменной v_1 определится из равенства нулю базисной переменной:

$$v_3 = 15 - 5/4 v_1 = 0, \text{ откуда } v_1 = 12.$$

Подобно предыдущим шагам, определим новые базисные переменные и улучшенное значение целевой функции:

$$v_1 = 12, v_2 = 8, F = -68.$$

Выразим базисные переменные v_1 и v_2 и функцию качества F через нулевые переменные v_3 и v_4 :

$$v_1 = 12 - 4/5 v_3 + 1/5 v_4,$$

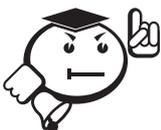
$$v_2 = 8 - 1/5 v_3 - 1/5 v_4,$$

$$F = -68 + 16/5 v_3 + 1/5 v_4.$$



Выводы

В получившемся опорном решении нулевые переменные v_3 и v_4 входят в целевую функцию с отрицательными коэффициентами, что свидетельствует об его оптимальности. Решение, полученное симплекс-методом, полностью совпадает с решением, полученным ранее графическим путём. В качестве упражнения можно проверить, что при выборе других начальных опорных решений процедура симплекс-метода приведет к тому же оптимальному решению.



Этот итеративный процесс обычно представляется в виде так называемых симплекс-таблиц, упрощающих вычисление коэффициентов при переменных в уравнениях-ограничениях и целевой функции. Однако при большом числе переменных такое решение требует очень больших затрат машинного времени и неудобно для реализации на ЭВМ, поэтому в программных реализациях используется модифицированный симплекс-метод, или метод жорданова исключения.

Методы решения целочисленной задачи линейного программирования

В начале п. 3.3 приводились примеры формулировки задач линейного программирования, в которых переменные могут принимать только целочисленные значения. Такие задачи являются частным случаем задач дискретного программирования, когда переменные принимают определенный ряд дискретных значений, не обязательно целочисленных. Дискретные задачи математического программирования образуют обширный класс нерегулярных задач, в которых область допустимых решений является невыпуклой и несвязной и определяется ограничениями двух типов: обычными регулярными и условиями дискретности. В дальнейшем ограничимся только целочисленными задачами линейного программирования, которые могут быть сформулированы следующим образом: минимизировать целевую функцию

$$F = \bar{C}^T \cdot \bar{V} \rightarrow \min$$

при наличии регулярных ограничений $A \bar{V} = \bar{A}$ и ограничений

$$D_j \leq v_j \leq u_j, \quad j = 1, \dots, r, \quad (3.25)$$

где v_j – целые числа, D_j и u_j – нижняя и верхняя целочисленные границы j -го индекса переменной v_j , вектор \bar{C}^T – включает коэффициенты при базисных и нулевых переменных.

Если $r < n$, где n – общее число переменных в задаче, то задачу называют частично целочисленной, если же $r = n$, то задачу называют полностью целочисленной.



Невыпуклость и несвязность области допустимых решений задачи (3.25) делает невозможным применение стандартных алгоритмов линейного программирования. Поэтому большинство известных алгоритмов решения целочисленных задач основано на регуляризации задачи, т. е. многократном решении непрерывных задач с последующей дискретизацией переменных за счет введения новых ограничений. Рассмотрим один из методов решения задач целочисленного программирования – *методы ветвей и границ*, получивший наибольшее распространение для решения комбинаторных задач.



Основная идея **метода ветвей и границ** заключается в осуществлении на каждом шаге решения задачи ветвления по одной из базисных нецелочисленных переменных путем разбиения задачи на две, в одну из которых вводится дополнительное ограничение

$$D_j \leq v_j \leq N_j, \quad (3.26)$$

а в другую – ограничение

$$N_{j+1} \leq v_j \leq u_j, \quad j = 1, \dots, m, \quad (3.27)$$

где N_j – целое число, m – число ограничений.



Каждая из этих задач решается как обычная задача линейного программирования в непрерывном пространстве изменения переменных с игнорированием требований их целочисленности. Этот процесс повторяется для различных j и различных N_j до тех пор, пока все требуемые переменные не окажутся целочисленными. Очевидно, что эффективность процесса зависит и от того, по какой переменной v_j осуществляется ветвление и как выбирается величина целочисленного значения средней границы переменной.



Для выбора значения N_j может быть использован следующий прием: перед началом ветвления по переменной v_j задача (3.25) решается в непрерывном пространстве и определяется оптимальное значение всех переменных, в том числе и значение v_j . Затем полагается $N_j = \lfloor v_j \rfloor_{\text{опт}}$, где $\lfloor v_j \rfloor_{\text{опт}}$ – целая часть оптимального значения переменной v_j при решении в непрерывном пространстве.

Выполняя процесс ветвления по переменной, формируют левую задачу линейного программирования добавлением в задачу (3.25) ограничения (3.26):

$$\begin{aligned}
 F &= \bar{C}^T \cdot \bar{V} \rightarrow \min, \\
 \text{а } \bar{V} &= \bar{A}, \\
 D_j &\leq v_j \leq \left[v_j \right]_{\text{опт}} = N_j.
 \end{aligned}
 \tag{3.28}$$

По аналогии добавлением в задачу (3.25) ограничения (3.27) формируется правая задача линейного программирования:

$$\begin{aligned}
 F &= \bar{C}^T \cdot \bar{V} \rightarrow \min, \\
 \text{а } \bar{V} &= \bar{A}, \\
 \left[v_j \right]_{\text{опт}} + 1 &= N_j + 1 \leq v_j \leq u_j.
 \end{aligned}
 \tag{3.29}$$

Если задачи (3.28) и (3.29) имеют решения, то они решаются в непрерывном пространстве. Задача, в которой в результате решения целевая функция окажется больше, отбрасывается. Остающиеся задачи образуют главный список задач. Ветвление каждой задачи из главного списка ведет к формированию двух новых левой и правой задач по одной из базисных переменных, не удовлетворяющих требованиям целочисленности. Последовательность главных задач образует древовидную структуру, пример которой показан на рисунке 3.7. Ограничения каждой из двух новых задач, полученных в результате ветвления, состоят из ограничений породившей их задачи и новых ограничений на переменную, по которой осуществляется ветвление.



.....
 Процесс ветвления может закончиться по следующим причинам.

1. Получено целочисленное допустимое решение, если только оно существует.
2. Решение левой и правой задач оказывается невозможным. Это становится все более вероятным по мере продолжения процесса ветвления, когда все большее число ограничений вида (3.26) и (3.27) добавляется к уже имеющимся.
3. Значение целевой функции оказывается большим, чем значение, отвечающее наилучшему из найденных ранее целочисленных решений.

Добавление ограничений посредством ветвления не может улучшить значения целевой функции, так что дальше продолжать процесс ветвления нецелесообразно.

Скорость сходимости процесса ветвления зависит не только от выбора граничных значений переменной, по которой осуществляется процесс ветвления, но и от способа выбора самой переменной v_j . Обычно выбор переменной для ветвления осуществляется с помощью метода штрафов.

.....

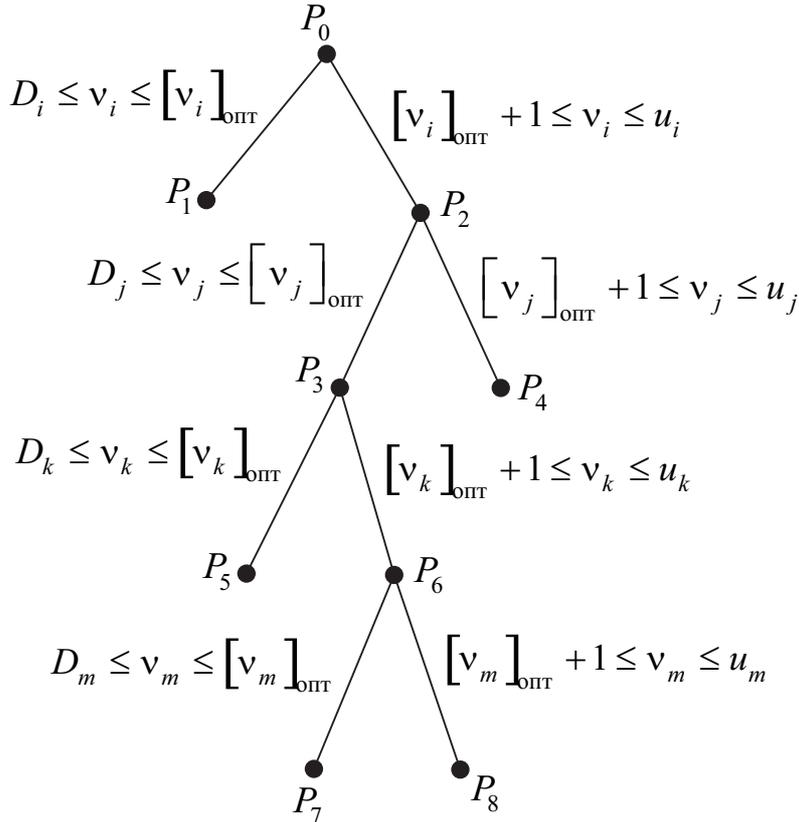


Рис. 3.7 – Древоподобная структура ветвления в методе ветвей и границ

Рассмотрим j -ю базисную переменную в оптимальном решении какой-либо порождающей задачи. Выражение для v_j через нулевые переменные имеет вид

$$v_j = D_j - \sum_{i=1}^{n-m} a_{ij} v_i, \text{ где } v_i \in \bar{V}_n.$$

Так как оптимальное решение для v_i получено в непрерывном пространстве, то значение D_j , которому численно равна переменная $v_{j \text{ опт}}$, состоит из целой и дробной частей: $D_j = [D_j] + d_j$. Если переменная v_j выбрана для

ветвления, то наложение ограничения $\lceil D_j \rceil + 1 \leq v_j \leq u_j$ должно заставить v_j возрасти на величину $1 - d_j$. Это может произойти за счет возрастания небазисной переменной v_l , имеющей максимальное и положительное значение коэффициента α_{lj} на величину $(1 - d_j) / \alpha_{lj}$, чтобы выполнялось равенство $v_j = 1 + \lceil D_j \rceil$. Изменение небазисной переменной v_l приводит к ухудшению функции качества, причём минимальное значение, на которое возрастает функция качества, определится выражением

$$S_{ij} = \min \left\{ (1 - d_j) \cdot (\beta_l / \alpha_{lj}) \right\}, \alpha_{lj} > 0, l \in V_n, \quad (3.30)$$

где β_l – относительная оценка небазисной переменной v_l .



.....
 Функция S_{ij} называется *штрафом сверху* по переменной v_j .

Аналогично, наложение на переменную ограничения для левой задачи линейного программирования (3.26) должно привести к уменьшению этой переменной на величину дробной части d_j . Для выполнения равенства $v_j = \lfloor D_j \rfloor$ необходимо уменьшить переменную v_j за счет увеличения p небазисной переменной, имеющей максимальное по модулю и отрицательное значение коэффициента $\alpha_{jp} < 0$.

Минимальное значение, на которое увеличится целевая функция вследствие возрастания небазисной переменной v_p , составит

$$S_{dj} = \min \left\{ d_j \frac{\beta_p}{\alpha_{jp}} \right\}, \alpha_{jp} < 0, p \in \bar{V}_n, \quad (3.31)$$

где β_p – относительная оценка небазисной переменной v_p .



.....
 Функция S_{dj} называется *штрафом снизу* по переменной v_j .

Ветвление осуществляется по базисной переменной v_j , которой соответствует максимальный суммарный штраф. Это делается для того, чтобы уменьшить число возвратов по ветвям дерева, если процесс ветвления остановится до

получения целочисленного решения задачи. На практике процесс ветвления обычно заканчивается, как только будет найдено целочисленное допустимое решение, значение целевой функции которого мало отличается от значения целевой функции исходной непрерывной задачи.

Если ветвление останавливается по причинам невозможности решения левой или правой задач линейного программирования либо получения значения целевой функции большего, чем значение, отвечающее лучшему из числа найденных ранее целочисленных решений, то для продолжения процесса решения необходимы возврат по дереву ветвления и выбор новой порождающей задачи. Для этого может быть использован метод наилучшей проекции, предсказывающий, какая задача приведет к наилучшему целочисленному решению. Это предсказание основано на оценке ухудшения значения целевой функции, которое произойдет при выполнении условия целочисленности по всем переменным, входящим в базис порождающей задачи. Рассмотрим эту оценку подробнее.



На каждой итерации имеется ряд порождающих задач по всем нецелочисленным переменным. Оценку относительного увеличения целевой функции на единицу уменьшения суммы дробных частей нецелочисленных переменных можно выполнить с помощью выражения

$$R_0 = (-F_{\min} + F_0) / D,$$

где F_{\min} – значение целевой функции, соответствующее решению исходной задачи в непрерывном пространстве; D – сумма невязок условий целочисленности, равная

$$D = \sum_j \min \{ d_j, 1 - d_j \},$$

где d_j – дробные части нецелочисленных переменных, а F_0 – минимальное значение целевой функции из числа найденных ранее целочисленных решений. Тогда для каждой порождающей задачи, которой соответствуют значение целевой функции F_k и сумма дробных частей нецелочисленных переменных D_k , можно найти величину $F_0 - R_0 T_k$. Для ветвления выбирается задача, которой со-

ответствует максимальное значение $F_k - R_0 T_k$, т. е. минимальное ухудшение целевой функции при дискретизации переменных.

.....

В ряде других алгоритмов порядок ветвления определяет пользователь, исходя из физических соображений и важности переменных.



Пример

Приведём пошаговое описание алгоритма метода ветвей и границ.

Шаг 1. Решается исходная задача в непрерывном пространстве; определяется рекордное значение целевой функции.

Шаг 2. Вычисляются штрафы по всем нецелочисленным базисным переменным, выбирается переменная, по которой будет осуществляться ветвление по максимальной величине суммарного штрафа $S_{uj} + S_{dj}$.

Шаг 3. Формируются левая и правая задачи ветвления, и они решаются в непрерывном пространстве. Если решение левой и правой задач невозможно, то осуществляется переход к шагу 5.

Шаг 4. Отбрасывается задача, имеющая худшее значение функции качества F , и оставшаяся задача включается в главный список. Далее – переход к шагу 6.

Шаг 5. Если значение целевой функции задачи главного списка недопустимо большое, то по методу наилучшей проекции определяется новая переменная ветвления и выполняется переход к шагу 3.

Шаг 6. Проверяется целочисленность переменных. Если не все требуемые переменные целочисленны, то осуществляется переход к шагу 2.

Шаг 7. Оформляется целочисленное решение.

.....

В большинстве случаев для сокращения затрат машинного времени в программах предусмотрен выход из задачи, если все требуемые переменные целочисленные, а полученное решение ненамного хуже оптимального, полученного при решении исходной задачи в непрерывном пространстве, так как поиск точного оптимального решения может потребовать очень больших затрат машинного времени.



Выводы

Как видно из приведенного материала, решение целочисленной задачи линейного программирования методом ветвей и границ связано с большими вычислительными сложностями. Скорость решения задачи может быть значительно увеличена за счет разумного формулирования после предварительного анализа, какие переменные должны быть обязательно целочисленны, а какие – не обязательно, наложения более жестких и двусторонних ограничений на переменные, чтобы уменьшить размеры множества, подлежащего исследованию.

Из других методов решения дискретных задач линейного программирования наибольшее распространение получили методы отсечений, сущность которых заключается в многоэтапном решении непрерывной задачи линейного программирования с добавлением при переходе от задачи к задаче новых линейных ограничений, отсекающих от симплекса решенной непрерывной задачи область с оптимальным решением с дробными значениями переменных, но не затрагивающих ни одного из допустимых целочисленных решений.



Контрольные вопросы по главе 3

1. Какие конструкторские задачи относятся к задачам структурного синтеза в проектировании ЭС?
2. Перечислите критерии оценки качества решения задачи компоновки.
3. Перечислите критерии оценки качества решения задачи размещения.
4. Перечислите критерии оценки качества решения задачи трассировки.
5. В каких постановках решается задача компоновки?
6. Для каких типов монтажных соединений применяется алгоритм Прима?
7. Для каких типов монтажных соединений применяется алгоритм Хана-на?
8. Приведите стандартную форму записи математической модели линейного программирования.
9. Приведите классификацию типовых проектных процедур.
10. В чём состоит одновариантный анализ?
11. В чём состоит многовариантный анализ?

12. В чём состоит параметрический анализ?
13. В чём состоит структурный анализ?
14. Объясните суть итерационного проектирования.
15. Назовите методы решения целочисленной задачи линейного программирования.

4 Типовые задачи конструкторского проектирования электронных средств и алгоритмы их решения

Стандартом ГОСТ 26632-85 определено значение термина «радиоэлектронное средство» как изделия и его составных частей, в основу функционирования которых положены принципы радиотехники и электроники.

Согласно данному стандарту разработана классификация РЭС по функциональным и конструктивным признакам.

Уровни разукрупнения РЭС по функциональной сложности подразделяют на системы, комплексы, устройства и узлы:

- *радиоэлектронная система* – РЭС, представляющее собой совокупность функционально взаимодействующих автономных радиоэлектронных комплексов и устройств, образующих единое целое, обладающее свойством перестроения структуры в целях рационального выбора и использования входящих средств при решении технических задач;
- *радиоэлектронный комплекс* – РЭС, представляющее собой совокупность функционально связанных радиоэлектронных устройств, обладающее свойством перестроения структуры в целях сохранения работоспособности и предназначенное для решения технических задач;
- *радиоэлектронное устройство* – РЭС, представляющее собой функционально законченную сборочную единицу, которая реализует функции передачи, приема, преобразования информации или техническую задачу на их основе и выполнена на несущей конструкции;
- *радиоэлектронный узел* – РЭС, представляющее собой функционально законченную сборочную единицу, которая реализует функцию преобразования сигнала и не имеет самостоятельного эксплуатационного значения [4, 5].

Уровни разукрупнения. РЭС по конструктивной сложности подразделяют на шкафы, каркасы и ячейки (рис. 4.1):

- *третий уровень* – радиоэлектронный шкаф (стойка, тумба, пульт), предназначенный для размещения радиоэлектронных систем и комплексов. Функционально законченный радиоэлектронный шкаф, выполненный на основе базовой несущей конструкции третьего уровня

(БНК-3) и обладающий свойствами конструктивной взаимозаменяемости, называется *радиоэлектронным модулем третьего уровня* (РЭМ-3);

- *второй уровень* – радиоэлектронный каркас (рама, корзина), предназначенный для размещения радиоэлектронных устройств. Функционально законченный радиоэлектронный блок, выполненный на основе БНК-2, обладает свойствами конструктивной взаимозаменяемости и его называют радиоэлектронным модулем второго уровня (РЭМ-2);
- *первый уровень* – радиоэлектронная ячейка, предназначенная для размещения радиоэлектронных узлов. Функционально законченную радиоэлектронную ячейку (кассету), выполненную на основе БНК-1 и обладающую свойствами конструктивной взаимозаменяемости, называют радиоэлектронным модулем первого уровня (РЭМ-1);
- *нулевой уровень* – функционально и конструктивно законченное РЭС (ИМС, микросборка, микросхема), размерно-координируемое с БНК-1 и обладающее свойствами конструктивной взаимозаменяемости, называется радиоэлектронным модулем нулевого уровня (РЭМ-0) [4, 5].

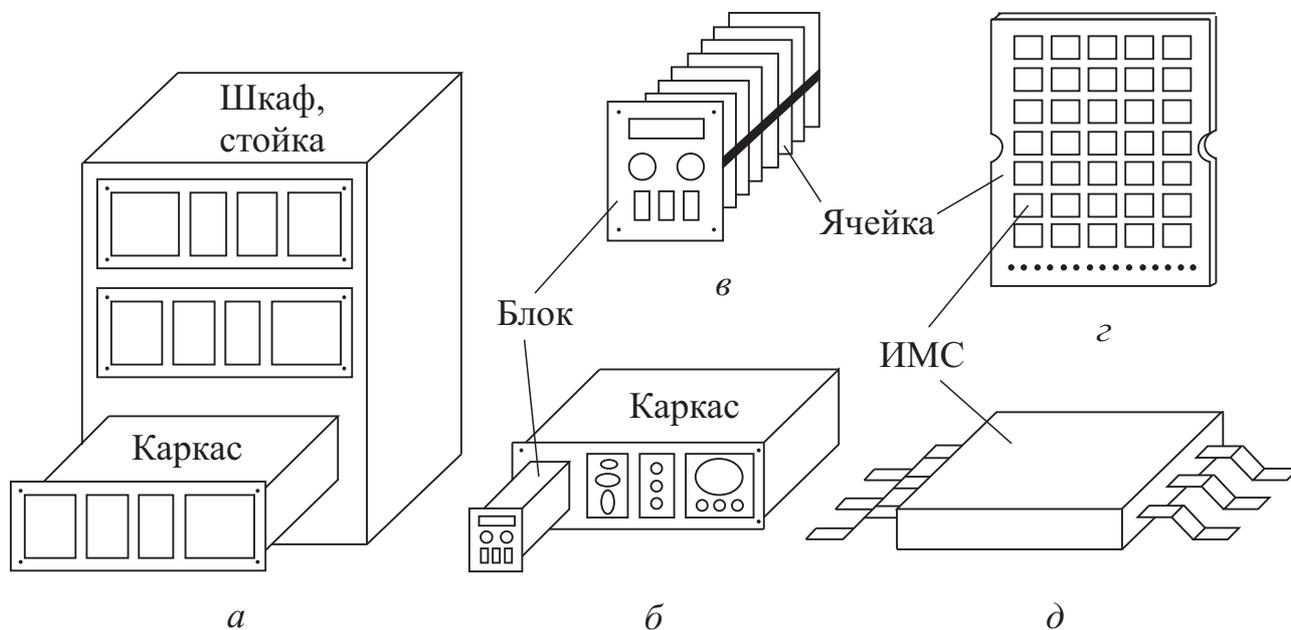


Рис. 4.1 – Уровни разукрупнения РЭС по конструктивной сложности:
a – радиоэлектронный шкаф; *б* – радиоэлектронный каркас;
в – радиоэлектронный блок; *г* – радиоэлектронная ячейка;
д – микросхема

Общие характеристики РЭС и этапы развития их конструкций и технологий

Современный научно-технический прогресс тесно связан с достижениями радиоэлектроники, внедрением РЭС во все отрасли науки и промышленности, с их постоянным усложнением.

Одновременно растут и требования к качеству, надежности, серийно-пригодности и конкурентоспособности РЭС:

- *качество* – совокупность параметров и характеристик изделия, относящихся к его способности удовлетворять установленные или предполагаемые потребности;
- *надежность* – совокупность параметров и характеристик изделия, относящихся к его способности противостоять воздействию неблагоприятных факторов условий эксплуатации;
- *серийнопригодность* – совокупность параметров изделия, характеризующих их приспособленность к серийному или массовому выпуску в условиях современного производства;
- *конкурентоспособность* – способность изделия обеспечить прибыль в условиях конкурентной борьбы за рынки сбыта.

4.1 Алгоритмы и модели компоновки



В настоящее время при разработке современной электронно-вычислительной аппаратуры основными элементами являются интегральные микросхемы (ИМС). К ИМС относятся микроэлектронные изделия, выполняющие функции преобразования, обработки сигнала или накопления информации и имеющие высокую плотность упаковки электрически соединенных элементов и кристаллов, которые рассматриваются как единое целое. Под элементом ИМС понимается его часть, реализующая функцию произвольного электрорадиоэлемента, не выделяемая как самостоятельное изделие. Компонент ИМС также реализует функцию электронной аппаратуры, но рассматривается как самостоятельное изделие. Сложность ИМС характеризуют числом содержащихся в них элементов и компонентов и называют степенью интеграции. Величину степени инте-

грации (СИ) определяют по формуле $СИ = l_g N$, где $l_g N$ – число элементов и компонентов в ИМС.

.....

При решении задач конструкторского проектирования радиоэлектронной и электронно-вычислительной аппаратуры с помощью САПР должно быть предусмотрено использование унифицированной конструкторско-технологической базы современной микроэлектроники. Поэтому они строятся по блочному принципу.

Основными элементами электронной вычислительной аппаратуры являются монолитные интегральные БИС, СБИС, для которых значения СИ соответственно равны 1 000 (и более) и 10^6 элементов и компонентов.

Конструкция ЭВА всегда содержит иерархию типовых элементов конструкций (ТЭК). Под ТЭК можно понимать конструктивно законченное изделие любой сложности.

Съемные ТЭК называют типовыми элементами замены (ТЭЗ).

Обычно говорят о конструировании на определенном ранге (уровне).

Например, ТЭК первого (низшего) ранга – это транзистор, резистор, интегральная микросхема (ИМС), модуль, микромодуль и т. д., т. е. конструктивно неделимый базовый элемент, на котором ведется конструирование.

На втором уровне элементы первого ранга, например ИМС, объединяются в схемные сочетания, образуя ТЭЗы (типовой элемент замены или ячейки и т. д.).

Так, ТЭК i -го ранга объединяются в ТЭК $(i + 1)$ -го ранга.

Как правило, конструкторско-техническая база ЭВА состоит из пяти уровней (рангов) (см. схему на рис. 4.2).

В общем случае при переходе от ранга i к рангу $(i + 1)$ приходится решать *задачу компоновки*, т. е. разбиение ТЭК уровня i на ТЭК уровня $(i + 1)$.

Исключение составляет компоновка элементов самого низкого ранга, которые составляются не из конструктивных элементов, а из *функциональных базовых элементов*.

Задачи компоновки на всех уровнях иерархии конструктивного оформления ЭВА тесно связаны между собой. Компоновка рассматривается как процесс принятия решения. В результате решения задачи компоновки типовые элементы конструкции (ТЭК) i -го уровня размещаются в ТЭК $(i + 1)$ -го уровня. При этом учитывается выбранный критерий качества.

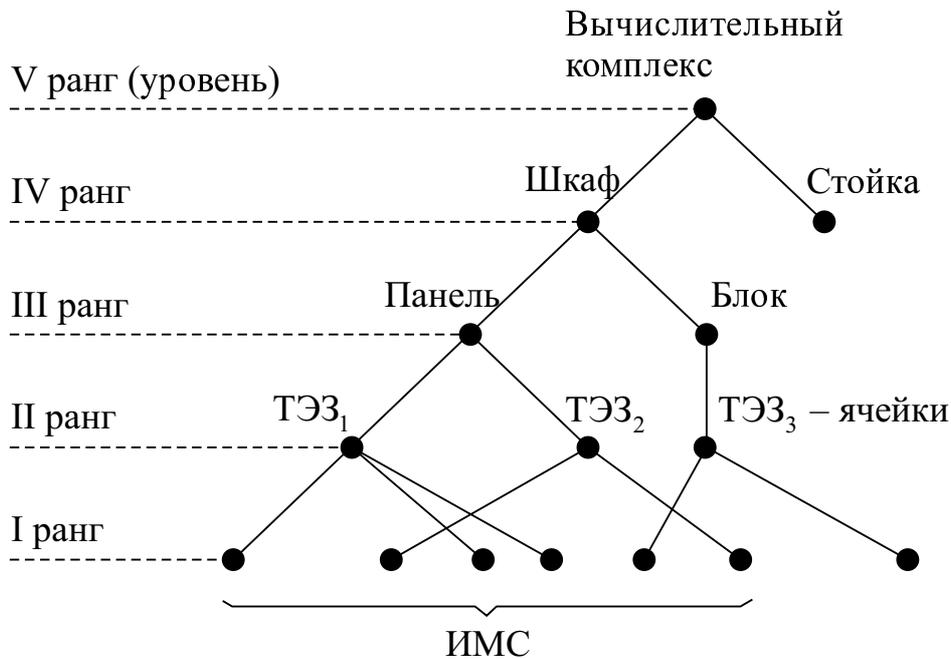


Рис. 4.2 – Ранги конструкторско-технической базы ЭВА

Основным для компоновки является критерий электромагнитной и тепловой совместимости элементов низшего уровня. Данный критерий определяет область допустимых разбиений схемы, на которой формулируются другие критерии. Такими критериями могут быть: *минимум типов конструктивно законченных частей, плотность компоновки, минимум соединений между устройствами, простота диагностирования схем* и др. Очевидно, что внешние соединения между частями схем являются одним из важнейших факторов, определяющих надежность электронной аппаратуры. Поэтому наиболее распространенным критерием является *критерий минимума числа внешних связей*. Выполнение этого критерия обеспечивает минимизацию взаимных наводок, упрощение конструкции, повышение надежности и т. д. В связи с этим рассмотрение методов компоновки электрических схем будет проводиться в основном на примере критерия минимума числа внешних связей.

Задача компоновки обычно решается «снизу вверх». Компоновка «снизу вверх» связана с распределением известных схем $(i - 1)$ -го уровня по конструкциям i -го уровня. Так, например, на самом низшем уровне элементами могут быть корпуса микросхем, а конструкциями (блоками) – типовые элементы замены, связанные друг с другом путём разъёмных соединений.

Среди методов компоновки коммутационных схем выделяют два характерных класса. К первому классу относятся такие, в которых осуществляется разбиение коммутационной схемы на части (блоки) с учетом ограничений на число элементов в блоках, либо на число внешних выводов на блоках, либо

суммарную площадь, занимаемую элементами и соединениями, и т. д. Основными критериями для такого разбиения являются:

- число образующихся блоков;
- число межузловых соединений;
- электромагнитотепловая совместимость и т. д.

Задачи такого вида возникают при разбиении коммутационных схем на блоки большой степени сложности, к которым не предъявлены требования в отношении схемной унификации.

Второй класс образуют методы, в которых кроме конструктивных характеристик образующихся частей существенны и их функциональные характеристики. Они возникают на этапе перехода от функциональных (логических) схем электронно-вычислительной аппаратуры к коммутационным схемам, ориентированным на заданную систему элементов, состоят в назначении элементов логической схемы в типовые модули из заданного набора. Указанный класс называют методом покрытия или задачами компоновки типовых блоков. Основными критериями при покрытии схем являются:

- число модулей, необходимых для покрытия исходной схемы;
- число типов используемых модулей;
- число неиспользованных элементов во всех модулях схемы и другие.

Существуют три варианта постановки задачи компоновки, в зависимости от принятых критериев:

- типизация – разбиение функциональной схемы на части различных типов по критерию минимума числа разнотипных узлов (минимум номенклатуры узлов);
- покрытие – преобразование функциональной схемы в принципиальную электрическую схему, т. е. в схему соединений ТЭЖ, номенклатура которых заранее известна;
- разрезание – разбиение функциональной схемы (или принципиальной электрической) на части с минимизацией числа связи между этими частями и ограничением числа связей каждой части [5].

Исходными данными для решения задачи компоновки является коммутационная схема (КС). Для алгоритмизации и формального решения задачи производится переход от КС к графу одним из методов, описанных выше.

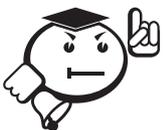
Классификация алгоритмов компоновки приведена на рисунке 4.3.



Рис. 4.3 – Классификация алгоритмов компоновки



Алгоритмы компоновки конструктивных блоков классифицируются по критериям, ограничениям на формирование частей или по структуре вычислительной процедуры. С этой точки зрения их делят на **последовательные, параллельно-последовательные и итерационные**. В алгоритмах первого типа вводится последовательный процесс компоновки частей, на каждом шаге которого в очередную часть добавляется один из элементов математической модели, выбираемый по определенному приоритету. В параллельно-последовательных алгоритмах сначала выделяется некоторое исходное множество групп элементов, которое затем распределяется по частям с учетом критериев и ограничений на компоновку. Обычно эти алгоритмы используются при решении задач компоновки со специальными требованиями, например минимизация числа разнотипных блоков. Итерационные алгоритмы служат для улучшения некоторого начального варианта компоновки в соответствии с принятым критерием [1, 5].



Решение типовых задач конструкторского проектирования ЭВС начинается с разработки графовой модели, интерпретирующей конструктивные элементы электронного устройства и связи между ними.

При использовании итерационных алгоритмов сначала граф разбивается на определенное число частей произвольным образом, например с помощью последовательных алгоритмов. Затем по некоторым правилам производится перестановка вершин из одной части в другую с целью минимизации числа внешних ребер.

В алгоритмах разбиения, основанных на математических методах, в основном используются методы ветвей и границ, а также решение задачи о назначениях. Алгоритмы разбиения с применением методов ветвей и границ состоят из следующих этапов. Сначала определяется нижняя оценка разбиения графа на заданное число частей. Затем производится построение дерева решений и осуществляется поиск оптимального результата.

Задачу разбиения графа схемы на части можно свести к задаче о назначении. При этом ищется назначение кандидатов (вершин графа) на все части, дающие минимальные суммарные затраты. При этом каждая вершина может быть назначена только на одну часть и в каждой части должны содержаться различные вершины графа.

Исходными данными для решения задачи компоновки электронной аппаратуры является коммутационная схема. Для алгоритмизации и формального решения задачи производится переход от коммутационной схемы к графу, мультиграфу, гиперграфу.



Сформулируем задачу компоновки коммутационной схемы как задачу разбиения графа $G = (X, U)$ на части $G_i = (X_i, U_i)$, $X_i \subseteq X$; $U_i \subseteq U$, $I \in I = \{1, 2, \dots, l\}$, где l – число частей, на которое разбивается граф.



Совокупность частей $B(G)$ называется **разбиением** (разрезанием) графа $G = (X, U)$, если $\forall G_i \in B(G)$ ($G_i \neq \emptyset$), $i \in I$; $\forall G_i, \forall G_j \in B(G)$, выполняются условия:

$$G_i \neq G_j \text{ и } X_i \cap X_j = \emptyset \wedge (U_i \cap U_j = U_{i,j} \vee U_i \cap U_j = \emptyset),$$

$$\bigcup_{i \in I} G_i = G. \quad (4.1)$$

Другими словами, совокупность $B(G) = \{G_1, G_2, \dots, G_i, \dots, G_l\}$ является разбиением графа G , если любая часть из этой совокупности не пустая, если для любых двух частей из $B(G)$ пересечение множества вершин пусто, а пересечение множества ребер может быть не пустым, а также если объединение всех частей в точности равно графу G .

В выражении (4.1) $U_{i,j}$ определяет подмножество рёбер, попадающих в разрез (сечение) между частями графа $G - G_i$ и G_j . Например, пусть задан граф G , содержащий 8 вершин и 17 рёбер. Одно из возможных разбиений графа G на три части показано на рисунке 4.4, где числа на ребрах равны числу кратных рёбер, связывающих соответствующие вершины.

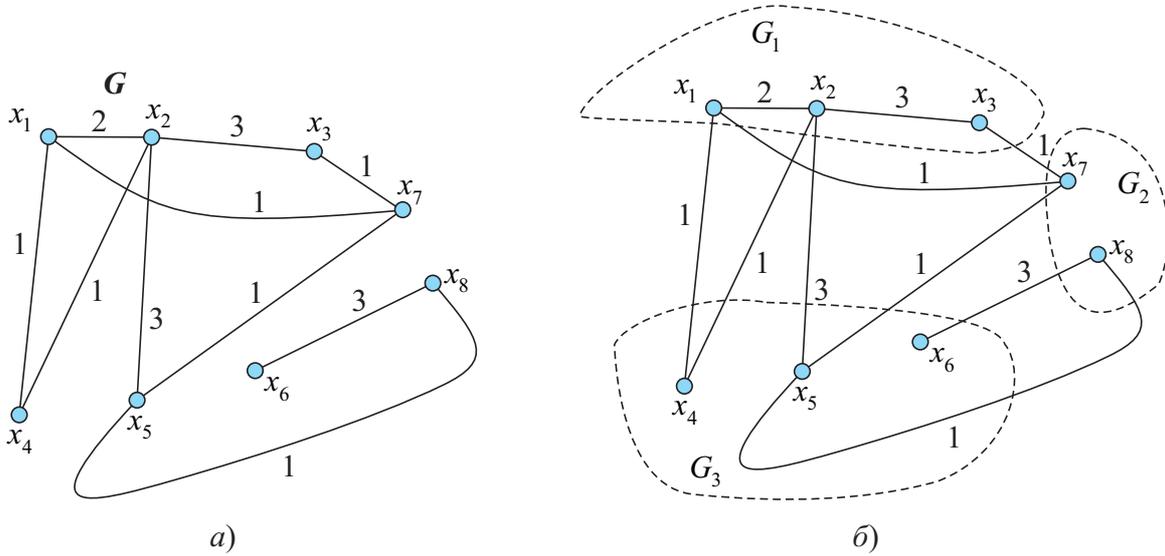


Рис. 4.4 – Пример разбиения графа $G(a)$ на три части: G_1, G_2, G_3 (б)

Обозначим $|U_{i,j}| = k_{i,j}$ и назовем его *числом реберного соединения частей* G_i и G_j графа G . Число реберного соединения K всех частей графа вычисляется по формуле:

$$K = 1/2 \sum_{i=1}^l \sum_{j=1}^l k_{i,j}, \quad i \neq j. \quad (4.2)$$

В рассмотренном примере число реберного соединения всех трех частей графа $K = 12$.

Задачей разбиения графа $G = (X, U)$ является нахождение такой совокупности частей, при которой число реберного соединения графа G удовлетворяет заданному критерию оптимальности.

Пусть граф G разбит на части G_1, G_2, \dots, G_l . В соответствии с этим разбиением множество рёбер U графа G можно представить в виде

$$U = U_1 \cup U_2 \cup \dots \cup U_l = \bigcup_{i=1}^l U_i. \quad (4.3)$$

$G_l = (X_l, U_l)$ с числом вершин в каждой части соответственно $|X_1| = n_1$, $|X_2| = n_2, \dots, |X_l| = n_l$ и $n_1 + n_2 + \dots + n_l = \sum_{i=1}^l n_i = n$ определится по формуле

$$B_y = n! / \prod_{i=1}^l (l_i!). \quad (4.6)$$

Например, число *упорядоченных* разбиений графа на рисунке 4.4 на три части по 3, 3 и 2 вершины в каждой равно

$$B_y = 8! / 3!3!2! = 560.$$

Число *неупорядоченных* разбиений графа $G = (X, U)$, $|X| = n$ на l частей, G_1, G_2, \dots, G_l ($n_1 + n_2 + \dots + n_l = n$):

$$B_n = B_y / \prod_{j=1}^n P_j!, \quad (4.7)$$

где P_j определяет число частей, имеющих по j элементов. Величина $1 / \prod_{j=1}^n P_j!$

введена в выражение (4.7) для учёта эквивалентных разбиений, связанных с упорядочиванием частей графа. Например, комбинации подмножеств вершин X_1, X_2 и X_2, X_1 при разбиении графа $G = (X, U)$ на $G_1 = (X_1, U_1)$ и $G_2 = (X_2, U_2)$ являются эквивалентными неупорядоченными разбиениями, и считать их различными при разбиении моделей схем нецелесообразно. Например, для графа на рисунке 4.4 число *неупорядоченных* разбиений составит

$$B_n = B_y / \prod_{j=1}^n P_j! = 560 : 2!1! = 280.$$

При разбиении графа схемы на конструктивно законченные части необходимость учёта эквивалентных разбиений очевидна, так как упорядочение в разбиении не приводит к изменению числа внешних рёбер. Поэтому в дальнейшем будут рассматриваться только *неупорядоченные* разбиения.

Поскольку число возможных разбиений, как видно из (4.6), (4.7), достаточно велико, возникает задача отыскания такого разбиения, при котором либо $\Delta(G)$ принимает наибольшее значение, либо K – наименьшее.

Для получения оптимального результата при разбиении графа G на части можно предложить тривиальный метод, заключающийся в получении всех воз-

можных разбиений и выборе из него наилучшего. Такой путь для конструктора неприемлем, так как он чрезвычайно затруднителен даже для современных быстродействующих ЭВМ. Наибольшее применение нашли эвристические алгоритмы разбиения. Они позволяют за конечное число шагов получать локальный минимум, который в общем случае достаточен для практических целей.

Введем некоторые классы разбиений. *Поэлементным* назовем такое разбиение графа $G = (X, U)$, при котором в каждую часть $G_i = (X_i, U_i)$ попадает по одной вершине. В этом случае число рёберного соединения всех частей равно общему числу рёбер графа, т. е. $K = m$.

Разбиение назовем *целым*, если $B(G_i) = \{G\}$. Число рёберного соединения всех частей при целом разбиении равно нулю. Для графа G , изображенного на рисунке 4.4, а, поэлементное и целое разбиения показаны на рисунке 4.5. Числа, стоящие на рёбрах, показывают количество кратных рёбер, связывающих вершины x_i, x_j . Очевидно, поэлементное и целое разбиения тривиальны, но играют важную роль при разработке эвристических алгоритмов. В частности, на практике, если это возможно, нужно стремиться к целому разбиению схемы. Другими словами, размещать всю схему в одном корпусе интегральной схемы.

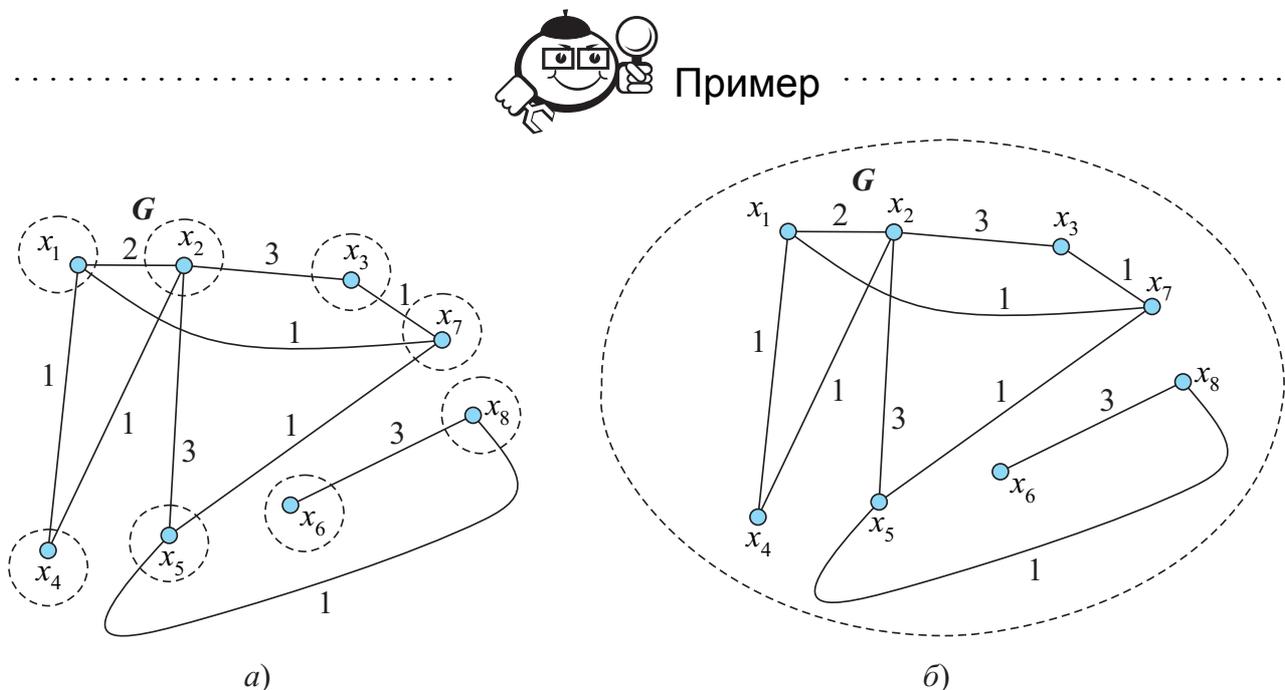


Рис. 4.5 – Разбиения графа G : а – поэлементное; б – целое

Существует большое число алгоритмов разбиения графа, которые можно условно подразделить следующим образом: *последовательные; итерационные,*

основанные на идеях математического программирования; *смешанные* (параллельно-последовательные).

.....

Выше был сформулирован критерий разбиения графа на части с минимизацией числа соединяющих ребер. Если так формировать части $G_i = (X, U)$ графа G , чтобы каждая содержала возможно большее число ребер в множестве $U_{i,i}$, то нетрудно видеть, что при этом будем получать локальный минимум K . Формирование частей графа схемы в соответствии с записанным положением составляет суть последовательных алгоритмов разбиения. Опишем один из алгоритмов последовательного типа.

Пусть задан граф схемы $G = (X, U)$, который необходимо разбить на l частей G_1, G_2, \dots, G_l с числом вершин в каждой соответственно:

$$n_1, n_2, \dots, n_l \quad (n_1 + n_2 + \dots + n_l = n).$$

Работа начинается с формирования первой части G_1 . В графе G определяется вершина $x_i \in X$ с наименьшей локальной степенью $p(x_i) = \min$. Если таких вершин несколько, то предпочтение отдается той вершине, которая имеет большее число кратных ребер. С этой вершины начинается построение. С этой целью в G_1 первоначально включаются x_i и все вершины, смежные ей. Обозначим это множество Γx_i . Если полученное число вершин равно n_1 , то G_1 образована. Если это число больше n_1 , то удаляем «лишние» вершины, связанные с остающимися вершинами G меньшим числом ребер. В случае, когда мощность множества Γx_i меньше n_1 , из Γx_i выбирается вершина, удовлетворяющая условию:

$$\sigma(x_j) = p(x_j) - a_j = \max \sigma(x_j) \quad \text{из } x_j \in \Gamma x_i,$$

где U_j – число ребер, соединяющих вершину x_j со всеми невыбранными вершинами. Строим множество вершин Γx_j , смежных x_j , и процесс выборки вершин G_1 повторяется. Образованный подграф G_1 исключаем из исходного графа. Получаем граф $G^* = (X^*, U^*)$, где $X^* = X \setminus X_1$, $U^* = U \setminus U_1$. Далее в графе G^* выбирается вершина с наименьшей локальной степенью. Производится ее помещение в G_2 , и процесс повторяется до тех пор, пока граф G не будет разре-

зан на l частей. Описанный алгоритм прост, позволяет быстро получать результаты разбиения, однако в общем случае может привести к неоптимальным результатам. Наибольшая эффективность данного метода последовательного разбиения графа имеет место, когда число вершин графа G значительно больше числа вершин в любой части разбиения $n \gg n_1, n_2, \dots, n_l$.

Рассмотрим пример работы алгоритма. Пусть граф G задан матрицей смежности:

$$R = \begin{array}{c|ccccccc} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ \hline x_1 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\ x_2 & 2 & 0 & 1 & 1 & 0 & 0 & 0 \\ x_3 & 3 & 1 & 0 & 0 & 0 & 1 & 0 \\ x_4 & 0 & 1 & 0 & 0 & 1 & 4 & 0 \\ x_5 & 0 & 0 & 0 & 1 & 0 & 1 & 5 \\ x_6 & 0 & 0 & 1 & 4 & 1 & 0 & 1 \\ x_7 & 0 & 0 & 0 & 0 & 5 & 1 & 0 \end{array}.$$

Вычислим значения локальной степени $p(x_i)$ для всех вершин путём суммирования элементов строк матрицы R :

$$p(x_1) = 5, p(x_2) = 4, p(x_3) = 5, p(x_4) = 6, p(x_5) = 7, p(x_6) = 7, p(x_7) = 6.$$

Требуется разрезать граф G на три части: G_1, G_2, G_3 , содержащие соответственно 3, 3 и 2 вершины.

Выбираем вершину x_2 , обладающую наименьшей локальной степенью, и строим множество $\Gamma x_2 = \{x_2, x_1, x_3, x_4\}$. Так как мощность множества $|\Gamma x_2| > n_1$, необходимо удалить лишнюю вершину. Для этого произведём условное удаление каждой вершины из Γx_2 и подсчитаем число рёбер, связывающих эту вершину с оставшимися вершинами из Γx_2 . При удалении вершины x_1 число рёбер, соединяющих её с $\Gamma x_2 \setminus x_1$, равно $Zx_1 = 5$. Аналогично для x_3 $Zx_3 = 4$, для x_4 $Zx_4 = 1$. Следовательно, из множества Γx_2 удаляется вершина x_4 . В результате получаем $G_1 = (X_1, U_1)$, $X_1 = \{x_1, x_2, x_3\}$. После удаления из графа G подграфа G_1 получаем граф G^* , матрица смежности R^* которого примет вид:

	x_4	x_5	x_6	x_7
x_4	0	1	4	0
$R^* = x_5$	1	0	1	5
x_6	4	1	0	1
x_7	0	5	1	0

Вычислим локальные степени вершин $p(x_4) = 5$, $p(x_5) = 7$, $p(x_6) = 6$, $p(x_7) = 6$.

Выбираем вершину x_4 , обладающую наибольшим числом кратных рёбер локальной степени, и строим множество $\Gamma x_4 = \{x_4, x_5, x_6\}$. Так как мощность множества $|\Gamma x_4| > n_2$, то необходимо удалить лишнюю вершину. Произведём условное удаление каждой вершины из Γx_4 и подсчитаем число рёбер, связывающих эту вершину с оставшимися вершинами из Γx_4 . При удалении вершины x_4 число рёбер, соединяющих её с $\Gamma x_4 \setminus x_4$, равно $Zx_4 = 5$. Аналогично для x_6 $Zx_6 = 5$, для x_5 $Zx_5 = 2$. Следовательно, из множества Γx_4 удаляется вершина x_5 . В результате получаем $G_2 = (X_2, U_2)$, $X_2 = \{x_4, x_6\}$. Оставшийся подграф G_3 соответственно включает вершины $X_3 = \{x_5, x_7\}$. В результате разбиения графа G на три части G_1 , G_2 , G_3 получили число рёберного разбиения $K = 5$.

В параллельно-последовательных алгоритмах сначала выделяется некоторое множество групп элементов, которое затем распределяется по частям с учётом критериев и ограничений на компоновку. Обычно эти алгоритмы используются при решении задач компоновки со специальными требованиями, например минимизация разнотипных блоков. Алгоритмы этих двух типов применяются для создания базового (начального) варианта компоновки. Итерационные алгоритмы служат для улучшения некоторого начального варианта компоновки в соответствии с принятым критерием путём перестановки вершин из одной части в другую, руководствуясь некоторыми правилами.

Алгоритмы компоновки типовых блоков сводятся к задачам *покрытия и типизации*. Под *покрытием* понимается представление функциональной схемы ЭВА типовыми конструктивными элементами, на которых она будет реализована, и связями между ними с выполнением заданных конструктивных ограни-

чений. При покрытии выделяют задачи с несвязными элементами и функциональными ячейками. В первом случае решают задачи определения необходимого числа ячеек для покрытия коммутационной схемы с минимальной суммарной стоимостью (математическая модель (4.8)–(4.10)). Во втором случае решают задачи покрытия коммутационной схемы заданным классом функциональных ячеек с минимизацией числа ячеек и суммарного числа межъячеечных связей (математическая модель (4.11)–(4.13)).

$$L(x) = \sum_{j \in J} c_j \cdot x_j \rightarrow \min \quad (4.8)$$

при условиях (ограничениях):

$$\sum_{j \in J} a_{i,j} \cdot x_j \geq b_i, \quad i = \overline{1, m}; \quad (4.9)$$

$$x_j \geq 0, \quad x_j - \text{целое для всех } j, \quad (4.10)$$

где $a_{i,j}$ – количество элементов i -го типа, содержащееся в ячейке j -го типа;

x_j – количество ячеек j -го типа, используемых при покрытии элементов коммутационной схемы;

c_j – стоимость ячейки j -го типа.

$$L(x) = \sum_{j \in J} \sum_{i=1}^k x_j \cdot \rho_{i,j} \rightarrow \min \quad (4.11)$$

при условиях (ограничениях):

$$\sum_{j \in J} a_{i,j} \cdot x_j \geq b_i, \quad i = \overline{1, m}; \quad (4.12)$$

$$x_j \geq 0, \quad x_j - \text{целое для всех } j, \quad (4.13)$$

где $a_{i,j}$ – количество элементов i -го типа, содержащееся в ячейке j -го типа;

x_j – количество ячеек j -го типа, используемых при покрытии коммутационной схемы;

$\rho_{i,j}$ – количество внешних связей ячейки типа j при реализации ею элемента схемы i -го типа;

b_i – количество элементов i -го типа, содержащееся в коммутационной схеме.

Типизация – это разбиение коммутационной схемы на части по критерию оптимальности – минимуму номенклатуры частей разбиения или по критерию

оптимальности – максимуму однотипности используемых ячеек. Сокращение номенклатуры типовых элементов конструкций позволяет уменьшить затраты на дальнейшее проектирование.

Математическая модель задачи типизации имеет вид:

$$L(x) = \sum_{j \in J} \sum_{i \in I} x_{i,j} \rightarrow \min \quad (4.14)$$

при условиях (ограничениях):

$$\sum_{j \in J} x_{i,j} = 1, \quad i = \overline{1, m}; \quad (4.15)$$

$$\sum_{i=1}^m n_i \cdot x_{i,j} = k_j, \quad j \in J; \quad (4.16)$$

$$x_{i,j} = \{0; 1\}, \quad (4.17)$$

где $x_{i,j}$ – переменная проектирования; $x_{i,j} = 1$, если ячейка типа j реализует i -й элемент схемы; $x_{i,j} = 0$, если ячейка типа j не может реализовать i -й элемент схемы;

k_j – количество элементов, содержащееся в ячейке j -го типа;

n_i – количество элементов i -го типа, содержащееся в ячейке j -го типа.

Ограничение (4.15) показывает, что каждый элемент коммутационной схемы может быть реализован только одной ячейкой соответствующего типа.

Ограничение (4.16) показывает, что число элементов схемы, реализуемых ячейкой j -го типа, не должно превышать числа элементов, содержащихся в ней.

Важной задачей в общей проблеме компоновки коммутационной схемы является покрытие. Как отмечалось выше, под покрытием схемы понимается представление функциональной схемы типовыми элементами конструкций, на которых она будет реализована, и связями между ними. Решение «задачи покрытия» на первом уровне конструкторского проектирования дает возможность представить функциональную схему ЭВА или ее частей в виде принципиальной электрической схемы соединений элементов. Элементами могут быть резисторы, конденсаторы, транзисторы, ИМС и другие, т. е. ТЭК 1-го уровня. Поэтому при решении задачи покрытия можно рассматривать вопросы выбора класса элементов и минимизации числа типов элементов. Конечной целью покрытия является выбор оптимальной элементно-технической базы объекта проектирования.

Исходной информацией для этого этапа проектирования являются схема функциональная и набор ячеек $Я = \{Я_1, Я_2, \dots, Я_n\}$, параметры которых оказывают решающее влияние на результаты покрытия схемы.

Пусть задана схема, состоящая из элементов x_1, x_2, \dots, x_n , для каждого из которых известен тип $t(x_i)$, $t \in \{1, \dots, l\}$. Состав схемы по типам элементов описывается вектором $M = \{m_1, m_2, \dots, m_l\}$, в котором m_t – число элементов типа t . Состав ячеек заданного набора $Я$ описывается матрицей $R = \|r_{v,t}\|$, $v \in \{1..k\}$, $t \in \{1..n\}$, где $r_{v,t}$ – число элементов типа t в ячейке $Я_v$. *Схема считается покрытой ячейками из набора $Я$, если каждый элемент схемы реализуется элементами, входящими в состав выбранных ячеек.* Пусть известны стоимости c_v ячеек каждого типа. Требуется отыскать покрытие с минимальной стоимостью F .

$$F = \sum_{t=1}^n c_v \cdot |R_v|.$$

Для формализации процесса покрытия схемы ячейками различные наборы делятся на классы.

1. В состав ячеек входят логические элементы одного типа, не связанные между собой; входы и выходы элементов имеют отдельные выводы на ячейке.
2. Ячейки содержат разнотипные элементы, не связанные между собой; входы и выходы элементов имеют отдельные выводы на ячейке.
3. Состав ячеек аналогичен классу 1, однако часть элементов связана между собой.
4. Состав ячеек аналогичен классу 2, однако часть элементов связана между собой.

Рассмотрим задачу, относящуюся к классу 2. Математическая модель задачи покрытия для данного класса запишется:

$$F = \sum_{t=1}^n c_v \cdot |Я_v| \Rightarrow \min \quad (4.18)$$

при условиях:

$$\sum_{v \in V} r_{v,t} \cdot |Я_v| - \sum_{a=1}^l S_{t,a} + \sum_{v=1} S_{a,t} \geq m_t; \quad (4.19)$$

$$|Y_v| \geq 0; \quad (4.20)$$

$$S_{t,a} \geq 0 - \text{целые числа}, \quad (4.21)$$

где m_t – число элементов типа t , используемых в схеме;

$S_{t,a}$ – количество элементов типа « t », которые могут быть заменены элементами типа « a »;

$S_{a,t}$ – количество элементов типа « a », которые могут быть заменены элементами типа « t »;

$|Y_v|$ – число ячеек типа Y_v , использованных при реализации схемы.

Задача (4.14)–(4.17) относится к задачам целочисленного программирования. Ее решение точными методами затруднено. Поэтому применяют приближенную эвристическую процедуру решения поставленной задачи. На практике хорошо зарекомендовал себя алгоритм Селютина. Рассмотрим алгоритм Селютина компоновки ячеек с несвязными элементами. Пример такой ячейки «K155LA3» приведен на рисунке 4.6. В состав данной ячейки входят однотипные элементы, реализующие функцию «И-НЕ» на два входа.

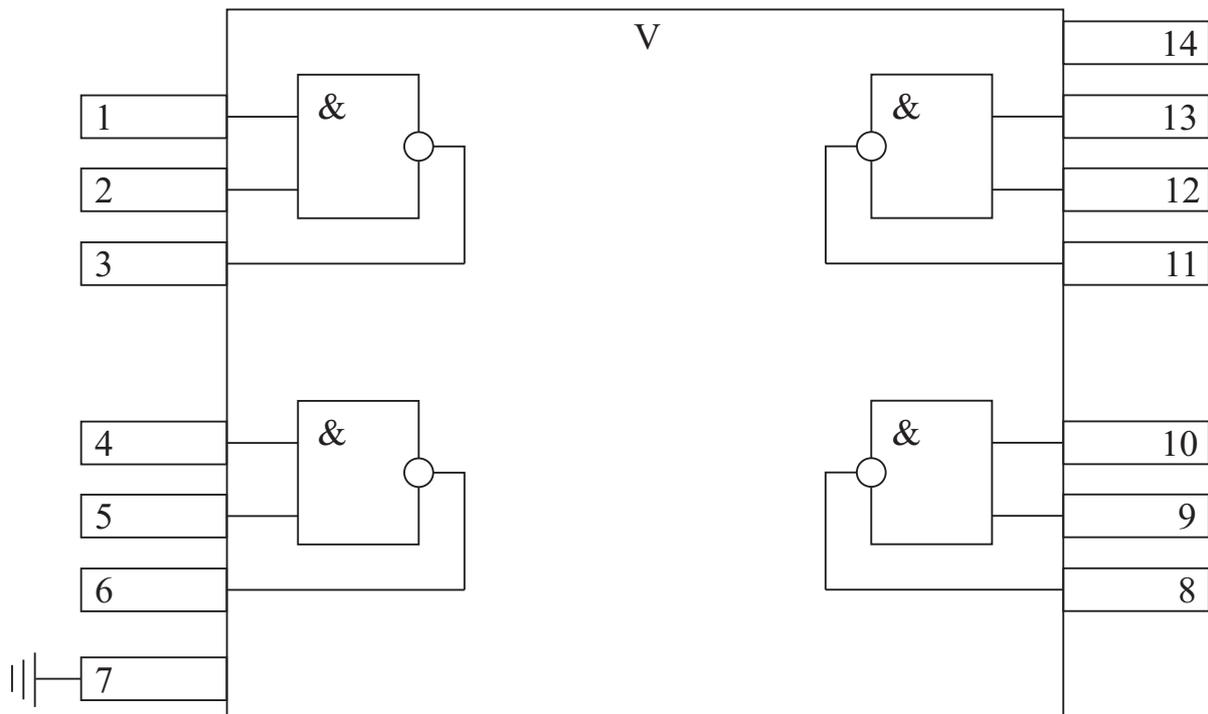


Рис. 4.6 – Типовая ячейка K155LA3

Для общности примем, что набор ячеек относится к классу 2, т. е. в него входят ячейки с разнотипными элементами.

При решении задачи будем считать, что каждый элемент схемы реализуется элементом того же типа в ячейках набора $Я$. В качестве дополнительного критерия оптимизации принимается число межъячеечных соединений. Решение задачи разбивается на два этапа: 1) определение необходимого числа ячеек с минимальной суммарной стоимостью; 2) минимизация числа связей между ячейками. Пусть дана матрица $R = \|r_{v,t}\|$, $v \in \{1..k\}$, $t \in \{1..n\}$ и вектор числа элементов в схеме по типам $M_0 = \|m_t\|$.

Алгоритм решения задачи:

1. Упорядочить ячейки набора $\{Я_v\}$ по возрастанию их стоимости: $Я_1$; $Я_2$; $Я_3$; ...; $Я_v$ ($C_1 \leq C_2 \leq \dots \leq C_k \leq \dots \leq C_v$). Выбирается ячейка $Я_k$ с наименьшим значением стоимости C_k . Задаем начальное значение номера итерации $i = 0$. По результатам структурного анализа покрываемой схемы определяем вектор M_0 , элементы которого равны числу каждого типа элементов, использованных в схеме.

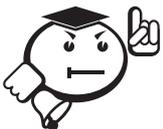
2. Определить «локально-минимальное» число ячеек $Я_k$ для итерации $i = i + 1$: $A_k^{(i)} = \min \{m_t / r_{k,t}\}$, где $r_{k,t}$ – количество элементов типа « t » в ячейке типа « k ».

3. Вычислить вектор «непокрытых элементов» M_i :

$$M_i = M_{i-1} - A_k^{(i)} R_k.$$

4. Если $M_i > 0$, перейти к п. 2, если $M_i = 0$, перейти к п. 5.

5. Подсчитать число ячеек каждого типа: $|Я_v| = \sum_{i=1}^p A_v^{(i)}$, где p – число выполненных при покрытии итераций; $v = \{1, 2, 3, \dots, k, \dots, v\}$.



.....

Исходными данными для решения задачи покрытия являются логическая схема устройства (пример на рис. 4.7), типовые логические элементы $\{t_i\}$, $i = 1, 2, \dots, n$ (пример на рис. 4.8), на которых построена данная схема, и типовые конструктивные ячейки (модули), с помощью которых должно быть реализовано устройство.

.....

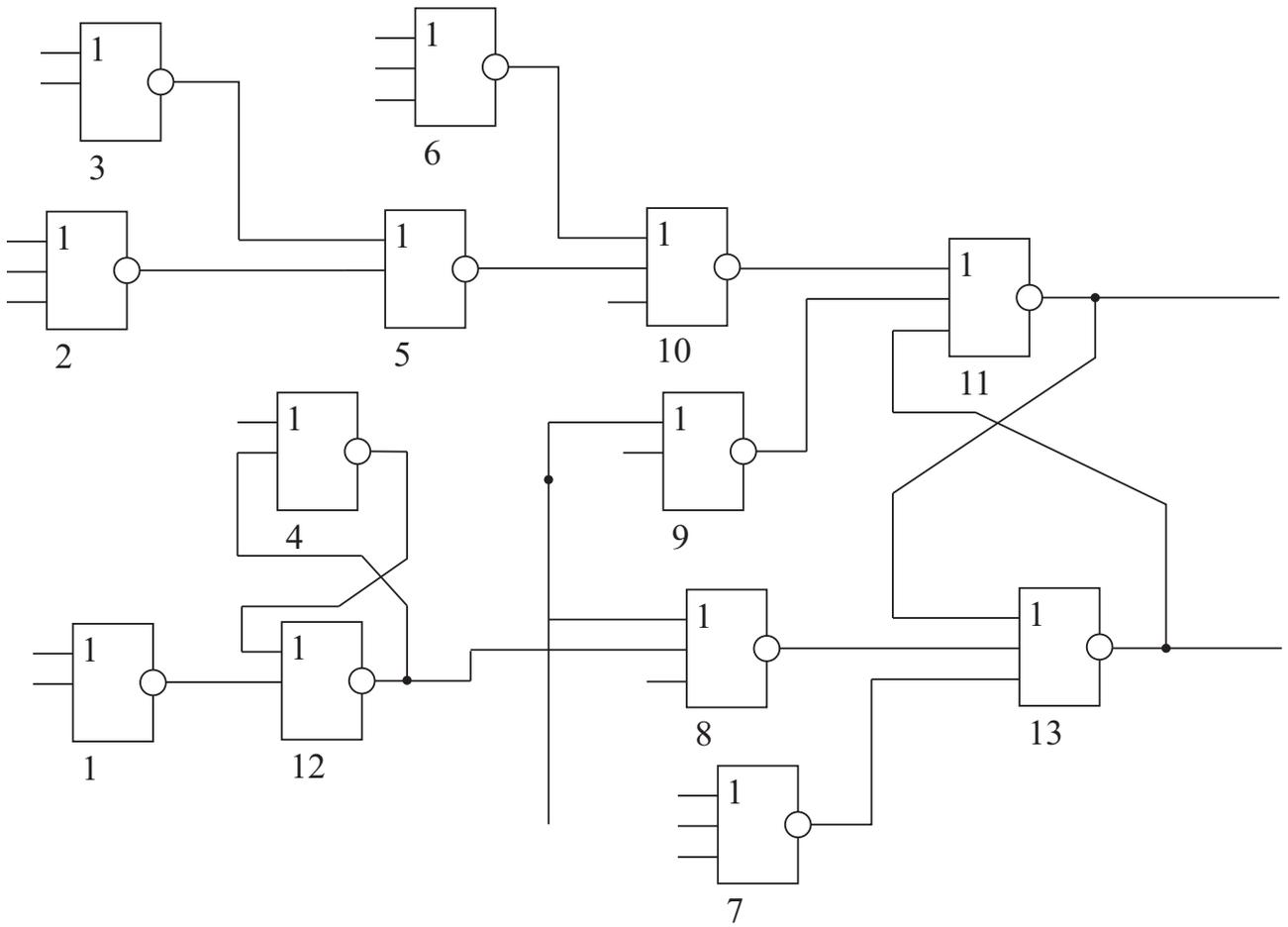


Рис. 4.7 – Логическая схема узла РЭС

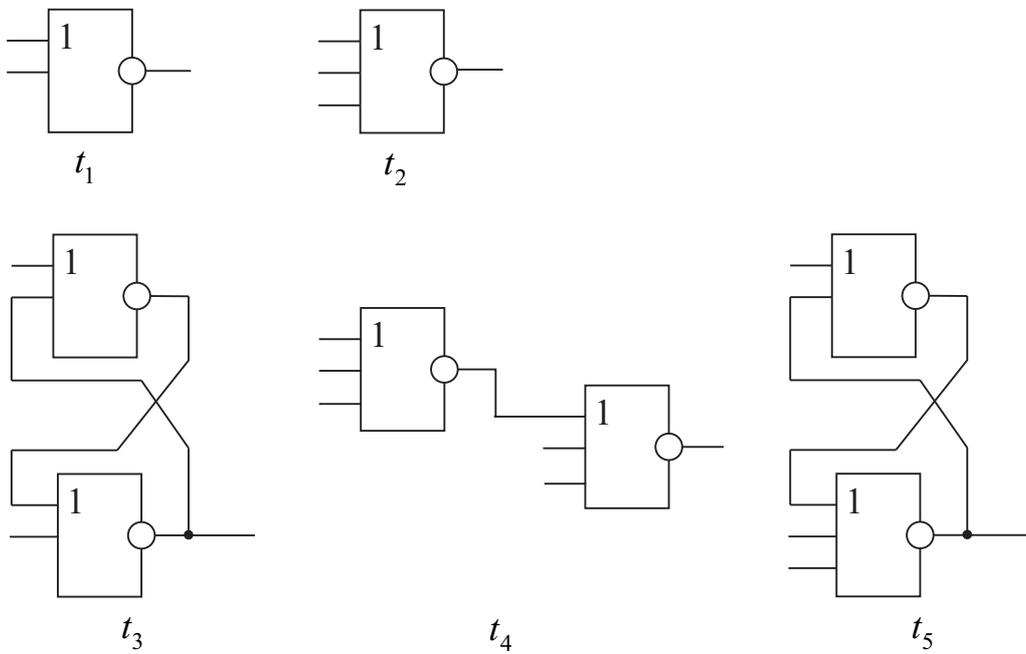


Рис. 4.8 – Типовые логические элементы

Условное изображение ячеек $Я_1, Я_2, Я_3, Я_4$ с указанием содержащихся в них типовых логических элементов t_1, t_2, t_3, t_4, t_5 показано на рисунке 4.9.

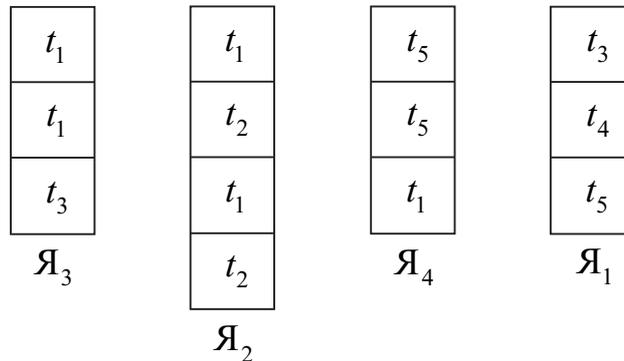


Рис. 4.9 – Ячейки $Я_1, Я_2, Я_3, Я_4$

Для решения задачи покрытия данные о наборе ячеек или конструктивных модулях, с помощью которых должно быть реализовано устройство, удобно задавать матрицей смежности $R = \|r_{ij}\|$, строки которой соответствуют типу ячейки ($Я_i$), а столбцы – типовым логическим элементам $\{t_j\}$.

Данные структурного анализа схемы приведены в таблице 4.1.

Таблица 4.1 – Данные структурного анализа схемы

Тип элемента схемы	t_1	t_2	t_3	t_4	t_5
Количество использований элемента в схеме	5	4	1	1	0

В соответствии с исходными данными, математическая модель задачи покрытия запишется в виде:

$$L = c_1x_1 + c_2x_2 + c_3x_3 \rightarrow \min \quad (4.22)$$

при условиях:

$$2x_1 + x_3 \geq 5; \quad (4.23)$$

$$x_1 + x_2 \geq 4; \quad (4.24)$$

$$x_4 \geq 1; \quad (4.25)$$

$$x_3 \geq 1; \quad (4.26)$$

$$\forall x_j \geq 0; \quad x_j - \text{целое}, \quad (4.27)$$

где x_j – количество ячеек j -го типа, выбранных для покрытия схемы.

Решение данной задачи покрытия с помощью алгоритма Селютина [5]

1. По данным таблицы 4.1 запишем вектор $M_0 = (5, 4, 1, 1)$.

2. Упорядочим ячейки заданного набора $\{Y_v\}$ по возрастанию их стоимости:

$$Y_1; Y_2; Y_3 \quad (C_1 \leq C_2 \leq C_3).$$

3. Выбираем ячейку Y_1 , так как она имеет наименьшую стоимость.

Определяем «локально-минимальное» число $A_1^{(1)}$ использования ячейки Y_1 . Ячейка Y_1 может реализовать элементы схемы типа t_1, t_2 (см. матрицу R_v).

Для реализации элементов схемы типа t_1 требуются две ячейки:

$$Y_1 \rightarrow A_{1,1} = (5 \text{ div } 2) = 2;$$

для реализации элементов схемы типа t_2 – четыре ячейки:

$$Y_1 \rightarrow A_{1,2} = (4 \text{ div } 1) = 4.$$

Для нашего примера $A_1^{(1)} = \min \{A_{1,1}, A_{1,2}\}; A_1^{(1)} = 2$. Следовательно,

$$|Y_1| = 2 \text{ шт.}$$

Переход к следующей итерации: $i = i + 1, i = 0 + 1, i = 1$.

4. Вычисляем вектор «непокрытых элементов» M_i , где i – номер выполненной итерации ($i = 1$).

$$M_1 = M_0 - A_1^1 \cdot r_{1,1}; \text{ получаем } M_1 = (1, 2, 1, 1).$$

5. Так как вектор $M_1 \neq 0$, то выполняем вторую итерацию ($i = 2$) для ячейки Y_2 . Все вычисления проводятся аналогично вычислениям на первой итерации, но с новыми значениями вектора M_i .

$$6. A_2^2 = \min \begin{cases} A_{2,2} = 2, \\ A_{2,3} = 1; \end{cases} A_2^2 = 1; |Y_2| = 1; M_2 = (1, 1, 0, 1).$$

Так как $M_2 > 0$, выполняем 3-ю итерацию.

7. Третья итерация выполняется для ячейки Y_3 .

$$A_3^3 = \min \begin{cases} A_{3,1} = 1 \\ A_{3,4} = 1 \end{cases}; \quad |\mathbf{Я}_3| = 1.$$

Вычисляем $M_3 = (0, 1, 0, 0)$.

Так как $M_3 > 0$, выполняем 4-ю итерацию.

8. Вектор M_3 содержит только один ненулевой элемент – m_2 . Следовательно, в схеме остался только один нереализованный элемент типа t_2 , который может быть реализован ячейками $\mathbf{Я}_1$ и $\mathbf{Я}_2$.

Так как $C_1 < C_2$, выбираем ячейку $\mathbf{Я}_1$.

9. Вычисляем вектор M_4 : $M_4 = (0, 0, 0, 0)$.

Так как $M_4 = 0$, решение задачи заканчивается. Остается подсчитать суммарное количество ячеек каждого типа, вычисленное на каждой итерации.

Для данного примера $|\mathbf{Я}_1| = 3$, $|\mathbf{Я}_2| = 1$, $|\mathbf{Я}_3| = 1$.

Выше говорилось о том, что ячейка – это неделимый базовый элемент конструкции, поэтому при покрытии допускается некоторая избыточность элементов ячеек ввиду их недоиспользования, например одна из трех ячеек $\mathbf{Я}_1$ имеет избыточность по элементам типа t_1 .

Функциональная схема радиоэлектронной аппаратуры может быть представлена графом $G = (X, U)$, где X – множество вершин, интерпретирующих логические элементы схемы; U – множество рёбер, интерпретирующих связи между элементами схемы.

Необходимо покрыть данную схему комплексом интегральных микросхем (БИС). Тогда каждый компонент БИС можно представить в виде подграфа G' . В результате получим некоторое множество подграфов, соответствующих ИМС заданного комплекса. Задача покрытия формулируется теперь как покрытие графа G подграфами из множества $\{G'_1, G'_2, \dots, G'_p\}$. Один из возможных путей решения этой задачи следующий. Каждой вершине графа G присвоить код $S_{i,p}$, где i – номер вершины; p – тип элемента логической схемы.

Каждому ребру графа ставится в соответствие значение функции веса $\mathcal{G}_{i,k}$.

$$g_{i,k} = \begin{cases} 1, & \text{если ребро } u_i \text{ инцидентно однотипным вершинам;} \\ 0, & \text{в противном случае.} \end{cases}$$

Внутри ИМС, как правило, объединяются однотипные компоненты. Из нескольких вариантов покрытия схемы выбирают тот, для которого

$$g = \frac{1}{2} \sum_i^n \sum_k^n g_{ik} \Rightarrow \min.$$

Рассмотрим компоновку как задачу типизации. *Типизация* – это разбиение коммутационной схемы на части с минимизацией номенклатуры частей разбиения. Однотипными называются ТЭК, имеющие одинаковый состав элементов и одинаковую коммутационную схему. В этом случае задача типизации формулируется как задача выделения в графе изоморфных подграфов [4, 5]. Задача выделения в графе изоморфных подграфов формулируется следующим образом: найти разбиение $\varphi(G) \in \Phi$ графа G на множество групп изоморфных подграфов, удовлетворяющих следующим условиям:

- 1) любые два подграфа G_i, G_k , принадлежащие одной группе разбиения, должны быть изоморфны;
- 2) множества вершин любых двух подграфов не должны пересекаться;
- 3) число вершин любого подграфа разбиения не должно превышать заданное (конструктивное ограничение, связанное с числом элементов на типовом элементе конструкции);
- 4) суммарное число внешних ребер каждого подграфа не должно превышать заданное (конструктивное ограничение, связанное с числом элементов разъема или длиной параметра корпуса ТЭК).

Критерием оптимальности при типизации является минимальное число групп изоморфных подграфов, полученных в результате разбиения.

Для решения задачи типизации путем сведения ее к задаче выделения в графе изоморфных подграфов необходимо:

- 1) описать исходную коммутационную схему с помощью графовой модели;
- 2) решить для графа задачу выделения в нем изоморфных подграфов;
- 3) от полученных результатов решения задачи на графе необходимо перейти к результатам решения задачи типизации в коммутационной схеме.

4.2 Размещение конструктивных элементов в монтажном пространстве

Размещение определяет качество трассировки, быстродействие, надежность. Из-за малого времени переключения схем ЦВМ длина соединений влияет на задержки в распространении сигнала, затрудняет синхронизацию.

Исходные данные для размещения:

- 1) прямоугольная конструкция (ячейка, кристалл, панель);
- 2) число элементов для размещения;
- 3) граф схемы соединений элементов $G = (X, U)$.

На заданную конструкцию накладывается координатная сетка (решетка) и оси координат s, t , представляя ее таким образом в виде графа G_r . Теперь задача размещения сводится к отображению заданного графа схемы $G = (X, U)$ в координатную сетку G_r таким образом, чтобы вершины множества X размещались в ее узлах и суммарная длина связей $L(G) = \frac{1}{2} \sum_{i,j} d_{i,j} r_{i,j}$ или суммарное

число пересечений $P(G) = \frac{1}{2} \sum_{u_{i,j} \in U} P(u_{i,j})$ были наименьшими для возможных способов отождествления вершин графа и узлов сетки. В приведенных формулах переменная $r_{i,j}$ равна числу кратных ребер между вершинами x_i, x_j , а $P(u_{i,j})$ – числу пересечений ребра $u_{i,j}$ со всеми остальными ребрами графа, $d_{i,j}$ – расстояние между узлами сетки, в которых размещены вершины x_i, x_j .

Расстояние $d_{i,j}$ между узлами сетки, в которых размещены вершины x_i, x_j , определяется выражением – $d_{ij} = |s_i - s_j| + |t_i - t_j|$.

Расстояние $d_{i,j}$ между узлами сетки, в которых размещены вершины x_i, x_j , определяется выражением – $d_{ij} = |s_i - s_j| + |t_i - t_j|$.

Функция расстояний для узлов сетки G_r задается матрицей расстояний D

$$D = \|d_{ij}\|; \quad d_{ij} = \begin{cases} 0, & \text{если } x_i = x_j; \\ d_{ij}, & \text{если } x_i \neq x_j. \end{cases}$$

Существует большое число алгоритмов размещения графа схемы с минимизацией суммарной длины соединений и внутрисхемных пересечений. Все алгоритмы можно разделить на две группы: непрерывно-дискретные и дискретные. К первой группе относятся алгоритмы, основанные на градиентных

методах. Ко второй группе относятся итерационные, последовательные, смешанные, а также алгоритмы, основанные на идеях метода ветвей и границ.

Последовательные алгоритмы размещения заключаются в выборе первоначально размещенного элемента или группы элементов с последующим подсоединением неразмещенных элементов. После размещения элементов они уже не перемещаются. Правила выбора и расстановки элементов зависят от конкретных методов.

Итерационные алгоритмы размещения с улучшением качества работают в итеративном режиме. Для изменения позиций размещения элементов выбираются одиночные элементы или группы элементов. Затем по заданным правилам производится перемещение элементов для уменьшения общей длины соединений, что позволяет получать более качественные результаты, чем последовательные за счет больших затрат машинного времени. К группе итерационных алгоритмов относятся *стохастические методы* размещения. Основная идея этих методов следующая. Случайным образом распределяются элементы по посадочным местам плоскости с учетом плотности распределения вероятности, которую считают равномерной. Далее определяется суммарная длина соединений и в полученном размещении результат сравнивается с предыдущим. Лучшее размещение оставляется. Процесс продолжается до тех пор, пока не окончится отведенное время или не будет просмотрено заданное число размещений.

Алгоритмы, основанные на идеях *метода ветвей и границ*, относятся к точным. При этом множество всех допустимых решений разбивается на меньшие по мощности подмножества, в которых производится поиск оптимального размещения. Метод сопровождается вычислением низших границ. Поиск оптимального решения прекращается, когда граничное значение начинает превышать значение при найденном допустимом размещении. Процесс продолжается до тех пор, пока не будет закончен поиск в каждом подмножестве разбиения или не будет найдено оптимальное размещение.

При реализации алгоритмов в общем случае могут получаться локальные минимумы целевой функции.

Рассмотрим *последовательно-итерационный алгоритм* размещения вершин графа на плоскости с минимизацией суммарной длины соединений. *Последовательная часть*, которая применяется для упорядочивания множества вершин и получения начального размещения, использует понятие коэффициента связности для каждой вершины.

Исходными данными для решения размещения являются граф $G = (X, U)$, интерпретирующий схему электрическую принципиальную, и монтажная плоскость заданных размеров, на которой размещен граф сетки G_r . В общем случае считают, что монтажная плоскость имеет прямоугольную форму. В частном случае форма монтажной плоскости может быть любой. Требуется разместить вершины $x_i \in X$ графа G в узлы сетки G_r с минимизацией суммарной длины ребер $u_j \in U$. Число узлов сетки G_r больше или равно числу вершин графа G . Предположим, что часть узлов сетки G_r уже отождествлена с некоторыми вершинами графа схемы.

Первый этап размещения

Пусть вершина $x_i \in X$ помещена в $(j + 1)$ -ю позицию, где j – число занятых позиций, тогда коэффициентом связности $\Delta(x_i)$ называется выражение: $\Delta(x_i) = a_{i,h} - a_{i,z}$, где $a_{i,h}$ – число ребер, связывающих x_i -ю вершину с ранее размещенными; $a_{i,z} = p(x_i) - a_{i,h}$, т. е. число ребер, связывающих x_i с неразмещенными вершинами, где $p(x_i)$ – число ребер, инцидентных вершине x_i (локальная степень x_i). Тогда $\Delta(x_i) = 2a_{i,h} - p(x_i)$, где $a_{i,h}$ – число ребер, соединяющих x_i с ранее размещенными вершинами. Значение $\Delta(x_i)$ определяется для всех неразмещенных вершин, и выбирается x_i с $\max \Delta(x_i)$. Вершина x_i помещается в первую свободную позицию (узел сетки G_r). Как правило, размещение вершин начинается с крайнего левого узла сетки. Процесс последовательно продолжается до тех пор, пока не будут размещены все вершины графа G .

Второй этап размещения (итерационная часть алгоритма размещения)

Итерационная часть алгоритма размещения основана на понятии центра тяжести вершины.

Для оценки степени «предпочтительности» v -й позиции для каждой вершины графа x_j вводится понятие *средней длины ребра* – L_j :

$$L_j = \sum_{i=1}^n d_{i,j} \cdot r_{i,j} / \sum_{i=1}^n r_{i,j}, \quad i \neq j,$$

где $d_{i,j}$ – расстояние между узлами сетки, в которые помещены вершины x_i и все смежные ей j -е вершины; $r_{i,j}$ – число связей вершины x_j со всеми смежными с ней j -ми вершинами графа G ; L_j – средняя длина ребер, инцидентных вершине x_j , помещенной в позицию v .

Очевидно, что $(\forall u_{i,j} \in U_j) \left[(d_{i,j} = 1) \Rightarrow L_j = 1 \right]$, где U_j – множество ребер, инцидентных вершине x_j , помещенной в позицию v . *Необходимо стремиться к такому размещению вершин графа, когда средняя длина ребер стремится к минимальному значению.*

Из множества вершин графа, расположенных в сетке, выбирается вершина x_j , имеющая максимальное значение L_j , и производится её перестановка с другой вершиной с целью минимизации суммарной длины соединений:

$$L(G) = \frac{1}{2} \sum_{i,j} d_{i,j} r_{i,j}.$$

Очевидно, что в перестановках должны участвовать только смежные вершины.

Для характеристики системы точек, связанных с вершиной x_j , вводится понятие *центра тяжести*, что позволяет рассматривать их как систему связанных материальных точек и отыскать такое положение для x_j , которое обеспечит равновесие в сетке с точки зрения минимизации $L(G)$. Вычисление координат центра тяжести (s_c и t_c) системы материальных точек производится по формулам:

$$\left. \begin{aligned} s_c &= \sum_{j=1}^n S_j \cdot r_{i,j} / \rho(x_i) \\ t_c &= \sum_{j=1}^n t_j \cdot r_{i,j} / \rho(x_i) \end{aligned} \right\}$$

где $r_{i,j}$ – количество связей между вершиной x_i и смежными с ней вершинами $\{x_j\}$; вершина x_i – это вершина с максимальным значением L_j ; $\rho(x_i)$ – степень вершины x_i .

Оценка качества решения задачи размещения в целом производится по формуле:

$$L_{\text{общ}} = \frac{1}{2} \sum_i^N \sum_j^N d_{i,j} \cdot r_{i,j}.$$

Данная оценка вычисляется после выполнения каждой итерации алгоритма размещения, и ее значение сравнивается со значением, вычисленным на предыдущей итерации. Принятая перестановка вершин должна приводить к уменьшению величины суммарной длины соединений, в противном случае вершины не меняют свое местоположение.

На основании рассмотренного математического аппарата решения задачи размещения с помощью последовательно-итерационного алгоритма запишем основные его шаги и процедуры.

1. На заданную монтажную конструкцию наложить координатную сетку G_r с осями координат s , t , т. е. построить граф.
2. Выполнить первоначальное размещение вершин графа G , интерпретирующего схему соединений элементов совместив их с узлами сетки G_r .
3. Для всех вершин графа G вычислить значения суммарных длин связей L_j .
4. Выбрать вершину x_i с $\max L_j$.
5. Вычислить для вершины x_i координаты центра тяжести s_c и t_c .
6. Определить подмножество вершин X' , вершины которого определяют область возможных перестановок с вершиной x_i .
7. Произвести парные перестановки вершин множества X' с вершиной x_i и вычислить для каждой вершины $x_j \in X'$, а также для вершины x_i величины L_j^v и L_i^v с учетом нового местоположения (L_j^v – это длина связей j -й вершины, совмещенной с узлом сетки v).

8. Вычислить отклонение $\sigma_j^v = L_j^v - L_j^k$ – для $x_j \in X'$ для всех вершин, выбранных в соответствии с п. 4 и 6, где v и k – соответственно «старая» и «новая» позиции местоположения вершин.
9. Вычислить значения $\delta_{ij} = \sigma_i^v + \sigma_j^a$ парных перестановок вершин множества X' и вершины x_i . Выбрать такую пару вершин, для которой значение δ_{ij} максимальное.
10. Произвести перестановку вершин.

Пример решения задачи размещения

Дан граф $G = (X, U)$. Его матрица смежности $R = \|r_{i,j}\|_{9 \times 9}$ имеет следующие значения элементов:

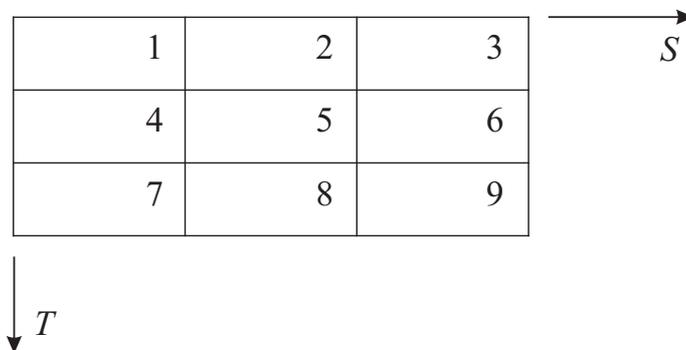
Значения элементов:

$$\begin{aligned} r_{1,4} = 3; r_{4,1} = 3; r_{2,3} = 2; r_{4,2} = 1; r_{1,7} = 2; r_{7,1} = 2; r_{2,4} = 1; r_{3,5} = 2; \\ r_{1,8} = 3; r_{8,1} = 3; r_{3,2} = 2; r_{5,3} = 2; r_{4,6} = 5; r_{6,7} = 5; r_{7,9} = 2; r_{7,8} = 3; \\ r_{6,4} = 5; r_{7,6} = 5; r_{9,7} = 2; r_{8,7} = 3; r_{5,6} = 2; r_{6,8} = 6; r_{5,9} = 4; r_{6,5} = 2; \\ r_{8,6} = 6; r_{9,5} = 4. \end{aligned}$$

Значение элемента $r_{i,j}$ матрицы R равно числу ребер, связывающих i -ю и j -ю вершины графа G .

Требуется элементы графа разместить на монтажной плоскости прямоугольной формы так, чтобы суммарная длина соединений была наименьшей.

Шаг 1. На монтажную плоскость накладываем граф-сетку G_r из 9 узлов с координатами $(S \times T)$. Примем, что координаты узла 1 имеют значения $(0,0)$.



Шаг 2. Отобразим граф G *схемы соединений элементов* в координатную сетку G_r так, чтобы номера его вершин совпадали с номерами узлов координатной сетки.



В общем случае номер вершины может не совпадать с номером узла (рис. 4.10a).

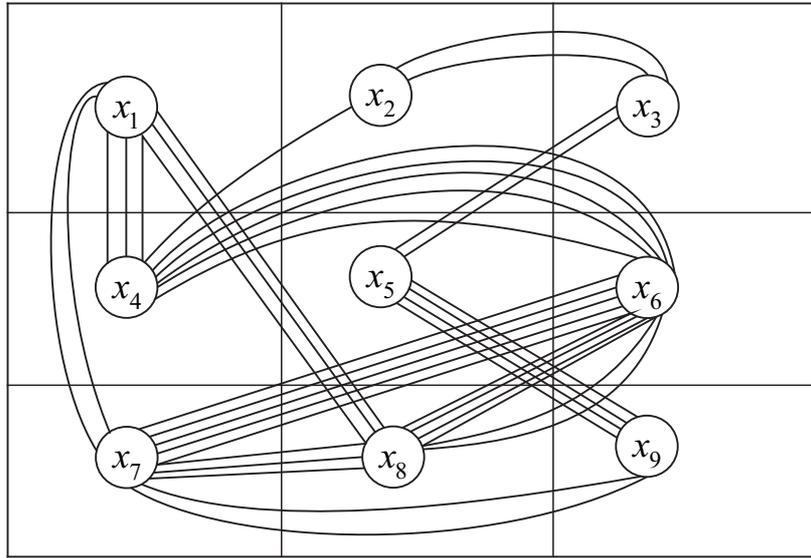


Рис. 4.10a – Отображение графа G соединений элементов схемы в координатной сетке G_r монтажной плоскости

Шаг 3. Определяем L_j для всех вершин $x_j \in X$. Значения $d_{i,j}$ определяем по формуле:

$$d_{ij} = |S_i - S_j| + |t_i - t_j|;$$

$$L_j = \frac{\sum_{i=1}^n d_{ij} r_{ij}}{\sum_{i=1}^n r_{ij}}, \quad i \neq j.$$

$$\begin{aligned} L_1 &= d_{14} \cdot r_{14} + d_{18} \cdot r_{18} + r_{17} \cdot d_{17}/r_{14} + r_{17} + r_{18} = \left\{ (10 - 01^1 + 10 - 11) \cdot 3 + \right. \\ &+ \left. (10 - 21^3 + 10 - 11) \cdot 3 + (10 - 01^2 + 10 - 21) \cdot 2 \right\} / 3 + 3 + 2 = \\ &= \{3 + 9 + 4\} / 8 = 16/8 = 2. \end{aligned}$$

Аналогично вычисляем:

$$L_2 \approx 1,3; L_3 \approx 1,5; L_4 \approx 1,7; L_5 \approx 1,7; L_6 \approx 2,2; L_7 \approx 2,2; L_8 \approx 2; L_9 \approx 2.$$

Вершина x_6 имеет наибольшую суммарную длину связей: $L_j = L_6 = 2,2$.

Шаг 4. Вычисляем координаты центра тяжести t_c , s_c для вершины x_6 , так как она имеет наибольшую суммарную длину связей L_j :

$$\left. \begin{aligned} t_c &= \sum_{j=1}^n t_j r_{i,j} / \rho(x_j), \\ s_c &= \sum_{j=1}^n S_j \cdot r_{i,j} / \rho(x_j). \end{aligned} \right\} \quad (4.28)$$

Для вычисления по формуле (4.28) выбираются все вершины, смежные x_6 . Это $x_4; x_5; x_7; x_8$, т. е. $j = \{4; 5; 7; 8\}$.

$$t_c^6 = (1 \cdot 5 + 1 \cdot 2 + 2 \cdot 5 + 2 \cdot 6) / (5 + 2 + 5 + 6) = 29 / 18 \approx 1,4;$$

$$s_c^6 = (0 \cdot 5 + 1 \cdot 2 + 0 \cdot 5 + 1 \cdot 6) / (5 + 2 + 5 + 6) = 8 / 18 \approx 0,4.$$

Шаг 5. По значениям координат (s_c^6 и t_c^6) определяем подмножество вершин, которые можно поменять местами с вершиной x_6 . Это все вершины, попавшие в радиус, равный 1 (из узла 4), – x_4, x_5, x_7, x_8 , так как они смежны вершине x_6 .

Шаг 6. Вычисляем новые значения $L_5^6, L_4^6, L_8^6, L_7^6$ для вершин x_4, x_5, x_7, x_8 , условно переставив каждую из них на место вершины x_6 , т. е. в узел 6, и значения $L_6^4, L_6^5, L_6^7, L_6^8$ для x_6 , условно поместив ее в узлы размещения вершин x_4, x_5, x_7, x_8 .

Пример вычисления L_4^6 :

$$L_4^6 = \frac{d_{64}r_{46} + d_{61}r_{41} + d_{62}r_{42}}{r_{46} + r_{41} + r_{42}} = \frac{2 \cdot 5 + 3 \cdot 3 + 2 \cdot 1}{5 + 3 + 1} = \frac{21}{9} \approx 2,3.$$

Аналогично вычисляются значения: $L_5^6 \approx 1$; $L_7^6 \approx 2,4$; $L_8^6 \approx 2,5$.

Пример вычисления L_6^4 :

$$L_6^4 = \frac{d_{46}r_{46} + d_{45}r_{65} + d_{47}r_{67} + d_{48}r_{68}}{r_{46} + r_{65} + r_{67} + r_{68}} \approx 1,6.$$

Аналогично вычисляются значения:

$$L_6^4 \approx 1,6; L_6^5 \approx 1,3; L_6^7 \approx 1,7; L_6^8 \approx 1,6.$$

Шаг 7. Вычисляем значения отклонений σ_j^y для вершин x_6 и $x_5; x_4; x_7; x_8$.

Для вершины x_6 :

$$\sigma_6^4 = L_6 - L_6^4 = 2,2 - 1,6 = 0,6;$$

$$\sigma_6^5 = L_6 - L_6^5 = 2,2 - 1,3 = 0,9;$$

$$\sigma_6^7 = L_6 - L_6^7 = 2,2 - 1,7 = 0,5;$$

$$\sigma_6^8 = L_6 - L_6^8 = 2,2 - 1,6 = 0,6;$$

для вершин $x_4; x_5; x_7; x_8$

$$\sigma_4^6 = L_4 - L_4^6 = 1,7 - 2,3 = -0,6;$$

$$\sigma_5^6 = L_5 - L_5^6 = 1,7 - 1 = 0,7;$$

$$\sigma_7^6 = L_7 - L_7^6 = 2,2 - 2,4 = -0,2;$$

$$\sigma_8^6 = L_8 - L_8^6 = 2 - 2,5 = -0,5.$$

Шаг 8. Вычисляем коэффициент $\delta_{ij} = \sigma_i^v + \sigma_j^a$ для вершин x_6 и $x_5; x_4;$

$x_7; x_8$

$$\delta_{4,6} = \sigma_6^4 + \sigma_4^6 = 0,6 + (-0,6) = 0;$$

$$\delta_{5,6} = \sigma_6^5 + \sigma_5^6 = 0,9 + 0,7 = 1,6;$$

$$\delta_{7,6} = \sigma_6^7 + \sigma_7^6 = 0,5 + (-0,2) = 0,3;$$

$$\delta_{8,6} = \sigma_6^8 + \sigma_8^6 = 0,6 + (-0,5) = 0,1.$$

Выбираем вершины с максимальным значением $\delta_{i,j}$.

Максимальное значение $\delta_{i,j} = \delta_{5,6}$ имеет взаимная перестановка вершин x_6 и x_5 . Следовательно, меняем местами вершины x_6 и x_5 .

На этом первая итерация алгоритма размещения заканчивается. Переходим к новой итерации, повторяя аналогичные вычисления, в соответствии с шагами 3–8.

Окончательный вариант размещения (рис. 4.10б) имеет суммарную длину $L(G) = 55$. Таким образом, суммарная длина уменьшилась с $L(G) = 76$ до $L(G) = 55$.

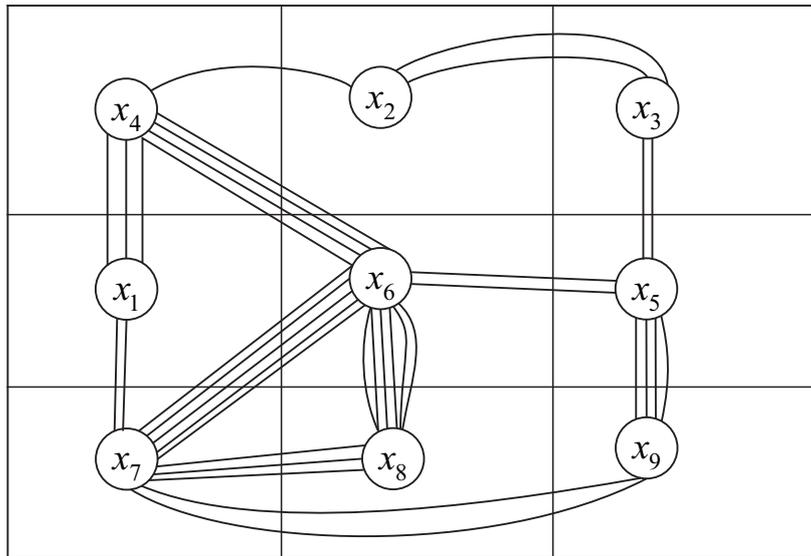


Рис. 4.10б – Окончательный вариант размещения вершин графа G в координатной сетке G_r монтажной плоскости

В случае, когда невозможно выбрать перестановку, уменьшающую длину соединений $L(G)$, делают перестановку групп вершин, которая выполняется с помощью так называемой *факторизации* графа по строкам и столбцам. Для этого каждой строке графа G_r ставят в соответствие вершину мультиграфа G_μ .

Так, графу, представленному на рисунке 4.11, соответствует мультиграф G_μ на рисунках 4.12–4.13.

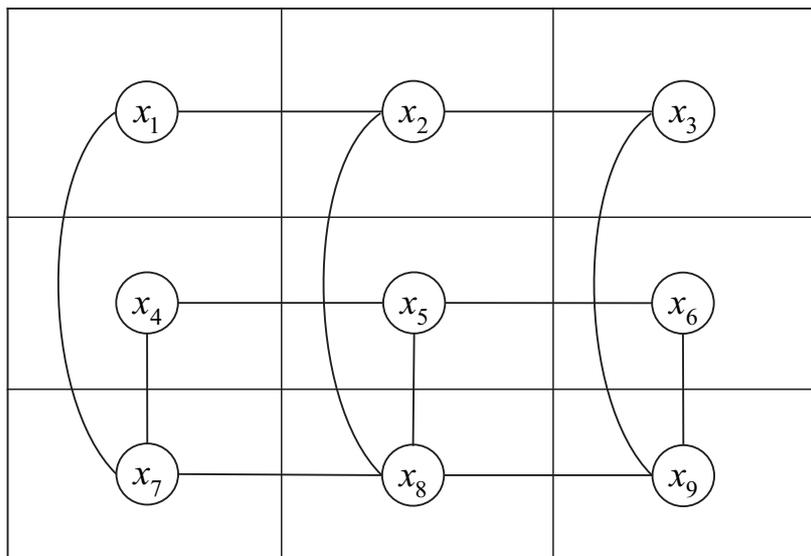


Рис. 4.11 – Размещение вершин графа G и их связей на монтажной плате

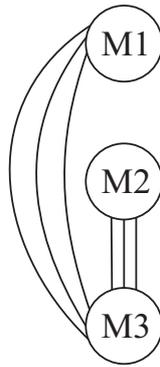


Рис. 4.12 – Мультиграф графа G ,
 $M1 (x_1, x_2, x_3)$, $M2 (x_4, x_5, x_6)$, $M3 (x_7, x_8, x_9)$



Рис. 4.13 – Мультиграф графа G после перестановки
на плате групп вершин $M1, M2, M3$

Далее «развернем» вершины мультиграфа, представленного на рисунке 4.13, соответственно вершинам графа на рисунке 4.11. Полученный граф, представленный на рисунке 4.14, соответствует лучшему размещению вершин графа с точки зрения меньшей суммарной длины соединений.

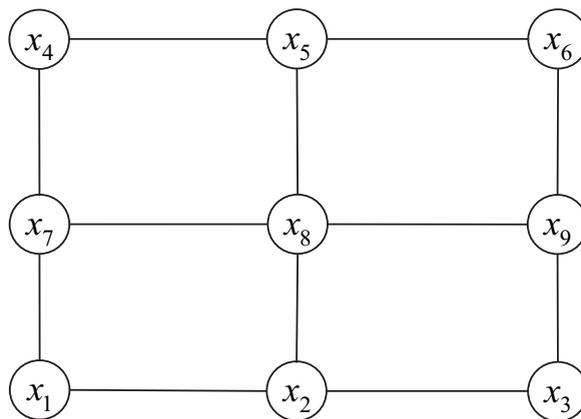


Рис. 4.14 – Улучшенный вариант размещения
вершин графа G на плате

Применение факторизации позволяет уменьшить общую длину соединений L_G дополнительно на 5–7%.

4.3 Трассировка соединений

Трассировка соединений является, как правило, заключительным этапом конструкторского проектирования ЭВА и состоит в определении линий, соединяющих эквипотенциальные контакты элементов, и компонентов, составляющих проектируемое устройство.

В зависимости от конструктивной реализации связей выделяют трассировку проводных соединений, трассировку печатных соединений, трассировку межсоединений на кристалле БИС.

4.3.1 Трассировка проводных соединений

Среди всех конструктивно-технологических реализаций связей проводной монтаж с точки зрения практического решения задачи трассировки наиболее прост. Объясняется это тем, что проводники изолированы друг от друга и не стоит задача об ограничениях на пересечения. Поэтому основным критерием трассировки проводных соединений является суммарная длина соединений.

Для проводного монтажа задача трассировки сводится к построению на вершинах графа дерева с минимальной суммарной длиной ребер. При этом вершины графа моделируют соединяемые контакты, а ребра – межсоединения. Единственное ограничение накладывается на максимальную локальную степень вершины, так как целесообразно подсоединять к одному контакту не более определённого конечного числа соединений.

Задача построения минимального дерева формулируется следующим образом. Пусть $P = \{p_1, p_2, \dots, p_n\}$ – множество точек плоскости, соответствующих выводам произвольной цепи. Рассмотрим полный граф $K_n = (X, U)$, вершины которого $x_i \in X$ соответствуют выводам цепи, а ребра $u_j \in U$ с приписанным к ним весом $\mu(u_j)$ характеризуют соединения между парами выводов. Значение $\mu(u_j)$ может представлять линейную комбинацию нескольких характеристик соединения:

$$\mu(u_j) = \mu_1 d_1(u_j) + \dots + \mu_s d_s(u_j),$$

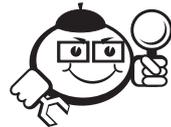
где μ_s – весовые коэффициенты; d_s – некоторая характеристика соединения u_j .

Теперь исходная задача сводится к определению в графе $G = (X, U)$ дерева, включающего все вершины $x_i \in X$ и имеющие минимальный суммарный

вес ребра, т. е. кратчайшее покрывающее дерево (КПД). Решение задачи дано в работах Крускала, Лобермана, Уйэнбергера и Прима.

Под расстоянием между вершинами графа будем понимать значение $\mu(u_j)$, приписанное соответственно ребрам графа, где $\mu(u_j)$ – весовая характеристика соединения между соответствующими парами контактов. Расстоянием от вершины до изолированного фрагмента будем считать минимум расстояний от отдельных вершин до фрагмента. Алгоритм Прима построения кратчайшего покрывающего дерева или цепи с n выводами может быть описан теперь следующим образом.

1. Для произвольного вывода цепи найти ближайший и провести соединение.
2. На каждом последующем шаге $i = 2, 3, \dots, n$ из множества не подсоединенных к цепи выводов выбрать тот, который находится ближе остальных (в указанном выше смысле) к группе уже связанных выводов, и подсоединить его к этой группе по кратчайшему пути.



Пример

Требуется соединить цепью эквипотенциальные контакты 1, 2, 3, 4, 5, расположенные на монтажной коммутационной плате (МКП), (рис. 4.15, а), так, чтобы к каждому контакту подходило не более заданного числа $\lambda = 3$ соединений.



Как правило, при разработке монтажных схем проводных соединений вводится ограничение на максимальное число соединений λ , подходящих к одному контакту. Если $\lambda > 6$, то необходимо использовать специальные алгоритмы построения КПД. Для решения такой задачи часто используются алгоритмы, основанные на методе ветвей и границ, однако при проведении большого числа цепей предпочтение отдают *эвристическим алгоритмам*.

Решение. Расстояние между контактами вычисляем в декартовой системе координат, принимая левую нижнюю точку платы за начало координат. Построение КПД выполним с помощью модифицированного алгоритма Прима:

1. Всякая изолированная вершина соединяется с ближайшей, не соединенной с другими вершинами.
2. Всякий изолированный фрагмент соединяется кратчайшим ребром с ближайшей вершиной, не соединенной с другими вершинами.

Иллюстрация построения КПД по алгоритму Прима приведена на рисунке 4.15 (а–д).

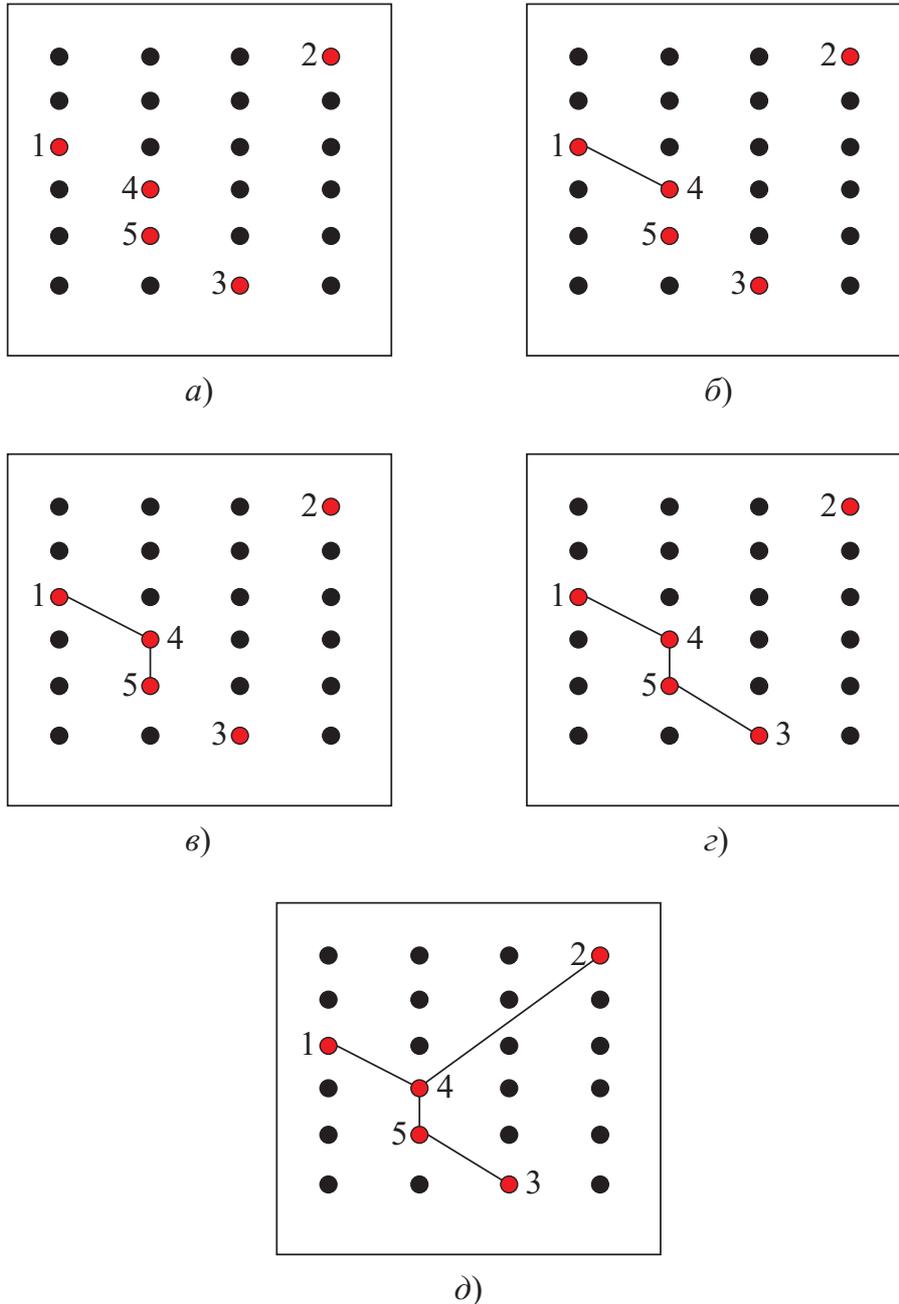


Рис. 4.15 – МКП (а); 1-й шаг трассировки (б); 2-й и 3-й шаги трассировки (в, г). Кратчайшее покрывающее дерево (д)

В некоторых случаях, помимо ограничения на степени вершин, задаются начальная и конечная точки цепи. Это имеет место, например, при разработке

монтажных схем для высокочастотных цепей, когда необходимо в определенной последовательности связать источник сигнала и несколько нагрузок. В этом случае задача сводится к построению кратчайшего пути между двумя заданными выводами, проходящего через все остальные выводы цепи. Данная задача родственна задаче о *маршруте коммивояжера*, а следуя терминологии теории графов это задача построения *кратчайшей гамильтоновой цепи* между заданными начальной и конечной вершинами.

Рассмотрим эвристический алгоритм, дающий приближенное решение этой задачи. Основой алгоритма являются $(n - 1)$ -шаговый процесс выбора кратчайших рёбер в полном графе K_n , проверка каждого ребра на выполнение ограничений задачи и составление из выбранных рёбер пути, соединяющего заданные точки.



Пример

На плате задано расположение точек 1, 2, 3, 4, 5, подлежащих соединению (рис. 4.16, а). Составим упорядоченную по возрастанию длин последовательность рёбер полного графа K_n : $(1n-3)$, $(1n-2k)$, $(2k-3)$, $(4-5)$, $(3-4)$, $(3-5)$, $(1n-4)$, $(2k-5)$, $(2k-4)$, $(1n-5)$. Отметим, что $1n$, $2k$ – соответственно начальная и конечная точки пути.

Очередное ребро $i = 1, 2, \dots, (n - 1)$ выбирается по порядку из этой последовательности при выполнении следующих условий:

1. Ребро не соединяет заданные конечную и начальную точки $(1n, 2k)$.
2. При включении ребра в путь степень вершин, соединяемых этим ребром, не превышает заданной ($\lambda = 1$ для начальной и конечной точек и $\lambda = 2$ для остальных точек).
3. Ребро не образует цикла с ребрами, уже включенными в путь.
4. При включении в путь любого ребра (кроме $(n - 1)$ -го) начальная и конечная точки остаются несвязанными.

Условия 1–3 непосредственно вытекают из ограничений задачи.

Условие 4 препятствует образованию тупиковых ситуаций, когда дальнейшее формирование пути становится невозможным – все подсоединенные точки, кроме начальной и конечной, имеют степень $\lambda = 2$.

Пошаговый процесс формирования связывающей цепи с заданными

начальной и конечной вершинами изображен на рисунке 4.16 (а–д).

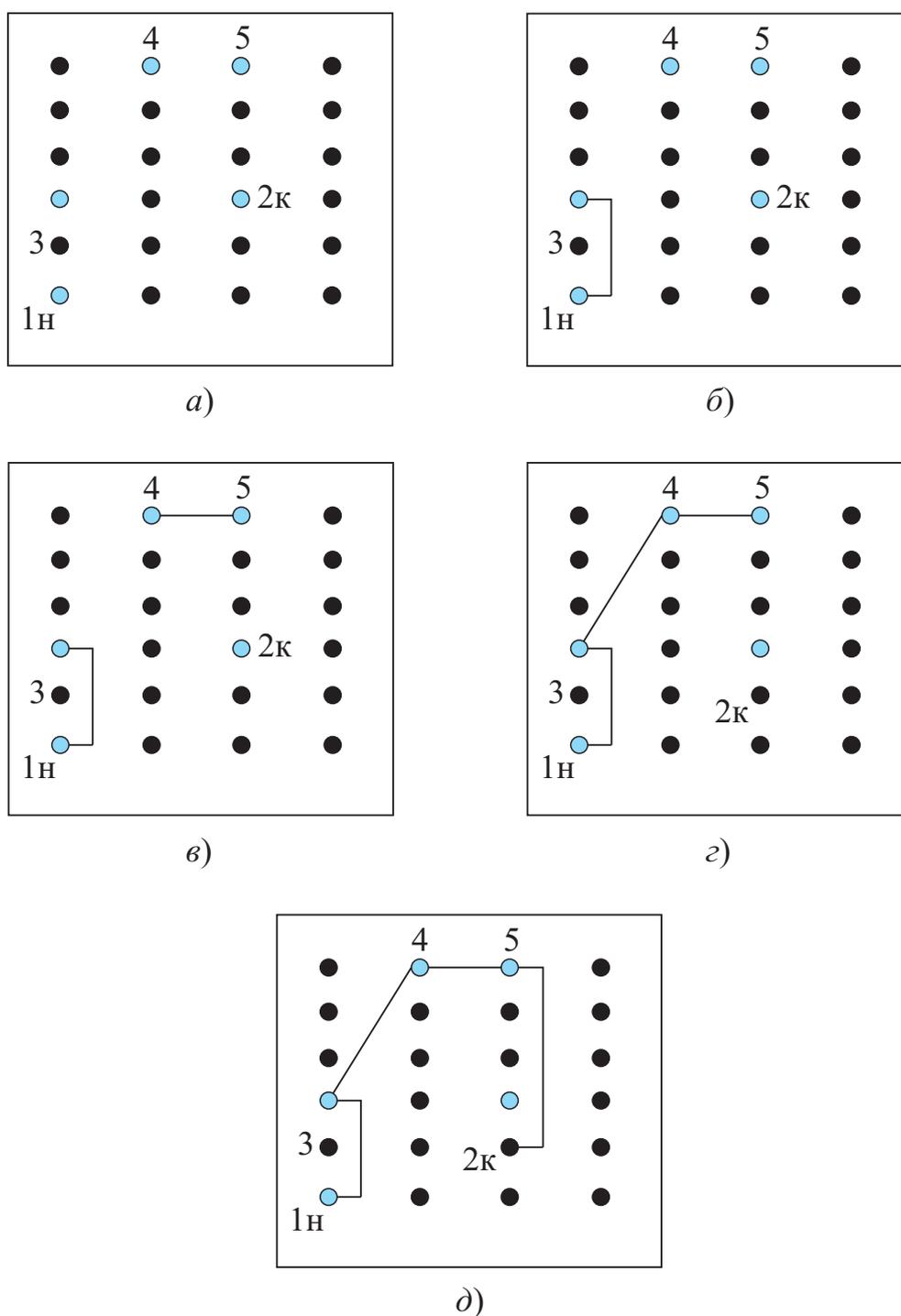


Рис. 4.16 – Построение КПД на монтажно-коммутационной плате с заданными начальной (1Н) и конечной (2к) точками (а): 1-й шаг (б); 2-й шаг (в); 3-й шаг (г); 4-й шаг – КПД (д)

Деревья Штейнера

Математической моделью технической задачи проектирования топологии электрических соединений на этапе глобальной трассировки является задача построения минимальных связывающих деревьев:

- без дополнительных вершин (деревья Прима – Краскала);
- с дополнительными вершинами (деревья Штейнера).

Если входы и выходы элементов считать вершинами в графе, а электрические проводники считать ребрами графа с весом, равным длине проводника, то отыскание оптимальной конфигурации электрической цепи сводится к задаче Штейнера. Специфика САПР электронной аппаратуры заключается в том, что для вычисления длин рёбер дерева используется ортогональная метрика.

Постановка задачи: есть несколько точек на плоскости, которые нужно связать системой соединений наименьшей суммарной длины таким образом, чтобы по этим соединениям можно было из каждой точки достичь любую другую. Число точек конечно. Это направление построения деревьев Штейнера получило название *сети Штейнера*.

Данная задача именуется «задача построения кратчайшей сети, или минимального дерева Штейнера (МДШ)».

При проектировании печатных соединений применяются связывающие деревья двух видов: кратчайшие связывающие сети либо деревья специального вида, называемые деревьями Штейнера. Кратчайшие связывающие сети могут строиться с помощью алгоритма Прима по кратчайшим расстояниям (см. рис. 14.5, 14.6) для проводного монтажа либо в ортогональной метрике, используемой в печатных платах. В ортогональной метрике при отыскании местоположения дополнительных вершин можно ограничиться узлами ортогональной сетки, построенной на основных вершинах. Степени вершин γ такого дерева ограничиваются соотношением $3 \leq \gamma \leq 4$. Формирование сети Штейнера (МДШ) изображено на рисунках 4.17, 4.18.

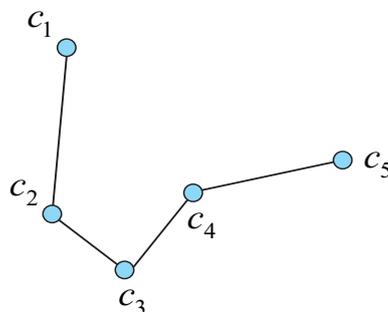


Рис. 4.17 – Кратчайшая связывающая сеть в декартовой метрике

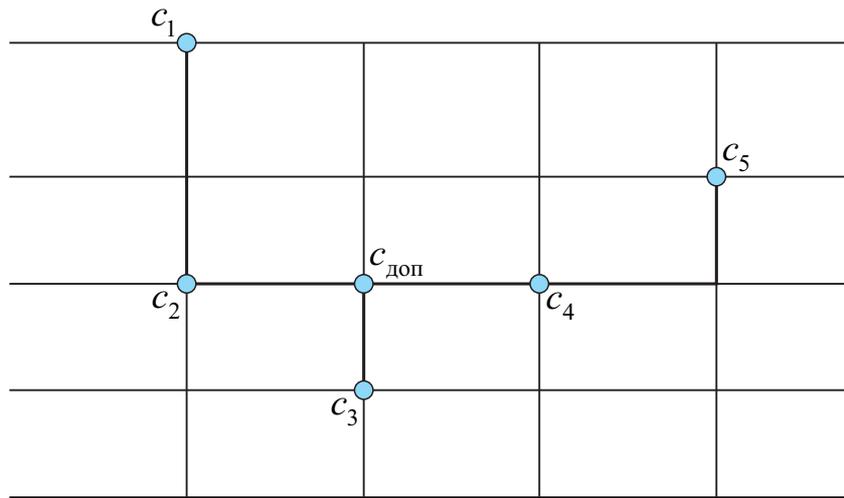


Рис. 4.18 – Дерево Штейнера в ортогональной метрике

Задача построения МДШ усложняется, если при соединении множества вершин (контактов цепи) разрешается использование дополнительных точек соединения. В общем виде данная задача формулируется так: для заданных x_1, x_2, \dots, x_n точек плоскости построить КПД с $n' \geq n$ вершинами. Данная задача называется задачей Штейнера. Обычно решение задачи Штейнера рассматривают в областях прямоугольной конфигурации. Можно сказать, что для $n \leq 5$ известно решение задачи Штейнера. В общем же случае известны условия, которым должны удовлетворять деревья Штейнера (ДШ) и эвристические алгоритмы. Дополнительные точки, вводимые при построении КПД, называют точками Штейнера (ТШ).

Приведем ряд лемм, полученных А. Фридманом и П. Менонем, дающих необходимые условия существования и построения деревьев Штейнера.



.....

Лемма 4.1. Для n вершин, которые должны быть соединены между собой, всегда можно построить дерево Штейнера, в котором каждая точка Штейнера g_i соединена по крайней мере с тремя другими.

.....

.....



Лемма 4.2. Для множества вершин $\{x_1, x_2, x_3\}$, которые должны быть соединены, координаты ТШ g , минимизирующей

$\sum_{i=1}^3 d(g, n_i)$, будут S_{cp} , T_{cp} и справедливо выражение

$$\sum_{i=1}^3 d(g, n_i) = 1/2 P(x_1, x_2, x_3),$$

где S_{cp} , T_{cp} – средние значения S_i ($i=1,2,3$) и t_j ($j=1,2,3$); $P(x_1, x_2, x_3)$ – длина периметра прямоугольника, построенного на вершинах x_1, x_2, x_3 ; $d(g, n_i)$ – прямоугольное расстояние между g и n_i .

.....

На рисунке 4.19 показан пример, иллюстрирующий условия леммы 4.2 при построении дерева Штейнера, соединяющего вершины x_1, x_2, x_3 . Точка Штейнера g имеет координаты $S_g = 3$, $T_g = 5$.

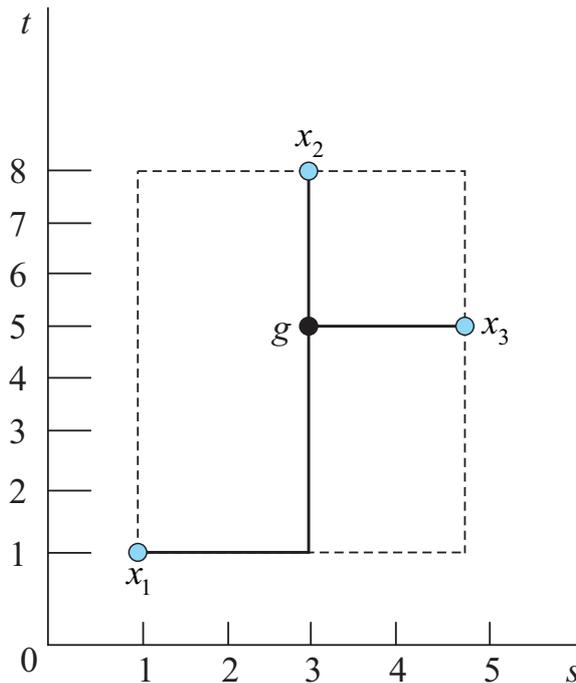


Рис. 4.19 – Пример использования леммы 4.2

.....



Лемма 4.3. Дано дерево Штейнера, состоящее из x_i точек плоскости. Пусть точка Штейнера g соединена только с тремя вершинами x_1, x_2, x_3 . Тогда g находится внутри прямоугольника, построенного на точках x_1, x_2, x_3 , и точка Штейнера единственна.

.....

Следствие 4.1. Для заданного множества вершин x_1, x_2, x_3 дерево Штейнера содержит одну точку Штейнера g с координатами (s_{cp}, t_{cp}) , если она не совпадает ни с одной из вершин x_1, x_2, x_3 , и суммарная длина рёбер дерева Штейнера равна $\frac{1}{2}P(x_1, x_2, x_3)$.



Лемма 4.4. Дано дерево Штейнера на множестве $X = \{x_1, x_2, x_3\}$, а точки Штейнера этого дерева образуют множество $g = \{g_1, g_2, \dots, g_k\}$. Если $g_l \subset g$ и $g_i \in I$, то g_l содержит элемент $g_{1,k}$, соединенный не менее чем с двумя вершинами из I , в противном случае g_l пусто. Здесь I – множество точек пересечения линий координатной сетки.

На основе приведенных лемм сформулирована известная теорема:



Теорема 4.1. Дано дерево Штейнера (ДШ) на множестве вершин $X = \{x_1, x_2, \dots, x_n\}$ с множеством точек Штейнера (ТШ) $g = \{g_1, g_2, \dots, g_k\}$. Тогда для этого множества существует такое другое ДШ с множеством ТШ g' , что

$$(\forall g'_i \in g'), (g'_i \in I).$$

Из рассмотрения теоремы следует метод определения ДШ на основе полного перебора в качестве ТШ всех возможных подмножеств точек из I . Очевидно, что такой метод применим только при построении ДШ для небольшого числа ТШ. В этой связи разрабатываются эвристические процедуры построения квазимиимальных деревьев Штейнера.

Процедура 1

1. Все вершины $x_i \in X$ проектируются на ось $S(T)$.
2. Расстояние от наименьшей до наибольшей координаты делится пополам, и из этой точки $i \in I$ или $[i]$ при $i \notin I$ проводится перпендикуляр (столб Штейнера).

3. Из каждой вершины $x_i \in X$ опускается перпендикуляр до пересечения со столбом Штейнера.
4. Построено дерево Штейнера. Конец работы алгоритма.

Данный алгоритм очень прост, но в смысле минимума суммарной длины проводников результаты далеки от совершенства.

Процедура 2 (алгоритм Ханана) построения деревьев Штейнера

1. Все множество вершин $x_i \in X$ разбивается на классы S_0, S_1, \dots, S_i в порядке возрастания координаты S_i так, чтобы у всех вершин одного класса была одинаковая координата S_i .
2. Анализируется множество вершин класса S_0 , и соединяются перпендикуляром все вершины этого множества с точкой S_{\min} оси S .
3. Образуется множество I' , состоящее из всех точек I , которые были до этого соединены с произвольной вершиной x_i , включая и вершины x_i строящегося дерева Штейнера.
4. Выбирается следующее множество S_{i+1} , имеющее наименьшую координату S . Определяются точка $i_m \in I'$ и вершина $x_j \in S_{i+1}$, для которых

$$d(i_m, x_j) \leq d(i_q, x_j); \quad \forall i_q \in I', \forall x_j \in S_{i+1}.$$

Другими словами, x_j находится на кратчайшем расстоянии от рассматриваемого фрагмента дерева Штейнера в классе S_{i+1} . Точка i_m с координатами (s_m, t_m) соединяется с вершиной x_j с координатами (s_j, t_j) двузвенной ломаной линией. При этом звенья параллельны координатным осям S и T .

5. Для каждой вершины $x_k \in S_{i+1}$ производятся операции, аналогичные п. 4. При этом вершина x_k соединяется с ближайшей в I' .
6. Пункты 4 и 5 повторяются для всех по порядку множеств S_j до построения дерева Штейнера.

На рисунке 4.20 показана реализация процедуры 2 (алгоритм Ханана) с упорядочиванием вершин в п. 1 в порядке убывания координат S .

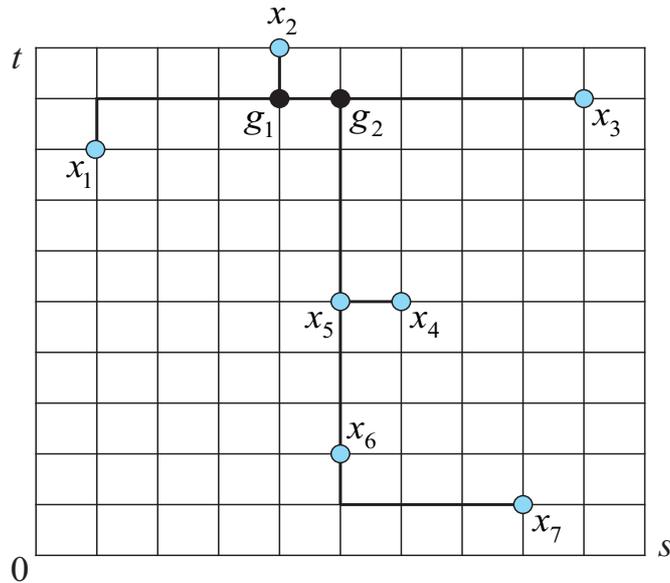


Рис. 4.20 – Пример построения ДШ



.....

Существует большое число модификаций описанных процедур. Можно проводить упорядочивание координат не по увеличению $S(T)$, а по уменьшению $S(T)$.

.....

Для построения можно использовать методы поиска в глубину, поиска в ширину или метод ветвей и границ, в зависимости от требований на временную сложность алгоритма.

4.3.2 Трассировка печатных соединений

Волновой алгоритм и его модификации. Для проведения соединений используют следующие алгоритмы или их комбинации и модификации: 1) волновой алгоритм или поиск пути в лабиринте; 2) лучевой алгоритм; 3) алгоритм канальной трассировки.

Наиболее универсальным является волновой алгоритм. Он позволяет найти путь во всех случаях, когда он существует. Однако волновой алгоритм требует больших затрат памяти ЭВМ и времени по сравнению с другими алгоритмами.

Качество алгоритмов оценивается, прежде всего, процентом неоттрассированных соединений.

Рассмотрим волновой алгоритм построения пути между двумя заданными точками коммутационного поля.

Коммутационное поле преобразуется в дискретное рабочее поле (ДРП).

Пусть A_x, A_y – размеры коммутационного поля; $h = h_1 + h_2$ – константа, равная сумме наименьшей ширины проводника и наименьшего расстояния между двумя проводниками, h – дискрет ДРП. Тогда $\{A_x / h\}, \{A_y / h\}$ – число дискретов по осям OX, OY ; $\{\dots\}$ – символ близлежащего большего целого.

Каждую ячейку ДРП можно представить центральной точкой $\alpha - (x, y)$ в системе координат ДРП.

Для точки α определим список соседних точек при допущении трассировки по четырём направлениям (рис. 4.21, а: $N(\alpha) = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$).

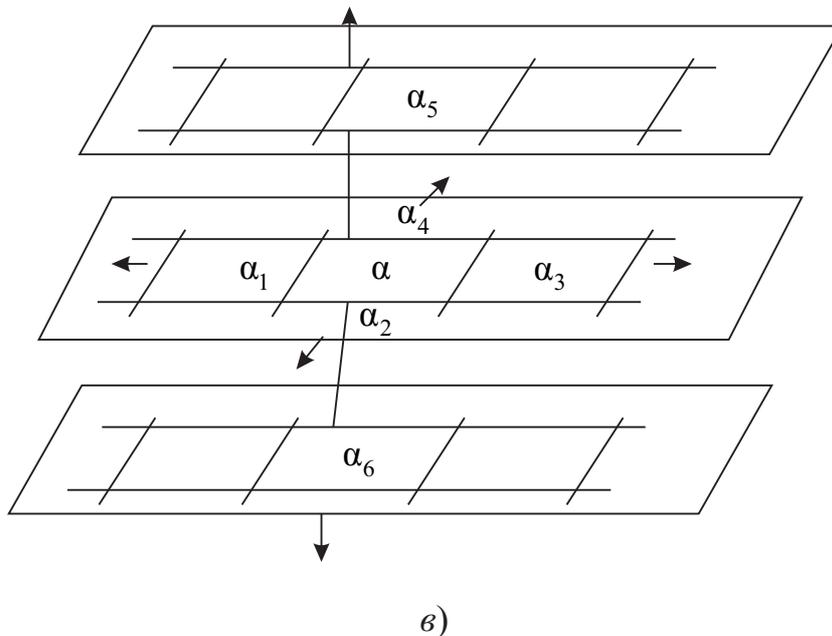
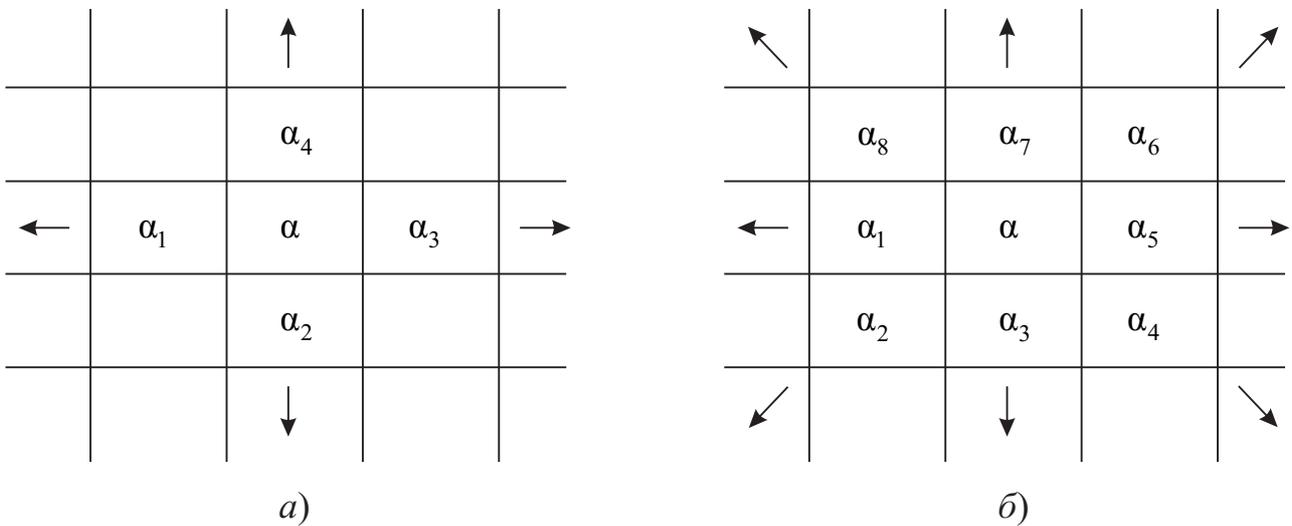


Рис. 4.21 – Множество соседних с α ячеек ДРП: а) – при использовании четырёх направлений; б) – при использовании восьми направлений; в) – при использовании шести направлений (трехмерный случай)

На рисунке 4.21, б показана модель МКП для трассировки по ортогональным направлениям и для допущения трассировки под углом 45° , т. е. по восьми направлениям: $N(\alpha) = (\alpha_1, \alpha_2, \dots, \alpha_8)$.

В качестве первой точки α_i множества N выбирается точка, лежащая слева от α , хотя такой выбор не является обязательным. Порядок расположения точек в списке N обычно соответствует обходу соседних с α точек против часовой стрелки. В рассматриваемом варианте выбор α_i производят из $N(\alpha)$, где

$$\alpha = (x, y), \alpha_1 = (x - 1, y), \alpha_2 = (x, y - 1), \alpha_3 = (x + 1, y), \alpha_4 = (x, y + 1).$$

Каждый элемент ДРП помечают признаком «занят», «свободен» или «соединяемые ячейки».

Первоначально в число занятых ячеек помещают области, запрещенные для трассировки, – выводы элементов, собственно элементы, технологические области, границы коммутационного поля. В процессе работы алгоритма к ним добавляют области, занятые построенными соединениями.

Волновой алгоритм трассировки. Основой алгоритма является построение кратчайшего пути на ДРП между двумя известными ячейками. Первую из них называют источником, вторую – целью. Алгоритм включает в себя этапы распространения волны и проведения пути (рис. 4.21).

Распространение волны. Пусть α_0 и α_t – точки источника и цели. Моделируется числовая волна вокруг источника. Волна представляется последовательностью фронтов Φ_k ($k = 1, 2, \dots$):

$$\begin{aligned} \Phi_1(\alpha_0) &= N(\alpha_0); \\ \Phi_k(\alpha_0) &= \bigcup_{\alpha \in \Phi_{k-1}(\alpha_0)} N(\alpha) \setminus \bigcup_{i=1}^{k-1} \Phi_i, \quad k = 2, 3, \dots \end{aligned}$$

Из списка $N(\alpha)$ рассматриваются только свободные ячейки. Ячейкам ДРП присваивается число k – номер фронта Φ_k , которым они захватываются.

Если очередная выбранная во фронт Φ_k ячейка является целью α_t , распространение волны заканчивается, если нет – продолжается.

На некотором шаге распространения волны может оказаться, что $\Phi_k = \emptyset$ (ни одна точка не включена во фронт). Это означает, что отсутствует доступ к ячейке-цели и соединение построить нельзя.

В этом случае пара точек (источник – цель) заносится в список непроведенных соединений, с ДРП стирается информация о волне (о номерах фронтов) и осуществляется переход к распространению волны для следующей пары ячеек (источник – цель).

Если же ячейка-цель достигнута, то осуществляется переход к следующему этапу.

Проведение пути. Пусть ячейка-цель $\alpha_t \in \Phi_m(\alpha_0)$. Рассматривают множество $N(\alpha_t) \cap \Phi_{m-1}(\alpha_0)$.

Выбирают некоторую точку $\alpha_{m-1} \in N(\alpha_t) \cap \Phi_{m-1}(\alpha_0)$, затем точку $\alpha_{m-2} \in N(\alpha_{m-1}) \cap \Phi_{m-2}(\alpha_0)$. Процесс повторяется, пока не достигается точка α_0 .

Последовательность точек $\alpha_t, \alpha_{m-1}, \dots, \alpha_1, \alpha_0$ определяет искомый путь длиной в m дискретов. Рисунок 4.22 иллюстрирует работу алгоритмов.

10	9	8	7	X	3	2	1	2	3	4	5
9	8	7	6	X	X	1	A	1	2	3	4
8	7	6	5	4	3	2	1	2	3	4	5
9	8	7	6	5	4	X	X	X	4	5	6
10	9	8	7	6	5	6	7	6	5	6	7
X	X	X	X	7	6	7	8	7	6	7	8
	X	B	9	8	7	8	9	8	7	8	9
	X			9	8	9		9	8	9	
	X				9				9		

Рис. 4.22 – Трасса, связывающая точки А – В, рассчитанная с помощью волнового алгоритма

Алгоритм Ли имеет следующие особенности:

- 1) если между точками α_0 и α_t путь существует, то его всегда можно найти;
- 2) из множества возможных путей можно определить кратчайший при заданном отношении занятости ячеек ДРП;

- 3) реализация алгоритма проста, но требуются значительные ресурсы памяти;
- 4) специфика расположения элементов и особенности конструкции учитываются выбором отношения занятости, не требуя модификации алгоритма.

При модификации волнового алгоритма преследуются следующие цели:

- а) уменьшение требуемых ресурсов памяти и машинного времени;
- б) усложнение понятий источника и цели для лучшего учета конструкторско-технологических ограничений и критерия;
- в) учет многокритериальности.

Использование путевых координат. Путевой координатой ячейки α_i фронта Φ_k называют направление на соседнюю с ней ячейку фронта Φ_{k-1} , по которой α_i получает свой вес (номер фронта).

Введение путевых координат имеет следующие преимущества: а) путь определяется однозначно; б) требуется два разряда памяти для метки фронта волны путевыми координатами.

Метод встречных волн. Поочередно моделируется каждый фронт для связываемых точек A и B (на каждом нечетном шаге A – источник, B – цель, на каждом четном – наоборот).

На каждом шаге включения очередной точки во фронт волны проверяется условие соприкосновения фронтов от разных источников. При соприкосновении фронтов распространение волны прекращается. Из общей точки γ строятся пути к точкам A и B . Если $\gamma \in \Phi_k(A)$, $\gamma \in \Phi_m(B)$, то искомым путем составляется из двух последовательностей ячеек $\gamma, \alpha_{k-1}, \alpha_{k-2}, \dots, \alpha_0 = A$ и $\gamma, \beta_{m-1}, \beta_{m-2}, \dots, \beta_0 = B$. Заметим, что k от m может отличаться не более чем на единицу.

Данный алгоритм уменьшает время распространения волны, так как площадь, захватываемая волной, в идеальном случае на ДРП без преград уменьшается вдвое. Действительно, если расстояние от A до B равно R , то в модели с однонаправленной волной площадь захватываемого ею ДРП составляет πR^2 , а в модели со встречными волнами площадь

$$2\pi(R/2)^2 = \pi R^2/2.$$

Существуют модификации метода встречных волн с k источниками A_1, A_2, \dots, A_k ($k > 2$), позволяющие уменьшать время распространения волны в

k раз. Преимуществом такого подхода является и то, что можно трассировать цепь целиком, пропуская этап построения списка двухконцевых соединений.

Заметим, что реальный выигрыш во времени несколько меньше указанного из-за усложнения алгоритма.

Ограничение области распространения волны. Выбирают прямоугольник, содержащий точки A и B , за пределы которого волна не распространяется. В простейшем случае A и B – концы диагонали прямоугольника $\Pi(A, B)$. Множество точек k -го фронта принимает следующий вид:

$$\Phi_k = \left(\bigcup_{\alpha \in \Phi_{k-1}} N(\alpha) \setminus \bigcup_{i=1}^{k-1} \Phi_i \right) \cap \Pi(A, B).$$

В этом случае уменьшается время работы алгоритма за счет увеличения процента нетрассируемых соединений.



Контрольные вопросы по главе 4

1. Назовите уровни разукрупнения РЭС по функциональной сложности.
2. Приведите алгоритмы решения задачи компоновки.
3. Приведите алгоритмы решения задачи размещения.
4. Приведите алгоритмы решения задачи трассировки.
5. Расскажите о графовых моделях МКП для решения задачи размещения.
6. Расскажите о графовых моделях МКП для решения задачи трассировки.
7. Перечислите способы измерения расстояния между соединяемыми контактами модулей с координатами x_i, x_j и y_i, y_j .
8. Приведите алгоритмы разбиения графа на части.
9. Что понимается под оптимальным разбиением графа на части?
10. Приведите алгоритмы размещения конструктивных элементов РЭС в монтажном пространстве.
11. Назовите способы измерения расстояния между соединяемыми контактами модулей с координатами x_i, x_j и y_i, y_j .
12. Приведите математическую модель задачи покрытия типовыми модулями.

5 Системы автоматизированного проектирования ЭС

Сложность и трудоемкость проектируемых изделий, сжатые сроки проектирования, высокие требования к качеству разработок требуют новых совершенных технологий и систем для выполнения задач проектирования.

Это, прежде всего, комплексное применение компьютеров, современных математических методов в проектировании, на базе которых возможно создание систем автоматизированного проектирования (САПР).

Применение САПР особенно эффективно при создании сложных объектов и технических систем.

5.1 Общая характеристика процесса проектирования и базовые определения

Проектирование – это не только создание идеи построения объекта, но и обоснование способа его реализации.

Проектирование можно рассматривать как целенаправленную последовательность актов принятия проектных решений, приводящих к построению описания проектируемого объекта с заданной степенью детализации.

Некоторые особенности проектной деятельности:

- 1) продуктом проектирования является упорядоченная совокупность сведений, служащих знаковой моделью объекта, реально еще не существующей в момент проектирования;
- 2) нестабильность и неопределенность постановки задачи, ее изменчивость до завершения процесса проектирования;
- 3) проектирование, как правило, носит итерационный многовариантный характер.

Процесс проектирования реализуется в соответствии с определенным планом, который принято представлять в виде логической схемы (логического графа) построения проекта, на которой видна очередность проектных процедур. Под проектной процедурой понимается элементарное действие, совершаемое конструктором в процессе проектирования.

Стратегия проектирования – это поиск и определение последовательности операций с целью решения поставленной задачи. Стратегия подразумевает выбор и использование методов и алгоритмов проектирования.

Технология проектирования – это опробованная последовательность действий или операций, позволяющая технически выполнить проектирование заданного объекта.

При проектировании сложных объектов с помощью САПР применяется метод *декомпозиции*, т. е. разбиения.

Уровни декомпозиции:

- *системный* – наиболее общее описание назначения объекта и его связей;
- *архитектурный* – описание структуры объекта;
- *функциональный* – описание законов функционирования подсистем объекта;
- *конструктивный* – детальный выбор и описание всех элементов системы (объекта).

В составе САПР можно выделить следующие основные подсистемы:

- 1) *информационная подсистема*, предназначена для сбора, хранения, поиска, упорядочения, пополнения и выдачи всей необходимой для обеспечения процесса проектирования информации;
- 2) *подсистема поиска решений технической задачи обеспечивает*:
 - анализ проектной ситуации и определение направления поиска решения;
 - поиск решения задачи;
 - форматирование критерия оценки вариантов;
 - выбор рационального или оптимального варианта;
- 3) *подсистема инженерного анализа* (моделирование объекта и оптимизация его характеристик). Ее назначение:
 - выполнение всех вычислительных работ, связанных с детализацией выбранного варианта решения проектной задачи;
- 4) *подсистема ведения и изготовления документации* [1, 5].

Назначение – обеспечение получения проектных документов (чертежей, технических описаний, схем, графиков, таблиц и т. п.), необходимых для создания объекта проектирования. (Это составляет обычно 45–60% всех затрат. В то же время это нетворческая, рутинная, однообразная работа.) Автоматизация этого этапа дает наибольший эффект.

Автоматизация проектирования технических систем особенно эффективна, когда от автоматизации выполнения отдельных инженерных расчетов пере-

ходят к комплексной автоматизации, создавая для этой цели САПР. Примером подобной САПР может служить интегрированная САПР PCAD для автоматизации проектирования и подготовки производства печатных плат. Она позволяет выполнить операции по созданию библиотек компонентов, графического ввода чертежей принципиальной схемы и чертежа печатной платы проектируемого устройства; проектированию программируемых логических матриц; выпуску конструкторской документации и технологической информации на проектируемую печатную плату и многое другое.

Перечислим важнейшие технические требования, которым должна удовлетворять современная САПР печатных плат, в порядке их значимости:

- Полная русификация системы, включая диалоговые окна, меню, подсказки, системные сообщения, руководство пользователя.
- Поддержка системой сквозного проектирования – от создания принципиальной схемы и трассировки платы до подготовки выходных данных (фотошаблонов, программ для сверлильных и сборочных автоматов) в реальном времени и в рамках единого проекта.
- Возможность адаптации к внедренной на предприятии технологии проектирования и производства печатных плат и устройств.
- Наличие импортных и отечественных баз данных элементов.
- Наличие функций автоматизированного размещения элементов и автоматической трассировки печатной платы.
- Наличие интерфейса с имеющимся оборудованием.
- Автоматизированный выпуск комплекта конструкторской документации в полном соответствии с ЕСКД.
- Возможности импорта/экспорта с другими аналогичными системами (в том числе с системами для моделирования печатных плат) через перечень элементов и список цепей [1, 3, 4].

Большинство современных САПР в той или иной степени удовлетворяют вышеперечисленным требованиям. Поэтому следующее требование, которое прямо или косвенно предъявляется к САПР печатных плат, – «экономическое», учитывающее соотношение «цена/возможности». Интегрированная САПР PCAD – это наиболее популярная в мире система автоматизации проектирования и подготовки производства печатных плат. Она позволяет выполнить операции по созданию библиотек компонентов; графический ввод чертежа принципиальной схемы и чертежа печатной платы проектируемого устройства; выпуск кон-

структурной документации и технологической информации на проектируемую печатную плату.

5.2 Принципы построения САПР

Система автоматизированного проектирования (САПР) – это организационно-техническая система, представляющая собой комплекс средств автоматизированного проектирования, связанный с подразделениями проектной организации, и выполняющая автоматизированное проектирование. Структура САПР показана на рисунке 5.1.

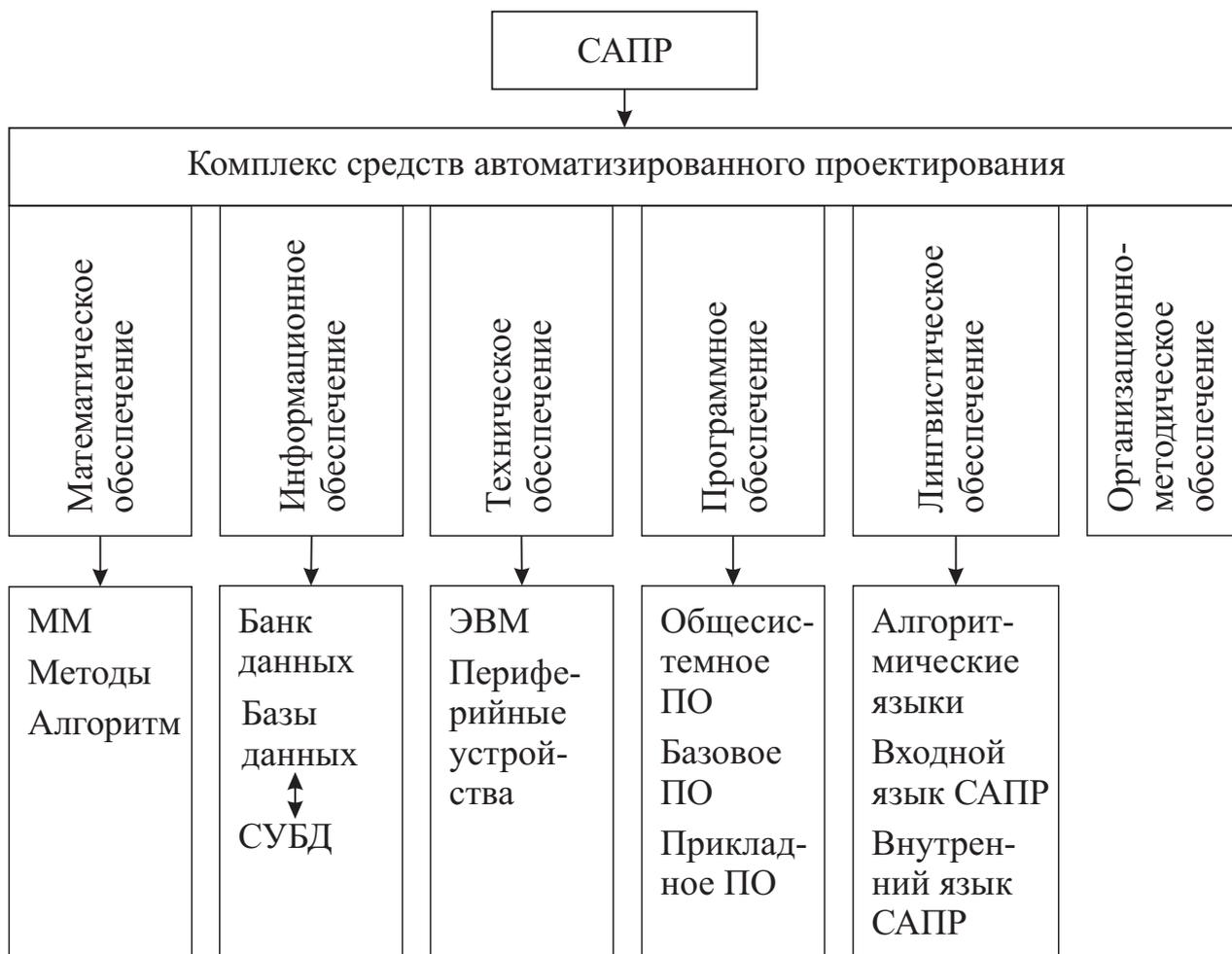


Рис. 5.1 – Структура САПР

В состав САПР входят следующие виды различного обеспечения процесса автоматизированного проектирования:

- математическое обеспечение (МО);
- техническое обеспечение (ТО);
- программное обеспечение (ПО);
- информационное обеспечение (ИО);

- лингвистическое обеспечение (ЛО);
- организационно-методическое обеспечение.

Математическое обеспечение представляет собой совокупность математических методов, ММ и алгоритмов, необходимых для организации автоматизированного проектирования.

Техническое обеспечение представляет собой взаимосвязанные технические средства, содержащие устройства вычислительной и организационной техники, средства передачи данных, измерительные и другие устройства.

Программное обеспечение САПР объединяет собственно программы для систем обработки данных на машинных носителях и программную документацию, необходимую для эксплуатации САПР. ПО САПР включает в себя прикладные, базовые и общесистемные программы.

Общесистемное ПО предназначено для организации функционирования технических средств и представлено операционными системами ЭВМ и ВС. Общесистемное ПО обычно создается для многих приложений и специфику САПР не отражает. Базовое и прикладное ПО создается для нужд САПР. В *базовое ПО* входят программы, обеспечивающие правильное функционирование прикладных программ. В *прикладном ПО* реализуется математическое обеспечение для непосредственного выполнения проектных процедур. Прикладное ПО обычно имеет форму пакетов прикладных программ, каждый из которых обслуживает определенный этап процесса проектирования или группу однотипных задач внутри различных этапов.

Информационное обеспечение САПР объединяет всевозможные данные, необходимые для выполнения автоматизированного проектирования. Основная составная часть ИО САПР – *банк данных*, представляющий собой совокупность средств для централизованного накопления и коллективного использования данных в САПР. Банк данных состоит из базы данных и системы управления базой данных.

База данных – сами данные, находящиеся в запоминающих устройствах ЭВМ и структурированные в соответствии с принятыми в данном банке данных правилами. *Система управления базой данных (СУБД)* представляет собой совокупность программных средств, обеспечивающих функционирование банка данных. С помощью СУБД производятся запись данных в банк, их выборка по запросам пользователей и прикладных программ, обеспечивается защита от несанкционированного доступа и т. д.

Лингвистическое обеспечение САПР представлено совокупностью языков, применяемых для описания процедур автоматизированного проектирования и проектных решений. Основная часть ЛО – языки общения человека с ЭВМ. В состав ЛО входят *внутренний и входной языки* самой САПР.

Организационно-методическое обеспечение САПР включает в себя документы, устанавливающие состав и правила эксплуатации средств обеспечения автоматизированного проектирования, а также другие документы, регламентирующие организационную структуру подразделений проектной организации и взаимодействие этих подразделений с комплексом средств автоматизированного проектирования.

С функциональной точки зрения САПР позволяет решать задачи моделирования объектов на этапе НИР, анализа и синтеза устройств, выпуска конструкторской документации. САПР позволяет моделировать устройства на уровне структурных, функциональных, принципиальных схем, а также выпускать соответствующую техническую документацию.

Современные САПР строятся на основе *следующих основных принципов*:

- *САПР – человеко-машинная система.* Коллектив разработчиков является составной частью САПР, выполняя проектные работы во взаимодействии с ЭВМ. Таким образом, при работе в САПР происходит ориентация не на автоматическое, а на автоматизированное проектирование.
- *Комплексная автоматизация всех уровней проектирования.* При этом достигается основная цель автоматизации: уменьшение материальных затрат и времени, повышение качества проектирования сложных объектов, повышение производительности труда.
- *Информационная согласованность подсистем и программ проектирования.* Все программы, входящие в состав САПР, на различных этапах проектирования должны использовать одинаковую структуру данных, единообразный входной язык.
- *Открытость САПР.* Данный принцип подразумевает возможность модификации системы во время ее эксплуатации. Внесение новых изменений в систему должно быть упрощено и доступно любому пользователю – это увеличивает универсальность САПР и продлевает ее жизненный цикл.
- *Совместимость традиционного и автоматизированного проектирования.* Выходная документация САПР должна соответствовать требо-

ваниям ЕСКД и ГОСТ. Это значительно упрощает внедрение САПР на действующем предприятии со сложившейся структурой, формами технической документации, не нарушая функционирования предприятия.

Задачи, решаемые САПР:

- совершенствование проектирования на основе применения современных математических методов и средств вычислительной техники;
- автоматизация процесса поиска, обработки и выдачи информации;
- создание единых банков данных, содержащих сведения справочного характера;
- повышение качества и уменьшение времени оформления документации за счет автоматизации нетворческих работ;
- унификация и стандартизация методов проектирования [1, 4, 6].

Важные проблемы, требующие решения при создании комплексных САПР – управление сложностью проектов и интеграция ПО. Эти проблемы включают вопросы декомпозиции проектов, распараллеливания проектных работ, целостности данных, межпрограммных интерфейсов и др.

5.3 Системы автоматизированного проектирования РЭС и их место среди других автоматизированных систем

Этапы жизненного цикла промышленных изделий

Жизненный цикл промышленных изделий включает ряд этапов, начиная от зарождения идеи нового продукта до утилизации по окончании срока его использования. Основные этапы жизненного цикла промышленной продукции представлены на рисунке 5.2. К ним относятся этапы проектирования, технологической подготовки производства (ТПП), собственно производства, реализации продукции, эксплуатации и, наконец, утилизации.

На всех этапах жизненного цикла изделий имеются свои целевые установки. При этом участники жизненного цикла стремятся достичь поставленных целей с максимальной эффективностью. На этапах проектирования, ТПП и производства нужно обеспечить выполнение ТЗ при заданной степени надежности изделия и минимизации материальных и временных затрат, что необходимо для достижения успеха в конкурентной борьбе в условиях рыночной экономики. Понятие эффективности включает в себя не только снижение

себестоимости продукции и сокращение сроков проектирования и производства, но и обеспечение удобства освоения и снижения затрат на будущую эксплуатацию изделий. Особую важность требования удобства эксплуатации имеют для сложной техники, например в таких отраслях, как авиа- или автомобилестроение.

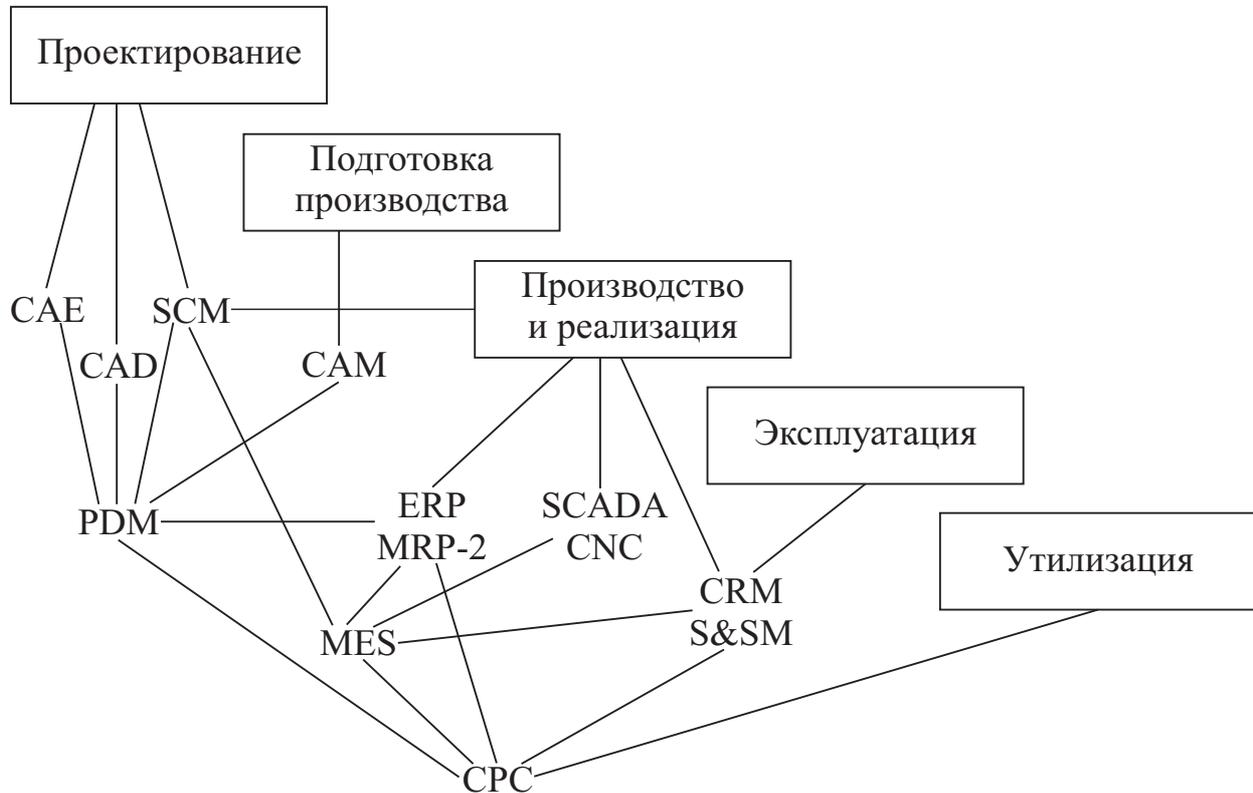


Рис. 5.2 – Этапы жизненного цикла промышленных изделий и используемые АС

Достижение поставленных целей на современных предприятиях, выпускающих сложные промышленные изделия, оказывается невозможным без широкого использования автоматизированных систем (АС), основанных на применении компьютеров и предназначенных для создания, переработки и использования всей необходимой информации о свойствах изделий и сопровождающих процессов. Специфика задач, решаемых на различных этапах жизненного цикла изделий, обуславливает разнообразие применяемых АС. Основные типы АС с их привязкой к тем или иным этапам жизненного цикла изделий указаны на рисунке 5.2.

Автоматизация проектирования осуществляется под управлением САПР. Принято выделять в САПР радиоэлектронной отрасли промышленности системы функционального, конструкторского и технологического проектирования.

Первые из них называют системами расчетов и инженерного анализа, или системами CAE (Computer Aided Engineering). Системы конструкторского проектирования называют системами CAD (Computer Aided Design).

Проектирование технологических процессов составляет часть технологической подготовки производства и выполняется в системах CAM (Computer Aided Manufacturing). Функции координации работы систем CAE/CAD/CAM, управления проектными данными и проектированием возложены на систему управления проектными данными PDM (Product Data Management). Уже на стадии проектирования требуются услуги системы управления цепочками поставок (SCM – Supply Chain Management), иногда называемой системой Component Supplier Management (CSM). На этапе производства эта система управляет поставками необходимых материалов и комплектующих.

Информационная поддержка этапа производства продукции осуществляется автоматизированными системами управления предприятием (АСУП) и автоматизированными системами управления технологическими процессами (АСУТП). К АСУП относятся системы планирования и управления предприятием ERP (Enterprise Resource Planning), планирования производства и требований к материалам MRP-2 (Manufacturing Requirement Planning), производственная исполнительная система MES (Manufacturing Execution Systems), а также SCM и система управления взаимоотношениями с заказчиками CRM (Customer Requirement Management).

Наиболее развитые системы ERP выполняют различные бизнес-функции. Они связаны с планированием производства, закупками, сбытом продукции, анализом перспектив маркетинга, управлением финансами, персоналом, складским хозяйством, учетом основных фондов и т. п. Системы MRP-2 ориентированы главным образом на бизнес-функции, непосредственно связанные с производством, а системы MES – на решение оперативных задач управления проектированием, производством и маркетингом.

На этапе реализации продукции выполняются функции управления отношениями с заказчиками и покупателями, проводится анализ рыночной ситуации, определяются перспективы спроса на планируемые изделия. Эти функции осуществляет система CRM. Маркетинговые задачи иногда возлагаются на систему S&SM (Sales and Service Management), которая, кроме того, используется для решения проблем обслуживания изделий. На этапе эксплуатации применяют также специализированные компьютерные системы, занятые вопросами ремонта, контроля, диагностики эксплуатируемых систем.

Автоматизированные системы управления технологическими процессами контролируют и используют данные, характеризующие состояние технологического оборудования и протекание технологических процессов. Именно их чаще всего называют системами промышленной автоматизации.

Для выполнения диспетчерских функций (сбор и обработка данных о состоянии оборудования и технологических процессов) и разработки ПО для встроенного оборудования в состав АСУТП вводят систему SCADA (Supervisory Control and Data Acquisition). Непосредственное программное управление технологическим оборудованием осуществляют с помощью системы CNC (Computer Numerical Control) на базе контроллеров (специализированных компьютеров, называемых промышленными), которые встроены в технологическое оборудование.

В последнее время усилия многих компаний, производящих программно-аппаратные средства АС, направлены на создание систем электронного бизнеса (E-Commerce). Задачи, решаемые системами E-Commerce, сводятся не только к организации на сайтах Internet витрин товаров и услуг. Они объединяют в едином информационном пространстве запросы заказчиков и данные о возможностях множества организаций, специализирующихся на предоставлении различных услуг и выполнении тех или иных процедур и операций по проектированию, изготовлению, поставкам заказанных изделий. Такие системы E-Commerce называют системами управления данными в интегрированном информационном пространстве CPC (Collaborative Product Commerce) или PLM (Product Lifecycle Management). Проектирование непосредственно под заказ позволяет добиться наилучших параметров создаваемой продукции, а оптимальный выбор исполнителей и цепочек поставок ведет к минимизации времени и стоимости выполнения заказа. Характерная особенность CPC – обеспечение взаимодействия многих предприятий, т. е. технология CPC является основой, интегрирующей информационное пространство, в котором функционируют САПР, ERP, PDM, SCM, CRM и другие АС разных предприятий.

Структура САПР

Как и любая сложная система, САПР состоит из подсистем. Различают подсистемы проектирующие и обслуживающие.

Проектирующие подсистемы непосредственно выполняют проектные процедуры. Примерами проектирующих подсистем могут служить подсистемы геометрического трехмерного моделирования механических объектов, изготов-

ления конструкторской документации, схмотехнического анализа, трассировки соединений в печатных платах.

Обслуживающие подсистемы обеспечивают функционирование проектирующих подсистем, их совокупность часто называют системной средой (или оболочкой) САПР. Типичными обслуживающими подсистемами являются подсистемы управления проектными данными, подсистемы разработки и сопровождения программного обеспечения CASE (Computer Aided Software Engineering), обучающие подсистемы для освоения пользователями технологий, реализованных в САПР.

Виды обеспечения САПР

Структурирование САПР по различным аспектам обуславливает появление видов обеспечения САПР. Принято выделять семь видов обеспечения САПР:

- *техническое* (ТО), включающее различные аппаратные средства (ЭВМ, периферийные устройства, сетевое коммутационное оборудование, линии связи, измерительные средства);
- *математическое* (МО), объединяющее математические методы, модели и алгоритмы для выполнения проектирования;
- *программное*, представляемое компьютерными программами САПР;
- *информационное*, состоящее из базы данных, СУБД, а также включающее другие данные, которые используются при проектировании; отметим, что вся совокупность используемых при проектировании данных называется информационным фондом САПР, база данных вместе с СУБД носит название банка данных;
- *лингвистическое*, выражаемое языками общения между проектировщиками и ЭВМ, языками программирования и языками обмена данными между техническими средствами САПР;
- *методическое*, включающее различные методики проектирования; иногда к нему относят также математическое обеспечение;
- *организационное*, представляемое штатными расписаниями, должностными инструкциями и другими документами, которые регламентируют работу проектного предприятия [1, 2, 4].

Классификацию САПР осуществляют по ряду признаков, например по приложению, целевому назначению, масштабам (комплексности решаемых задач), характеру базовой подсистемы – ядра САПР.

По приложениям наиболее представительными и широко используемыми являются следующие группы САПР:

- САПР для применения в отраслях общего машиностроения. Их часто называют машиностроительными САПР или системами MCAD (Mechanical CAD);
- САПР для радиоэлектроники: системы ECAD (Electronic CAD) или EDA (Electronic Design Automation);
- САПР в области архитектуры и строительства.

Кроме того, известно большое число специализированных САПР, или выделяемых в указанных группах, или представляющих самостоятельную ветвь классификации. Примерами таких систем являются САПР больших интегральных схем (БИС); САПР летательных аппаратов; САПР электрических машин и т. п.

По целевому назначению различают САПР или подсистемы САПР, обеспечивающие разные аспекты (страты) проектирования. Так, в составе MCAD появляются рассмотренные выше CAE/CAD/CAM-системы.

По масштабам различают отдельные программно-методические комплексы (ПМК) САПР, например: комплекс анализа прочности механических изделий в соответствии с методом конечных элементов (МКЭ) или комплекс анализа электронных схем; системы ПМК; системы с уникальными архитектурами не только программного (software), но и технического (hardware) обеспечения.

По характеру базовой подсистемы различают следующие разновидности САПР:

1. САПР на базе подсистемы машинной графики и геометрического моделирования. Эти САПР ориентированы на приложения, где основной процедурой проектирования является конструирование, т. е. определение пространственных форм и взаимного расположения объектов. К этой группе систем относится большинство САПР в области машиностроения, построенных на базе графических ядер. В настоящее время широко используют унифицированные графические ядра, применяемые более чем в одной САПР (ядра Parasolid фирмы EDS Urographies и ACIS фирмы Intergraph).
2. САПР на базе СУБД. Они ориентированы на приложения, в которых при сравнительно несложных математических расчетах перерабатывается большой объем данных. Такие САПР преимущественно встречаются в технико-экономических приложениях, например при проекти-

ровании бизнес-планов, но они имеются также при проектировании объектов, подобных щитам управления в системах автоматики.

3. САПР на базе конкретного прикладного пакета. Фактически это автономно используемые ПМК, например имитационного моделирования производственных процессов, расчета прочности по МКЭ, синтеза и анализа систем автоматического управления и т. п. Часто такие САПР относятся к системам САЕ. Примерами могут служить программы логического проектирования на базе языка VHDL, математические пакеты типа MathCAD [1, 2, 4].

Комплексные (интегрированные) САПР, состоящие из совокупности подсистем предыдущих видов. Характерными примерами комплексных САПР являются САЕ/CAD/CAM-системы в машиностроении или САПР БИС. Так, САПР БИС включает в себя СУБД и подсистемы проектирования компонентов, принципиальных, логических и функциональных схем, топологии кристаллов, тестов для проверки годности изделий. Для управления столь сложными системами применяют специализированные системные среды.

5.4 Основы построения автоматизированных систем технологической подготовки производства

В производственной практике объектом *автоматизации проектирования* является вся совокупность действий проектировщиков, разрабатывающих изделие или *технологический процесс*, или то и другое, и оформляющих результаты разработок в виде конструкторской, технологической и эксплуатационной документации.

В производственном процессе под технологией понимают систему взаимосвязанных способов обработки материалов и приемов изготовления продукции.

Понятие CALS-технологии

CALS-технология – это технология комплексной компьютеризации сфер промышленного производства, цель которой – унификация и стандартизация спецификаций промышленной продукции на всех этапах ее жизненного цикла. Основные спецификации представлены проектной, технологической, производственной, маркетинговой и эксплуатационной документацией. В CALS-системах предусмотрены хранение, обработка и передача информации в компьютерных средах, оперативный доступ к данным в нужное время и в нужном

месте. Соответствующие системы автоматизации называли автоматизированными логистическими системами, или CALS (Computer Aided Logistic Systems). Поскольку под логистикой обычно понимают дисциплину, посвященную вопросам снабжения и управления запасами, а функции CALS намного шире и связаны со всеми этапами жизненного цикла промышленных изделий, применяют и более соответствующую предмету расшифровку аббревиатуры CALS – Continuous Acquisition and LifeCycle Support.

Применение CALS позволяет существенно сократить объемы проектных работ, так как описания многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в базах данных сетевых серверов, доступных любому пользователю технологии CALS. Существенно облегчается решение проблем ремонтпригодности, интеграции продукции в системы и среды различного рода, адаптации к меняющимся условиям эксплуатации, специализации проектных организаций и т. п. Развитие CALS-технологии должно привести к появлению так называемых виртуальных производств, при которых процесс создания спецификаций с информацией для программно управляемого технологического оборудования, достаточной для изготовления изделия, может быть распределен во времени и пространстве между многими организационно автономными проектными студиями.

Среди несомненных достижений CALS-технологии следует отметить легкость распространения передовых проектных решений, возможность многократного воспроизведения частей проекта в новых разработках. Построение открытых распределенных автоматизированных систем для проектирования и управления в промышленности составляет основу современной CALS-технологии. Главная проблема их построения – обеспечение единообразного описания и интерпретации данных, независимо от места и времени их получения в общей системе, имеющей масштабы вплоть до глобальных.

Структура проектной, технологической и эксплуатационной документации, языки ее представления должны быть стандартизованными. Тогда становится реальной успешная работа над общим проектом разных коллективов, разделенных во времени и пространстве и использующих разные системы САПР. Одна и та же конструкторская документация может быть использована многократно в разных проектах, а одна и та же технологическая документация адаптирована к разным производственным условиям, что позволяет существенно сократить и удешевить общий цикл проектирования и производства. Кроме того, упрощается эксплуатация систем. Таким образом, информационная инте-

грация является неотъемлемым свойством CALS-систем. Поэтому в основу CALS-технологии положен ряд стандартов, обеспечивающих такую интеграцию. Важные проблемы, требующие решения при создании комплексных САПР – управление сложностью проектов и интеграция ПО. Эти проблемы включают вопросы декомпозиции проектов, распараллеливания проектных работ, целостности данных, межпрограммных интерфейсов и др.

CASE-технология

CASE-технология – программный комплекс, автоматизирующий технологический процесс анализа, проектирования, разработки и сопровождения сложных программных систем.

CASE-технология представляет собой методологию проектирования ИС, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область.

CASE-технология – это инструментарий для системных аналитиков, разработчиков и программистов, заменяющий бумагу и карандаш компьютером, автоматизируя процесс проектирования и разработки ПО.

CASE-технология в рамках методологии включает в себя методы, с помощью которых на основе графической нотации строятся диаграммы, поддерживаемые инструментальной средой.

Отличие CASE-технологии от традиционной технологии состоит в следующем.

Исторически так сложилось, что при проектировании новых объектов структурный аспект отражается в виде схем (принципиальные электрические схемы, механические схемы, комбинированные схемы и т. п.). CASE-технологии иначе ещё называют структурным системным анализом, что отражает тот факт, что и функционирование объекта (в рассматриваемом случае разрабатываемого ПО) отражается в различных схемах, таблицах, диаграммах, матрицах, картах и т. п.

Большинство CASE-технологий основано на парадигме – методология/метод/нотация/средство. Под *нотацией* понимаются правила формализованного описания структуры системы, элементов данных и других ее компонентов с помощью схем, диаграмм, формальных и естественных языков. Например, в качестве миниспецификаций некоторые CASE-технологии используют таблицы решений. Правила построения такой таблицы и представляют собой нотацию.

Средство – это инструментарий поддержки методов. Реализуются средства в ПО, которое создает среду разработчика ПО. Эти инструменты поддерживают работу пользователей-разработчиков при создании и редактировании проекта в интерактивном режиме, они выполняют проверки соответствия компонентов и кодируют на некотором языке программирования модули ПО (кодогенерация).

Следует отметить, что используемые в методологии структурного анализа средства весьма разнообразны. Наиболее часто и эффективно используются следующие:

- DFD (Data Flow Diagrams) – диаграммы потоков данных, совместно со словарями данных и спецификациями процессов или миниспецификациями;
- ERD (Entity-Relationship Diagrams) – диаграммы «сущность – связь»;
- STD (State Transition Diagrams) – диаграммы переходов состояний.

Современные структурные методологии анализа и проектирования классифицируются по следующим признакам:

- по отношению к школам – Software Engineering (SE) и Information Engineering (IE);
- по порядку построения моделей – процедурно-ориентированные, ориентированные на данные и информационно-ориентированные;
- по типу целевых систем – для систем реального времени и для информационных систем.

Основная особенность систем реального времени заключается в том, что они контролируют и контролируются внешними событиями: своевременное реагирование на эти события (отклик должен следовать не позднее регламентированного временного интервала) – основная и первоочередная функция таких систем. Другие отличия информационных систем от систем реального времени сведены в таблицу 5.1.

Таблица 5.1 – Отличия информационных систем от систем реального времени

Информационные системы	Системы реального времени
Управляются данными	Управляются событиями
Сложные структуры данных	Простые структуры данных
Большой объем входных данных	Малое количество входных данных

5.4.1 Задачи автоматизации технологической подготовки производства

За первыми системами в области автоматизации (САПР) последовали автоматизированные системы технологической подготовки производства (АСТПП) и автоматизированные системы управления технологическими процессами (АСУТП), причем последние были вызваны к жизни необходимостью автоматизации всех этапов создания новых изделий электронной техники: проектирования (САПР), подготовки производства (АСТПП), самого производства (АСУТП). Автоматизация работ на каждом этапе вначале развивалась автономно, сейчас появилась необходимость и возможность интеграции трех видов систем с целью создания интегрированных компьютеризованных производств.

Единая система технологической подготовки производства – это установленная государственными стандартами система организации и управления процессом технологической подготовки производства, предусматривающая широкое применение прогрессивных типовых технологических процессов, стандартной технологической оснастки и оборудования, средств механизации и автоматизации производственных процессов, инженерно-технических и управленческих работ.

При автоматизации технологической подготовки производства учитываются характер и взаимосвязь факторов, влияющих на построение технологического процесса и определяющих заданное качество изготавливаемых изделий и экономическую эффективность разрабатываемой технологии. С этой целью проводятся моделирование, структурная и параметрическая оптимизация технологического процесса, принимается целесообразное сочетание типовых и индивидуальных технологических решений на всех уровнях подготовки производства.

Возможность автоматизации технологической подготовки производства изделий электронной техники (ИЭТ) определяется развитием научных основ технологии ИЭТ, математических методов, технических средств и техники программирования. От закономерностей и положений, заложенных при автоматизации технологической подготовки производства (ТПП), от их практической и научной ценности зависит степень совершенства разработанного технологического процесса, который во многом определяет качество изделия. От использования научных основ технологии производства ИЭТ во многом зависит эффективность применения АСТПП.

Основные положения и определения

Процесс технологической подготовки производства изделий электронной техники заключается в разработке различной технологической и плановой документации, изготовлении и отладке технологической оснастки, в приобретении материалов и покупных комплектующих изделий, организации рабочих мест и всего технологического и производственного процесса, в разработке календарных планов и проведении ряда других технических и организационных мероприятий, необходимых для реализации технологических процессов изготовления отдельных деталей и сборки изделий. Из всех перечисленных мероприятий определяющим является проектирование технологических процессов. При автоматизации проектирования и управления производством часть элементов технологической подготовки производства передается САПР и АСУ предприятием. В самостоятельную группу выделяют элементы, связанные с проектированием технологических процессов и конструированием специальной технологической оснастки. Рассмотренную часть называют технологической подготовкой производства.

Развитие САПР и вычислительной техники, использование их в процессе подготовки управляющей информации для станков, конвейеров и всего производства в целом привели к разработке и созданию автоматизированных систем технологической подготовки производства (АСТПП). Под автоматизированными системами технологической подготовки производства понимают совокупность методов, алгоритмов, программного и математического обеспечения, технических средств и организационных мероприятий, объединенных для технологической подготовки производства. В отличие от автоматических систем, действующих без участия человека по заранее подготовленной программе, системы технологической подготовки производства являются автоматизированными, т. е. в них некоторые операции и действия производит ЭВМ, а некоторые – оператор. По мере развития программных и технических средств все больше функций будет передаваться вычислительным средствам, но при контроле оператора.

Создание АСТПП основывается на ряде современных теоретических и практических дисциплин. При создании АСТПП удобно воспользоваться отождествлением процессов и задач проектирования с некоторыми процессами и задачами автоматического управления, с задачами обработки информации системами, моделирующими некоторые процессы мышления человека при технологической подготовке производства. Автоматизированная система тех-

нологической подготовки производства является сложной по структуре и функционированию кибернетической системой, находящейся в постоянном движении.

Проектирование технологической подготовки производства – сложный и разветвленный процесс переработки информации самого разнообразного вида, формы и содержания. Информацию, используемую при технологической подготовке производства, называют *технологической*. Технологическая информация делится на неизменяемую и изменяемую. *Неизменной* называется информация, которая остается постоянной при решении множества задач одного класса в одних условиях производства и меняется только при переходе к изготовлению ИЭТ другого класса. К ней относятся сведения о содержании ГОСТов, нормативов, стандартное математическое обеспечение, таблицы, константы процессов, данные об установленном оборудовании, наличии на складах материалов, нормативов, комплектующих и т. д. Неизменная информация хранится в банках данных.

Изменяемой называется информация, меняющаяся при решении каждой задачи. Например, данные о детали, которые после проектирования технологии ее обработки изымаются из оперативной информационной системы АСТПП и заменяются данными по следующей детали.

Назначение АСТПП

Основная цель создания АСТПП – ускорение и совершенствование процессов технологического проектирования производства за счет широкого использования САПР и средств вычислительной техники.

Технологическая подготовка производства подразумевает выполнение функций, подразделяемых на основные группы:

- 1) обеспечение технологичности конструкции ИЭТ;
- 2) разработка технологических процессов;
- 3) проектирование и изготовление технологической оснастки;
- 4) организация и управление технологической подготовкой производства.

Структура целей АСТПП, отвечающая вышеперечисленным функциям, представлена на рисунке 5.1. Рассмотрим назначение основных структурных составляющих технологической подготовки производства.

Система управления технологической подготовкой связывает все подсистемы и способствует их согласованному функционированию. Данная система

может осуществлять связь с другими системами управления производством, например с системой календарного и оперативного планирования.

Система формирования исходных данных служит для подготовки нормативных и текущих сведений, необходимых для использования в различных подсистемах технологической подготовки производства. Система включает технические средства сбора и передачи информации, а также нормативное бюро, анализирующее документы и информационные потоки и разрабатывающее различные нормативы и классификаторы, нормативные данные.

Информационно-поисковая система предназначена для хранения и корректировки информативных данных, которые неоднократно используются различными подсистемами технологической подготовки производства, а также для быстрого поиска и выдачи информации в соответствии с поисковым предписанием.

Система подготовки технологической документации обеспечивает привязку объектов производства к технологическим процессам и операциям.

В системах специального назначения наиболее трудоемкой является система проектирования технологии, отвечающая за разработку технологических процессов основного производства, необходимой оснастки, инструмента и технологических процессов их изготовления.

Хотя в перечисленных системах используются многие общие стандартные методы и программы проектирования, они отличаются друг от друга результатами проектирования, способами оформления исходящей информации и характером решаемых задач. Общими для систем являются структуры подсистем, наименования и назначения составляющих их блоков, последовательность решения задач технологического проектирования и структуры технологических алгоритмов.

Во всех подсистемах проектирования технологических процессов следующая последовательность решения задач:

- анализ и подготовка исходных данных для принятия решений, определяющих технологический процесс;
- выбор технологических процессов, операций и переходов, необходимых для изготовления деталей, модулей, микросборок, удовлетворяющих техническим условиям;
- определение наиболее рационального технологического маршрута создания изделия;

- выбор универсального оборудования, приспособлений, инструментов и другой технологической оснастки;
- в случае отсутствия универсального оборудования и оснастки происходит подготовка заданий другим подсистемам на конструирование;
- определение оптимальных режимов обработки и технически обоснованных норм времени;
- группирование технологических процессов по общности технологии, оборудования и оснастки;
- выборка и формирование данных для автоматизированных систем управления производством;
- оформление и выдача технической документации, необходимой для производственной реализации спроектированного технологического процесса.

Конструирование технологической оснастки (приспособлений, инструментов, прессформ и т. д.) базируется на использовании стандартных, нормализованных и унифицированных конструкций, узлов, деталей и проектных решений. Задачи конструирования оснастки:

- выбор конструкций, решений и проектов из имеющихся типовых;
- выбор типовых элементов конструкций, выполнение проверочных расчетов и т. д.;
- компоновка конструкций из типовых элементов и проверка правильности размещения;
- доработка отдельных типовых элементов в соответствии с требованиями компоновки и привязки оснастки к сборочному автомату;
- технико-экономические расчеты, связанные с выбором конструкций элементов и компоновочных решений;
- оформление конструкторской документации;
- разработка в полном объеме технологических процессов изготовления специальных деталей оснастки [4, 5].

Подсистема моделирования, оптимизации и технико-экономических расчетов, входящая в систему проектирования технологии, предназначена для математического моделирования процессов технологического проектирования и их оптимизации в процессе моделирования. Так как решающим критерием оптимальности полученных при проектировании результатов является экономическая эффективность от их реализации, то в этой же подсистеме выполняется

ряд технико-экономических расчетов. Моделирование может выполняться как внутри подсистем АСТПП при решении отдельных задач, так и для всей АСТПП для оптимизации всего процесса технологической подготовки производства.

Структура АСТПП и основные требования к ней

Автоматизированная система технологической подготовки производства при своём полном развитии должна обеспечивать автоматизированное решение разнообразных задач технологического проектирования. Для этого необходима разработка достаточного количества функциональных систем и подсистем. Примерный состав этих подсистем, соответствующий современному уровню развития технологии производства ИЭТ, показан на рисунке 5.3. При необходимости он может быть расширен или сокращен. АСТПП является промежуточным звеном между конструкторской подготовкой производства (САПР) и производственными цехами, службами с их системами управления и связана с ними большим числом различных связей.

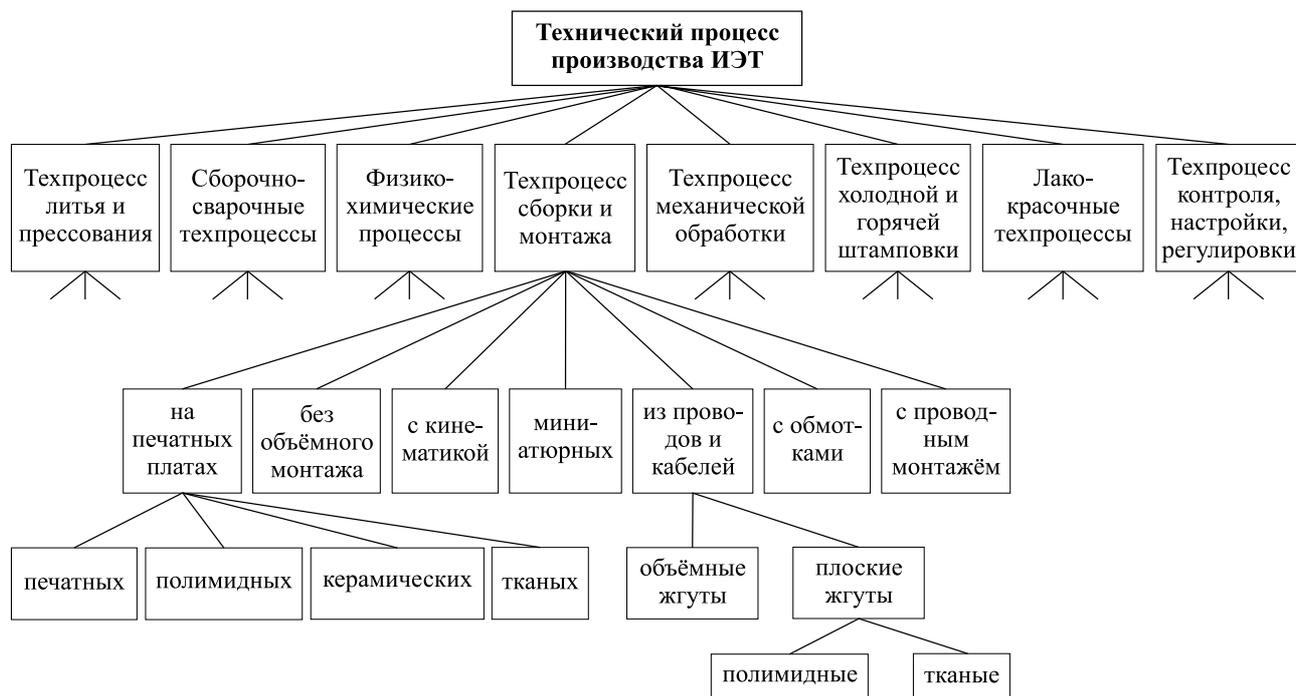


Рис. 5.3 – Структура технологического производства изделий электронной техники (ИЭТ)

В настоящее время проектирование, обработка данных и управление при ТПП осуществляются преимущественно автоматизированными методами, на основе унифицированных носителей информации и типовых проектных решений

разнообразных задач подготовки производства. Объекты автоматизации в технологической подготовке производства:

- проектирование технологических процессов и технологической оснастки;
- решение инженерно-технических задач;
- решение задач управления;
- информационный поиск деталей-прототипов, сведений о процессах их обработки и технологической оснастке для заимствования;
- разработка программ для оборудования и станков с числовым программным управлением.

Процесс технологического проектирования в АСТПП можно представить с помощью следующих этапов:

1. Информация с чертежа детали, узла, микросборки и т. д., для которой необходимо разработать технологическую документацию, переводится специально разработанными методами кодирования на язык описания геометрических образов, удобный для ввода в ЭВМ. Представленные в таком виде сведения после проверки, корректировки и занесения на машинные носители информации вводятся в ЭВМ.
2. В памяти ЭВМ записывается информация о производственной обстановке на предприятии: о составе оборудования и его характеристиках, о наличии инструмента и оснастки; нормативные данные массива типовых технологических процессов; программы проектирования технологических процессов; технологические методы согласно их классификации. На основании введенных данных ЭВМ относит информацию к той или иной группе, выбирает для группы типовой технологический процесс, доводит его до рабочего состояния и распечатывает маршрутную и операционную технологические карты.
3. На основании спроектированного технологического процесса ЭВМ проверяет наличие на предприятии выбранного инструмента и оснастки и достаточное их количество. Если на предприятии не оказалось какого-либо вида оснастки, то ЭВМ по типовым программам проектирует недостающий инструмент. Если типовая программа отсутствует, то ЭВМ формирует задания на интерактивное проектирование.
4. Машина проектирует технологический процесс изготовления новой оснастки (см. этап 2).

5. Если на предприятии есть оборудование с ЧПУ, то ЭВМ с помощью типовых программ формирует управляющую программу и заносит на цифровой носитель [4, 5].

Внедрение АСТПП обеспечивает сокращение сроков подготовки производства в 3–5 раз, повышение качественных показателей разрабатываемых технологических процессов и технологической оснастки, улучшение условий труда инженерно-технических работников, освобождение их от однообразной работы. Экономический эффект применения АСТПП достигается за счет снижения трудоемкости процесса проектирования и использования резервов в технологических процессах, таких как повышение качества изделий электронной техники, уменьшение расхода материалов, уменьшение отходов, оптимизация компоновки конструкций, оптимизация режимов травления, диффузии и т. д.

Основные требования, предъявляемые к автоматизированной системе технологической подготовки производства:

- обеспечение автоматизации процессов основных видов деятельности инженерно-технических работников при технической подготовке производства, гибкости при переходе на выпуск новых изделий;
- наиболее рациональное распределение функций между техническим работником и ЭВМ;
- использование современных методов проектирования технологических процессов;
- внедрение на предприятиях с различным характером и масштабом производства;
- унификация методов, языков взаимодействия, математического и программного обеспечения, технических средств и т. д.;
- объединение с САПР для построения интегрированных компьютеризованных производств;
- непрерывная модификация с учетом развития программных и вычислительных средств [4, 5].

5.4.2 Пакеты программ автоматизации проектирования РЭС

Системы программ, предназначенные для автоматизации проектирования РЭС, можно разделить на две основные группы: системы схемотехнического проектирования и конструкторского проектирования РЭС. Это деление весьма

условно, особенно для систем проектирования микроэлектронной аппаратуры, а также СВЧ и излучающих устройств. В последнее десятилетие наметилась явная тенденция интеграции пакетов программ для схемотехнического и конструкторского проектирования радиоустройств.

Кроме этих основных групп пакетов программ, в системах автоматизированного проектирования РЭС широко используются различные вспомогательные пакеты: математические пакеты, базы данных, графические и текстовые редакторы, электронные таблицы и т. п.

Пакеты программ для схемотехнического проектирования РЭС

В настоящее время известно большое число пакетов программ, предназначенных для выполнения схемотехнического проектирования радиоэлектронных средств и ориентированных на использование рабочих станций и персональных компьютеров. Рассмотрим некоторые из этих пакетов, получившие наибольшее распространение.

Системы схемотехнического моделирования аналоговых и цифровых устройств и проектирования печатных плат Design Center и Design Lab разработаны корпорацией MicroSim. В основу этих систем положен известный пакет программ PSPICE, первые версии которого были разработаны в начале 1970-х гг. в Калифорнийском университете. Возможности этих систем существенно зависят от варианта поставки и операционной системы.

Чаще всего системы позволяют выполнять моделирование аналоговых, смешанных аналого-цифровых и просто цифровых радиоустройств, синтез цифровых устройств на базе интегральных схем с программируемой логикой, выполнять моделирование с учетом паразитных емкостей и индуктивностей, присущих реальным печатным платам.

Системы Design Center и Design Lab оснащены графическим редактором принципиальных схем РЭС, который одновременно является управляющей оболочкой для запуска программных модулей на всех стадиях работы с системами. Системы позволяют выполнять расчет режимов радиоустройств по постоянному и переменному току, спектральный анализ, моделирование переходных процессов в РЭС, расчет уровней шума, выполнять статистический анализ, учитывать вариации температуры при работе устройства. Моделирование аналого-цифровых и цифровых устройств выполняется на логическом уровне с подключением аналого-цифровых и цифро-аналоговых интерфейсов для обеспечения связи аналоговой и цифровой частей РЭС.

В системах может выполняться параметрическая оптимизация аналого-цифровых устройств по заданным критериям при наличии нелинейных ограничений на параметры варьируемых компонентов.

Системы имеют графический редактор печатных плат, который в автономном режиме воспринимает информацию о соединениях в формате P-CAD, а в составе системы – от графического редактора принципиальных схем. Компоненты РЭС автоматически или вручную размещаются на одной или двух сторонах печатной платы, после чего возможна трассировка многослойных соединений в автоматическом или интерактивном режиме. По результатам трассировки создаются командные файлы для изготовления фотошаблонов для сверлильных станков с ЧПУ. Возможна передача данных в систему AutoCAD для выпуска конструкторской документации.

К системам Design Center и Design Lab прилагаются многочисленные библиотеки графических символов элементов РЭС и банки данных с более чем восемью тысячами математических моделей компонентов (диодов, тиристоров, биполярных и полевых транзисторов, оптопар, операционных усилителей, компараторов напряжения, кварцевых резонаторов, магнитных сердечников, цифровых и аналого-цифровых микросхем) производства различных фирм США, Западной Европы и Японии. Имеется возможность пополнения библиотек.

Примерно такими же возможностями с точки зрения схемотехнического проектирования обладает система ЮАР, разработанная фирмой Intusoft. В основу этой системы также положен пакет программ PSPICE. Система ICAP отличается удобным оконным редактором, возможностью работы с измерительным оборудованием, выходные файлы системы совместимы с пакетом P-CAD.

Получившая широкое распространение в конце 1970-х – начале 1980-х гг. первая версия пакета программ Super-Compact сделала фирму Compact Software одной из ведущих в мире по разработке программного обеспечения для проектирования радиочастотных устройств, особенно СВЧ-диапазона. Пакет Super-Compact оказался очень удачным и с тех пор интенсивно развивается. Достаточно точные математические модели широкого круга элементов СВЧ-цепей, удобство входного языка и ряд других достоинств привели к широкому распространению этого пакета, ориентированного на проектирование фильтров, согласующих цепей и цепей связи СВЧ-диапазона.

В пакете используются базовые элементы в виде двух-, четырех-, шести- и восьмиполюсников, характеризующихся A -, S -, Y -параметрами. Допускается включение произвольных многополюсников, описанных одними из перечис-

ленных параметров таблично, в зависимости от частоты. Пакет позволяет анализировать СВЧ-цепи, содержащие наиболее употребительные конфигурации элементов с распределенными и сосредоточенными параметрами. Распределенные элементы могут представлять различные типы связанных и одиночных отрезков линий передачи: микрополосковые и полосковые, линии с подвешенной подложкой и коаксиальные, компланарные и щелевые, прямоугольные и круглые волноводы. Распределенные элементы могут характеризоваться как геометрическими, так и электрическими параметрами. При моделировании СВЧ-устройств учитываются электрические неоднородности, влияние дисперсии, излучения, многослойной металлизации проводников, шероховатости поверхности, потерь в проводниках и диэлектрических потерь.

Модели полупроводниковых приборов описываются параметрами рассеяния либо матрицами классической теории цепей. Моделирование СВЧ-цепей выполняется на основе аппарата Y -матрицы, причем отдельные части цепи могут описываться с помощью аппарата A -матриц.

Описание устройства производится покомпонентно, причем в одной строке указываются аббревиатурой тип компонента, способ его подключения в цепи и параметры компонента с допустимыми границами изменения параметров в процессе оптимизации.

Оптимизация параметров устройства может выполняться одновременно для нескольких характеристик и в нескольких состояниях. Например, СВЧ-диодный переключатель может одновременно оптимизироваться в режимах «включено» и «выключено». Для оптимизации используется одна из разновидностей градиентного метода и случайный поиск с самообучением. Процесс оптимизации параметров устройства ведется в границах изменения варьируемых параметров, определяемых их физической реализуемостью и точностью моделей компонентов.

Для обеспечения разработчика информацией о параметрах полупроводниковых компонентов, проводников и диэлектрических материалов в пакет включено несколько банков данных, которые могут вызываться на командном уровне для отыскания компонента с заданными техническими характеристиками. Пакет позволяет разработчику формировать свои банки данных.

Вывод информации осуществляется на экран монитора, печать либо графопостроитель и может включать схему устройства и ее различные частотные характеристики в табличном виде либо в виде графиков в полярных или декартовых координатах. По требованию разработчика Super-Compact может по-

строить линии равного усиления, уровня шума, стабильности устройства. Для точного изучения отдельные участки характеристик могут быть представлены в увеличенном масштабе.

Дальнейшим развитием пакета Super-Compact является вышедшая в середине 1980-х гг. и полностью с ним совместимая система Microwave Harmonica [14], позволяющая выполнять моделирование и ряда нелинейных СВЧ-устройств, таких как маломощные усилители и усилители большого уровня мощности, смесители, генераторы, переключатели и другие устройства радиочастотных трактов. Для моделирования нелинейных устройств в этой системе используется метод гармонического баланса. В начале 1990-х гг. появилась версия системы Microwave Harmonica, ориентированная на платформу Windows. Различные версии системы Microwave Harmonica широко используются в России разработчиками радиоаппаратуры различного назначения.

В последние годы эту линию программ развивает фирма Ansoft, которой были разработаны новые программные системы:

- Serenade, позволяющая выполнять моделирование и оптимизацию не только СВЧ, но и оптоэлектронных устройств и имеющая много версий, в числе которых есть версии, ориентированные на топологическое представление моделируемого устройства. Имеются версии этой системы, ориентированные на платформы Windows 95 и Windows NT;
- Super-Spice, в которой путем подключения известной программы Spice решена задача моделирования СВЧ-устройств во временной области;
- Microwave Success, позволяющая выполнять моделирование систем радиотелефонии;
- Microwave Explorer – система для моделирования электромагнитных полей в различных элементах радиоэлектронных устройств и ряд других.

Кроме разработки программного обеспечения фирмой Ansoft поддерживаются банки данных с линейными и нелинейными моделями элементов СВЧ и оптоэлектронной техники.

Пакет программ Micro-CAP версии 5.0 и выше, разработанный фирмой Spectrum Software, предназначен для анализа и моделирования аналоговых и смешанных аналогово-цифровых устройств. Пакет находит широкое применение в учебном процессе и инженерной практике при разработке разнообразных радиоэлектронных устройств (усилителей, фильтров, модуляторов и демодуля-

торов, автогенераторов, логических элементов, стабилизаторов напряжения, устройств сопряжения аналоговых и цифровых устройств и т. п.).

Пакет проводит анализ характеристик электронных устройств, принципиальные схемы которых вводятся с помощью графического редактора. Программа позволяет проводить четыре вида анализа: расчет переходных процессов и режима по постоянному току нелинейных электрических цепей, расчет частотных характеристик линеаризованных цепей, а также спектральный анализ переходных процессов.

Библиотека компонентов содержит модели операционных усилителей, полупроводниковых диодов, биполярных и полевых транзисторов, трансформаторов, источников синусоидальных и импульсных сигналов, нелинейных зависимых источников тока и напряжения и различных пассивных компонентов. Разработчик имеет возможность пополнять библиотеки компонентов.

Вывод информации о результатах моделирования может осуществляться на экран монитора, печать либо графопостроитель и включает вывод схемы устройства, параметров моделирования и его результатов в табличной или графической форме.

Программы конструкторского проектирования РЭС

В большинстве из рассмотренных выше пакетов схемотехнического проектирования РЭС предусмотрены операции конструкторского проектирования.

Однако существуют чисто конструкторские пакеты, обеспечивающие более полное решение различных задач конструкторского проектирования РЭС.

Пакет программ P-CAD фирмы Personal CAD Systems, Inc. – это полное комплексное программное решение для проектирования электронных устройств, в частности ввода схемы и проектирование схемной печатной платы. Комплексное решение предполагает, что логика, описанная в схеме, воплощается в топологию печатной платы; программы осуществляют функции логического моделирования, проверяют соблюдение правил проектирования, создают список соединений для моделирования, автоматически размещают компоненты и трассируют печатную плату и создают документы для автоматизированных производственных систем. Пакет содержит взаимодействующие средства проектирования, удобную для пользователя оболочку и интеллектуальную базу данных, обширную библиотеку, диалоговые редакторы, средства сопряжения с популярными средствами анализа. Пакет имеет открытую архитектуру, обеспе-

чивает выдачу готовых документов для технологии монтажа и другую проектную документацию.

Вывод документации после контроля на дисплее может осуществляться на принтер, плоттер или фотоплоттер. Оболочка системы помогает пользователю двигаться сквозь процесс проектирования с помощью меню, подсказок и правок. Система проектирования печатной платы обеспечивает средства для полной разработки топологии: от диалогового редактора до автоматического размещения компонентов, автотрассировки, проверки соблюдения правил проектирования и сопряжения с производством.

Библиотека пакета содержит обширную информацию о компонентах электронных схем от дискретных и электромеханических деталей до существующих и заказных микросборок интегральных схем. Программные средства сопряжения превращают данные из списка соединения компонентов схемы в формат, необходимый для конкретной программы моделирования цифровой и аналоговой схем (типа PSPICE и др.). Пакет позволяет проектировать печатные платы, имеющие до 500 элементов и 2 000 связей.

Пакет программ OrCAD фирмы OrCAD System Corp. является законченным и гибким программным блоком схемотехнического и конструкторского проектирования. Он обеспечивает ввод и вывод на печать принципиальных схем, трассировку печатной платы и другие операции. Пакет управляется с помощью иерархической разветвленной системы меню, легок в обучении пользованию, обладает многими дополнительными возможностями ввода и вывода схем.

Библиотека пакета содержит более 2 700 изображений компонентов РЭС; можно легко создавать собственные начертания элементов. Простым нажатием клавиши легко выполняются многие графические операции при вводе и выводе схем: увеличение и уменьшение масштаба, преобразование (вращение, перенос, отображение и пр.) элементов и любых заданных фрагментов схемы. В системе предусмотрены создание перечня элементов (спецификаций), возможность разведения проводников, шин, входов модулей.

Пакет OrCAD в настоящее время является самым удобным и богатым по своим возможностям для ввода и вывода графических изображений принципиальных схем РЭС.

Пакет имеет удобный выход на подсистемы моделирования и анализа РЭС, а также другие графические пакеты (PSPICE, P-CAD и др.).

Пакет универсального назначения AutoCAD фирмы AutoDesk [15] разработан на самом современном уровне машинной графики и предоставляет разработчику исключительно широкие возможности проектирования разнообразных объектов, технических систем устройств: дома и печатные платы, станки, детали и одежда. Пакет представляет собой систему автоматизированной разработки чертежей, причем чертежи, рисунки и схемы создаются в интерактивном режиме, управляемом системой иерархических меню. В любой чертеж может быть вставлен поясняющий текст. В набор функций входят панорамирование, увеличение, масштабирование, поворот, секционирование, штриховка и другие операции преобразования изображений. В системе предусмотрены подсказки в любом состоянии и для любой команды.

В пакете разработан богатый выбор драйверов графических устройств – графических дисплеев, матричных принтеров, графических планшетов и плоттеров. Одним из важнейших достоинств пакета является возможность работы с трехмерной графикой, позволяющей строить реальные объекты, которые можно наблюдать в различных ракурсах (при желании невидимые линии на изображении стираются). Применен специальный метод полилиний для вывода сложных кривых контуров деталей.

Система AutoCAD непрерывно совершенствуется. Так, в последние версии системы включен интерпретатор языка AutoLisp – одной из версий языка LISP, широко применяемого в символьной обработке и в системах искусственного интеллекта. Использование этого языка позволяет пользователю, с одной стороны, определять собственные функции и команды в среде AutoCAD, с другой – обеспечивать связь AutoCAD с другими приложениями. Сейчас начинают появляться еще более сложные системы, включающие не только язык программирования, но и экспертные системы (экспертные настройки) для принятия решений и подсказок конструктору в процессе разработки. В эти настройки включен набор правил и математических моделей, и конструктор в процессе работы может получить «советы» по оптимальному выбору тех или иных параметров разрабатываемой системы.

Программные системы для электродинамического моделирования

В конце 1990-х гг. фирмой AWR была выпущена на рынок программная система Microwave Office. Эта система включает два основных программных модуля: EMSight и Voltaire XL, которые позволяют решать широкий круг задач, возникающих при проектировании высокочастотных интегральных и монолит-

ных СВЧ-микросхем, антенн, СВЧ согласующих цепей и фильтров, усилителей, смесителей и автогенераторов.

В пакете EMSight для анализа полей используется метод моментов Галеркина, представляющий точный и устойчивый алгоритм электродинамического моделирования. Здесь также используется алгоритм быстрого частотного свипирования, позволяющий на порядок сократить время моделирования по сравнению с обычным поточечным методом частотного анализа.

Пакет EMSight позволяет анализировать поля в слоистых средах с проводниками произвольной формы, рассчитывать поля антенны в дальней зоне и т. п. Пакет имеет расширенные графические возможности, которые позволяют разработчикам наблюдать реалистичное анимационное изображение токов высокой частоты, показывающее как амплитуду тока, так и его направление в трехмерном пространстве. Пакет также имеет обширные возможности для обработки и представления полученных при моделировании результатов.

В пакете программ *Voltaire XL* используются самые последние разработки методов схемотехнического моделирования. Модуль пакета, предназначенный для линейного моделирования в частотной области, содержит более 500 моделей сосредоточенных и распределенных элементов СВЧ-цепей, включая микрополосковые, копланарные и щелевые элементы, полые и диэлектрические волноводы.

Для анализа нелинейных устройств используются одно- и многочастотные методы гармонического баланса; для анализа интермодуляционных искажений – ряды Вольтерра; конверсионно-матричный метод для анализа смесителей; в пакете используется высокоскоростной метод шумового анализа, а также интегрированная система описания схем со встроенной поддержкой для файлов систем Spice и MMICAD. Пакет позволяет выполнять нелинейный анализ усилителей, смесителей и автогенераторов, нелинейный шумовой анализ, включая анализ фазовых шумов, шумов смесителей, а также нелинейный анализ их устойчивости.

EMSight и Voltaire XL написаны на объектно-ориентированном языке C, имеют общий дружественный пользователю интерфейс, могут интегрировать в себя новые методы моделирования по мере их появления.

Microwave Office работает под управлением 32-разрядных операционных систем, таких как Windows 95, 98 и Windows NT, и оптимизирован для работы под ними.

Фирмой EAGLEWARE рекламируется универсальная программная система Genesys, предназначенная для проектирования LC-фильтров, усилителей, смесителей и автогенераторов, распределенных СВЧ согласующих цепей и фильтров, активных фильтров, выравнивателей группового времени задержки и других элементов РЭС. Система Genesys включает множество программных модулей, основными из которых являются SuperStar Pro, Schemax, Layout, Filter, Oscillator, Match, T/Line и др.

Пакет SuperStar Pro – высокоскоростная программа для моделирования радиочастотных цепей, содержащая множество моделей пассивных компонентов цепей как СВЧ, так и более низкочастотных диапазонов. Активные компоненты радиочастотных цепей моделируются с использованием S -параметров. При моделировании допускается произвольное число портов в цепи, в процессе работы программы могут вычисляться различные импедансные, адмиттансные характеристики и характеристики рассеяния, а также распределение полей по проводникам моделируемого устройства. Пакет Schemax позволяет на интуитивном уровне создавать графические образы моделируемых элементов РЭС для передачи описания в SuperStar Pro. Программа импортирует файлы описания цепей из систем Design Center и Design Lab и других стандартных форматов описания цепей СВЧ.

Пакет Layout позволяет по описанию моделируемого устройства в SuperStar Pro и Schemax синтезировать его топологию и представить трехмерную анимационную картину распределения токов по проводникам. Layout позволяет использовать в устройстве до 128 слоев, включая металлизацию, диэлектрические подложки, монтажные слои и т. п. Программа содержит большую библиотеку геометрических образов пассивных и активных компонентов СВЧ-цепей. Выходной файл программы Layout соответствует DXF файлу системы AutoCAD.

Различные модификации пакетов Filter и программы Oscillator позволяют синтезировать различные типы СВЧ, LC- и активных фильтров, автогенераторов и усилителей с выполнением анализа шумов.

Пакет Match предназначен для синтеза сосредоточенных и распределенных согласующих цепей, включенных между двумя произвольными комплексными нагрузками. Пакет позволяет решать задачи максимизации полосы пропускания при заданном уровне отражения сигнала либо минимизации отражения в заданной полосе частот.

Пакет T/Line позволяет выполнять анализ и синтез одиночных и связанных передающих линий различной конфигурации по их геометрическим или электрическим характеристикам.

В системе Genesys используется весьма совершенный графический интерфейс, поддерживается библиотека для нескольких тысяч выпускаемых промышленностью активных и пассивных компонентов. Система Genesys имеет пользовательский интерфейс, полностью совпадающий со стандартным интерфейсом программного обеспечения фирмы Microsoft, снабжена многочисленными файлами помощи, что делает ее удобной для использования и сравнительно лёгкой для освоения.



Контрольные вопросы по главе 5

1. Назовите основные принципы построения современных САПР.
2. Что является объектом автоматизации в технологической подготовке производства?
3. Раскройте понятие «жизненный цикл» технического изделия.
4. Перечислите этапы жизненного цикла промышленных изделий.
5. Назовите основные типы АС.
6. Дайте определение САПР.
7. Что является основой САПР ТП?
8. Что характеризует поколения РЭС?
9. Охарактеризуйте виды обеспечения САПР.

Заключение

Автоматизация проектирования – важная составная часть научно-технического прогресса техники и, естественно, развивается вместе с соответствующей областью техники. В связи с дальнейшим совершенствованием радиоэлектронных средств происходит непрерывное развитие теории и практики как автоматизации научных исследований, так и автоматизации проектирования радиоэлектронных средств.

Системы автоматизации научных исследований и автоматизированного проектирования РЭС развиваются на базе новейших достижений специальных дисциплин, математики, вычислительной техники, системного программирования. При этом внутренние потребности развития САПР оказывают значительное влияние на развитие этих дисциплин, являются стимулом для их прогресса. Рассмотрим некоторые тенденции развития автоматизации проектирования радиоэлектронных средств.

Автоматизация научных исследований охватывает область от изучения слабо исследованных процессов до испытания новых образцов радиоэлектронных средств, разработанных на основе этих исследований. Особенности автоматизации научных исследований заключаются в том, что из-за чрезвычайно широкого многообразия решаемых в процессе исследований задач, как правило, не бывает рекомендаций по выбору методов и готовых алгоритмов и программ. В этом заключается основное отличие от проектных процедур, при которых обычно известны методики, используемые математические методы, алгоритмы, а часто в распоряжении разработчика бывают готовые пакеты либо комплексы программ.

Основные теоретические проблемы, стоящие перед разработчиками систем автоматизации научных исследований в радиоэлектронике, по нашему мнению, заключаются в разработке макромоделей различных элементов радиоэлектронных средств и методик их получения для любых уровней исследований и проектирования в любой форме представления, наиболее удобной для решения конкретной проектной задачи; разработке новых математических методов и алгоритмов; разработке путей развития дружественного пользователю программного обеспечения для решения проектных задач.

Несколько сложнее кратко сформулировать пути развития систем автоматизированного проектирования РЭС, так как проектные задачи существенно

различаются в зависимости от класса подлежащих проектированию объектов. Тем не менее можно предположить, что в ближайшем будущем практически все проектирование радиоустройств будет проводиться на основе комплексных САПР, причем основное внимание будет уделяться созданию САПР, способных настраиваться на решение широкого класса проектных задач, анализировать технические решения, производить их альтернативный выбор из нескольких вариантов, выполнять оптимизацию радиоэлектронных средств сразу по многим критериям, производить проектирование с учетом требований схемотехнического, конструкторского и технологического проектирования.

В новых САПР РЭС существенное внимание будет уделяться сокращению трудоемкости подготовки к решению проектных задач и формированию массивов исходных данных.

Перечисленные выше проблемы, стоящие перед системами автоматизированных исследований и проектирования радиоэлектронных средств, свидетельствуют о том, что разработчик и пользователь этих систем должны обладать фундаментальными знаниями в области специальных дисциплин, математики, теории алгоритмов и программирования, а возможности совершенствования таких систем являются практически неисчерпаемыми.

Литература

Основная литература

1. Автоматизация проектирования радиоэлектронных средств : учеб. пособие для вузов / под ред. О. В. Алексеева. – М. : Высш. шк., 2000. – 479 с.
2. Норенков И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры / И. П. Норенков, В. Б. Маничев. – М. : Высш. шк., 1983.
3. Системы автоматизированного проектирования / под ред. И. П. Норенкова. – М. : Высш. шк., 1986.
4. Корячко В. П. Теоретические основы САПР / допущено Министерством высшего и среднего специального образования СССР в качестве учебника для студентов вузов, обучающихся по специальности «Конструирование и производство электронно-вычислительной аппаратуры» ; В. П. Корячко, В. М. Курейчик, И. П. Норенков. – М. : Энергоиздат, 1987. – 383 с.
5. Курейчик В. М. Математическое обеспечение конструкторского и технологического проектирования с применением САПР : учебник для высших учебных заведений / допущено Министерством высшего и среднего специального образования СССР в качестве учебника для студентов вузов, обучающихся по специальности «Конструирование и технология радиоэлектронных средств». – М. : Радио и связь, 1990. – 352 с.

Дополнительная литература

6. Автоматизация схемотехнического проектирования / под ред. В. Н. Ильина. – М. : РиС, 1987.
7. Влах И. Машинные методы анализа и проектирования электрических схем / И. Влах, К. Синхгал. – М. : РиС, 1988.
8. Турчак Л. И. Основы численных методов / Л. И. Турчак. – М. : Наука, 1987.
9. Фидлер Дж. К. Машинное проектирование электрических схем / Дж. К. Фидлер, К. Найнтингейл. – М. : Высш. шк., 1985.

10. Разевиг В. Д. Применение программ PCAD и Pspice для схемотехнического моделирования на ПЭВМ / В. Д. Разевиг. – М. : РиС, 1992. – Вып. 1–4.
11. Баканов Г. Ф. Основы конструирования и технологии радиоэлектронных средств : учеб. пособие для вузов / Г. Ф. Баканов, С. С. Соколов, В. Ю. Суходольский. – М. : Академия, 2007. – 368 с. : ил.
12. Трофимов А. В. Автоматизация проектирования АСУ ТП тепловых электростанций / А. В. Трофимов // Теплоэнергетика. – 2009. – № 10. – С. 32–36.
13. Долгих Э. А. Основы применения CALS-технологий в электронном приборостроении. Версия 1.0 [Электронный ресурс] : электрон. учеб. пособие / Э. А. Долгих, А. В. Сарафанов, С. И. Трегубов. – Красноярск : ИПК СФУ, 2008.
14. Алексеева О. В. Проектирование и расчёт устройств СВЧ в системе Microwave Harmonica / О. В. Алексеева, А. А. Головков, В. Ю. Приходько. – СПбГЭТУ, 1997.
15. AutoCAD 14. Русская и англоязычная версии / Э. Т. Романычева, Т. М. Сидорова, С. Ю. Сидоров, Т. Ю. Трошина. – М. : ДМК, 1998.

Список условных обозначений и сокращений

БНК-1 – базовая несущая конструкция первого уровня.

БНК-2 – базовая несущая конструкция второго уровня.

БНК-3 – базовая несущая конструкция третьего уровня.

ЕСКД – Единая система конструкторской документации.

КД – конструкторская документация.

КП – коммутационное поле.

КС – коммутационная схема.

НИОКР – научно-исследовательская, опытно-конструкторская работа и/или разработка аванпроекта.

НИР – научно-исследовательская работа.

НК – несущая конструкция.

НК-0 – базовая несущая конструкция нулевого уровня.

ОКР – опытно-конструкторская работа.

РЭА – радиоэлектронная аппаратура. В настоящее время вместо этой аббревиатуры чаще используется аббревиатура *РЭС* – радиоэлектронное средство.

РЭМ-0 – модуль нулевого уровня.

РЭС – радиоэлектронное средство.

СКТ – суперкомпьютерный *инжиниринг*, современные технологии проектирования и разработки высокотехнологичной продукции.

СММ – справочно-методические материалы по общему конструированию РЭС.

СРПП – Система разработки и постановки продукции на производство.

ССБТ – Система стандартов безопасности труда.

Т-система – техническая система.

ТД – технологическая документация.

ТЗ – техническое задание.

ТЗОКР – техническое задание на опытно-конструкторскую работу.

ТПП – технологическая подготовка производства.

УФУ – унифицированный функциональный узел на основе дискретных ИМС.

ЦВМ – цифровая вычислительная машина.

ЭВМ – электронная вычислительная машина.

Экстенсивный путь развития производства – наращивание производственных мощностей (для его ранних стадий).

ЭРИ – изделие электронной техники, квантовой электроники или электротехническое (электрорадиоизделие). Для выражения этого и близких понятий в технической литературе и стандартах используются также аббревиатуры ИЭТ (изделие электронной техники), ЭРЭ (электрорадиоэлемент), ЭТИ (электротехническое изделие).

ЭРК – электрорадиокомпоненты (резисторы, конденсаторы, транзисторы и др.).

Глоссарий

Конструктивная сложность РЭС – в классификации по конструктивной сложности РЭС делят на средства в модульном и немодульном исполнении. Уровни разукрупнения РЭС в немодульном исполнении по конструктивной сложности включают: шкаф, блок и ячейку.

Нулевой уровень – функционально и конструктивно законченное РЭС (ИМС, микросборка, микросхема).

Радиоэлектронная система – РЭС, представляющее собой совокупность функционально взаимодействующих автономных радиоэлектронных комплексов и устройств, образующих единое целое.

Радиоэлектронное устройство – радиоэлектронное средство, представляющее собой совокупность функционально и конструктивно законченных сборочных единиц и используемое для решения технической задачи в соответствии с его назначением.

Радиоэлектронное устройство – РЭС, представляющее собой функционально законченную сборочную единицу, которая реализует функции передачи, приема, преобразования информации или техническую задачу на их основе и выполнена на несущей конструкции.

Радиоэлектронный комплекс – РЭС, представляющее собой совокупность функционально связанных радиоэлектронных устройств.

Радиоэлектронный узел – РЭС, представляющее собой функционально законченную сборочную единицу, которая реализует функцию преобразования сигнала и не имеет самостоятельного эксплуатационного значения.

Суперкомпьютерные технологии – технологии расчетного обоснования сложных изделий промышленности на качественно новом уровне.

Унификация – метод стандартизации.

Уровни разукрупнения радиоэлектронных средств в модульном исполнении по конструктивной сложности включают: электронный модуль; унифицированный электронный модуль; стандартный электронный модуль; специализированный стандартный электронный модуль и модули 3, 2, 1-го и нулевого уровня.

Фазовая плоскость – координатная плоскость, в которой по осям координат откладываются какие-либо две переменные (фазовые координаты), однозначно определяющие состояние системы второго порядка. Фазовая плоскость

является частным случаем фазового пространства, которое может иметь бóльшую размерность. Каждая точка фазовой плоскости отражает одно состояние системы и называется фазовой.

Фазовые переменные характеризуют физическое или информационное состояние объекта, а их изменения во времени выражают переходные процессы в объекте.

Функциональная сложность РЭС – по степени функциональной сложности выделяют следующие уровни разукрупнения РЭС: радиоэлектронные системы, комплексы, устройства и узлы.

Ячейка (РЭС) – радиоэлектронное устройство или радиоэлектронный функциональный узел, выполненное (выполненный) на основе несущей конструкции первого уровня.