

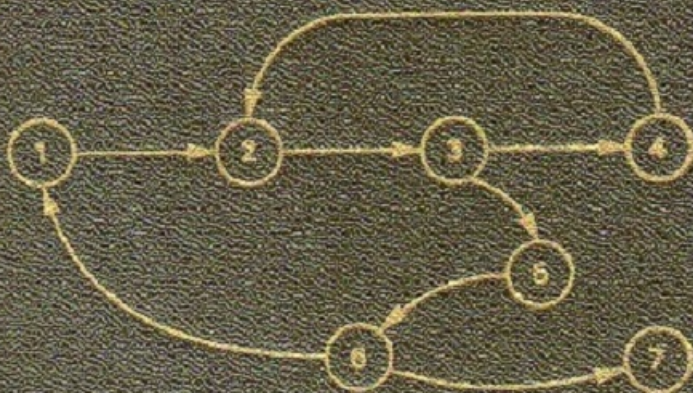
В. П. Корячко  
В. М. Курейчик  
И. П. Норенков

# ТЕОРЕТИЧЕСКИЕ ОСНОВЫ САПР

---

Для студентов вузов

---



В. П. Корячко,  
В. М. Курейчик,  
И. П. Норенков

# 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»



МОСКВА  
ЭНЕРГОАТОМИЗДАТ 1987

ББК 32.97

К 70

УДК 681.31.00

Рецензенты:  
кафедра «Конструирование электронно-вычислительной аппаратуры» Ленинградского института точной механики и оптики;

профессор, доктор техн. наук Е. Л. Глориозов (Московский институт электронного машиностроения)

**Корячко В. П. и др.**

**К70** Теоретические основы САПР: Учебник для вузов/В. П. Корячко, В. М. Курейчик, И. П. Норенков. — М.: Энергоатомиздат, 1987. — 400 с.: ил.

Изложены теоретические основы САПР, их технические и программные средства. Значительное внимание уделено основным сведениям об информационных потоках, структурах и технических средствах САПР, об устройствах машинной графики, программном обеспечении технических средств, системах управления базами данных, способах защиты данных.

Для студентов, обучающихся по специальности «Конструирование и производство электронно-вычислительной аппаратуры».

К  $\frac{2405000000-319}{051(01)-87}$  285—87

ББК 32.97

© Энергоатомиздат, 1987

## ПРЕДИСЛОВИЕ

Стратегическая линия ~~КССС~~ на ускорение социально-экономического развития — это выход в короткие сроки на самые передовые научно-технические позиции, на высший мировой уровень производительности общественного труда. Для реализации этой стратегии необходимо создание и быстрое качественное внедрение высокоэффективных машин, оборудования, приборов и технологических процессов, обеспечивающих высокие темпы научно-технического прогресса.

В решениях ~~ХНХН~~ съезда ~~КССС~~ по «Основным направлениям экономического и социального развития ~~СССР~~ на 1986—1990 годы и на период до 2000 года» сказано: «Внедрять автоматизированные системы в различные сферы производства, и в первую очередь в проектирование, управление оборудованием и технологическими процессами».

Современные задачи, возникающие перед наукой и техникой, вызывают необходимость проектирования все более сложных технических объектов в сжатые сроки. Удовлетворить противоречивые требования повышения сложности объектов, сокращения сроков и повышения качества проектирования с помощью простого увеличения численности проектировщиков нельзя, так как возможность параллельного проведения проектных работ ограничена и численность инженерно-технических работников в проектных организациях страны не может быть сколько-нибудь заметно увеличена. Выходом из этого положения является широкое применение вычислительной техники для решения проектных задач (автоматизация проектирования).

Цель автоматизации проектирования — обеспечить бездефектное проектирование, снизить материальные затраты, сократить сроки проектирования и ликвидировать рост количества инженерно-технических работников, занятых проектированием.

Знание математического аппарата, применяемого в инженерных исследованиях, умение пользоваться математическими моделями при оптимальном проектировании реальных объектов и систем, знание программных и технических средств САПР и умение пользоваться ими в качестве ин-

струмента проектировщика должны позволить современным инженерам ставить и решать задачи автоматизации проектирования по отраслям техники.

В связи с тем что существенно изменяются функции инженера в процессе автоматизированного проектирования и конструирования, возникла настоятельная необходимость подготовки специалистов в области создания и использования САПР. В технических вузах страны открыта специальность «Системы автоматизированного проектирования».

Кроме того, подготовка специалистов в области САПР проводится в рамках ряда специальностей, в том числе и специальности «Конструирование и производство электронно-вычислительной аппаратуры» (специализация 0648.02 «Конструирование и производство технических средств САПР»).

Предметом курса являются теоретические основы САПР, их технические и информационные средства.

Целью курса «Теоретические основы САПР» является систематическое изложение принципов организации, создание и функционирование САПР.

Содержание учебника полностью соответствует программе одноименного курса для студентов, обучающихся по специальности 0648.

Курс базируется на знаниях, полученных студентами при изучении марксистско-ленинской философии, общеобразовательных дисциплин (высшей математики, физики, инженерной графики, теоретической механики), курсов «Введение в специальность», «Лингвистическое и программное обеспечение САПР», «Теоретические основы конструирования, технологии и надежности ЭВА», «Математическое обеспечение конструкторского и технологического проектирования с применением САПР».

В гл. 1 рассматриваются общие сведения о проектировании технических средств объекта, классификация и основные структуры САПР, виды обеспечения САПР, анализируется и обосновывается выбор критериев оптимальности.

В гл. 2 разбираются вопросы организации технических средств САПР, технические средства машинной графики и систем передачи данных. Анализируется организация программных и информационных средств САПР.

В гл. 3 рассматриваются информационные потоки в САПР, классификация баз данных, организация системы управления базами данных, языки баз данных, описыва-

ются логическая и физическая организация баз данных, организация поиска данных.

В гл. 4 анализируются особенности математического аппарата для математического моделирования на различных иерархических уровнях проектирования БИС и ЭВА.

В гл. 5 рассматриваются методы анализа процессов функционирования элементов интегральных схем, методы анализа статических режимов и переходных процессов в объектах на различных уровнях, методы анализа тепловых режимов, методы анализа логических и функциональных схем ЭВА, методы многовариантного и статистического анализа.

Глава 6 посвящена синтезу технических объектов в САПР. Рассматриваются задачи структурного синтеза и параметрической оптимизации. Описываются методы поиска экстремума в задачах оптимального проектирования.

В гл. 7 даются характеристики, позволяющие оценивать качество создаваемой САПР, рассматриваются методы имитационного моделирования аппаратных и программных средств САПР.

Содержание книги построено на материале различных литературных источников, а также на оригинальных результатах работ по созданию подсистем САПР и на базе курсов лекций, прочитанных авторами в МВТУ им. Н. Э. Баумана, Рязанском и Таганрогском радиотехнических институтах.

Учебник предназначен для студентов высших учебных заведений, обучающихся по специальности 0648, и может быть рекомендован студентам смежных специальностей 0608, 0646, 0647, 0656, 0705 и др., а также аспирантам и инженерно-техническим работникам.

Предисловие, гл. 6, 7 и § 1.3, 1.4 и 2.6 написаны В. П. Корячко, гл. 2, 3 и § 1.1, 1.2, 1.5, 1.6 и 4.8 — В. М. Курейчиком, гл. 4, 5, § 6.5, 7.4 и 7.7 — И. П. Норенковым.

Авторы благодарны рецензентам профессору, доктору техн. наук Е. Л. Глорнозову, сотрудникам кафедры «Конструирование электронно-вычислительной аппаратуры» Ленинградского института точной механики и оптики, а также научному редактору В. Г. Одинокovu за ценные замечания, улучшившие содержание книги.

Замечания и пожелания следует направлять по адресу: 113114, Москва, М-114, Шлюзовая наб., 10, Энергоатомиздат.

*Авторы*

## ГЛАВА 1

# МЕТОДОЛОГИЯ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

### 1.1. ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТИРОВАНИИ

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 ЭВА. Первичное описание ЭВА, представленное в заданной форме, называется заданием на проектирование. В задании на проектирование ЭВА должны быть сведения о назначении ЭВА, ее 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.

Проектное решение или их совокупность, удовлетворяющие заданным требованиям, необходимые для создания объекта проектирования, будут являться *результатом проектирования*. 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*. Примерами проектных операций являются составление таблиц с данными вычисления, вычерчивание топологии, ввод и вывод данных, набивка перфокарт и т. п. Соответственно 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 8 стадий: предпроектные исследования; техническое задание; техническое предложение; эскизный проект; технический проект; рабочий проект; изготовление; 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 прорабатывают схемные, конструкторские и технологические решения, проводят испытания, изготовление.

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

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.

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.

Методология БИП базируется на трех концепциях: *разбиение и локальная оптимизация; абстрагирование; 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 по-

строении формальных математических моделей, отражающих только значимые в данных условиях свойства объектов. Повторяемость заключается в использовании существующего опыта проектирования.

Основное достоинство БИП — это упрощение процесса проектирования и получение возможности решать задачи проектирования доступными средствами.

Использование БИП помогает: упростить решение проблемы хранения данных, сократить размерность выполняемых программ и время проектирования, применять 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 (ВП), т. е. проектирование снизу вверх, характеризуется решением сначала задач низких иерархических уровней с последовательным переходом к решению задач более высоких уровней. Нисходящее проектирование (НП), т. е. проектирование сверху вниз, является противоположным по отношению к ВП.

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

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  $i_2$  происходит 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. Ниже приведен пример структурирования описания ЭВМ:

## Вертикальные уровни

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

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; расчеты 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 ЭВА.

Конструкторское проектирование заклю-

чается 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 следующие: покрытие функциональных схем, т. е. получение принципиальных электрических схем; конструкторский расчет геометрических размеров компонентов и площади размещения; компоновка элементов; размещение элементов с учетом конструкторских схмотехнических 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 ЭВА состоит из четырех основных горизонтальных уровней: системного, логического, схмотехнического, 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 вентилях и триггеров.

Алгоритмическое проектирование используется для разработки программного обеспечения ЭВА. 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.

## 1.2. ЗАДАЧИ ПРИНЯТИЯ РЕШЕНИЙ В САПР

Задачи проектирования делят на задачи синтеза и анализа. Под *синтезом* понимается построение описания системы по заданному функционированию. *Анализ* — это определение функционирования по заданному описанию системы. Задачи синтеза связаны с созданием проектных документов и самого проекта, а задачи анализа связаны с оценкой проектных документов.

Различают синтез структурный и параметрический. Цель структурного синтеза — получение структурных схем объекта, содержащих сведения о составе элементов и способах соединения их между собой. Цель параметрического синтеза — определение числовых значений параметров элементов. Синтез называется *оптимизацией*, если определяются наилучшие в заданном смысле структуры и значения параметров. При расчетах оптимальных значений параметров при заданной структуре говорят о параметрической оптимизации. Задачу выбора оптимальной структуры называют *структурной оптимизацией*.

При проектировании на основе САПР имеется возможность получать множество решений различных задач. Выделение некоторого подмножества решений задач относится к проблемам выбора и принятия решений. *Задачей принятия решений* называют кортеж  $\alpha = \langle W, \Theta \rangle$  (где  $W$  — множество вариантов решений задачи;  $\Theta$  — принцип оптимальности, дающий представление о качестве вариантов, в простейшем случае правило предпочтения вариантов). Решением задачи  $\alpha$  называют множество  $W_{\text{оп}} \subseteq W$ , полученное на основе принципа оптимальности.

Задачи принятия решений классифицируют по наличию информации о множестве  $W$  и принципе оптимальности  $\Theta$ .

Задачу, где  $W$  и  $\Theta$  могут быть неизвестными, называют *общей задачей принятия решений*. Данные для получения  $W_{\text{оп}}$  определяют в этой задаче в процессе решения. Задачу с неизвестным  $W$  называют *задачей выбора*, а задачу с известными  $W$  и  $\Theta$  называют *задачей оптимизации*. В САПР встречаются все три вида перечисленных задач.

В задачах проектирования свойства элементов множе-

ства  $W$  помогают находить решение. Если произвольное свойство варианта  $w_1 \in W$  выразить числом  $K = \{1, 2, \dots\}$ , т. е. предположить, что имеется отображение  $\varphi: W \rightarrow K$ , то такое свойство называют критерием, а число  $\varphi(W_i)$  — оценкой варианта  $W_i$  по критерию. Критериальным пространством считают пространство  $K_m$ , координаты точек которого — оценки по соответствующим критериям.

Например, пусть надо определить трассу, соединяющую две БИС на подложке. Различные возможные пути, соединяющие БИС 1 и БИС 2, будут вариантами. Пользователь или ЭВМ в соответствии с алгоритмом учитывает длину, стоимость, число изгибов, число пересечений и т. п. Значение длины трассы можно выразить числом, длину считать критерием.

Задачу  $\alpha$  решают следующим образом. Составляют множество  $W$ , если это возможно, т. е. определяют варианты, а затем решают задачу выбора. Отметим, что задача построения  $W$  в общем случае является задачей выбора. Следовательно, общую задачу принятия решений можно свести к решению последовательных задач выбора. В принятии решений в общем случае участвуют ЭВМ, лицо, принимающее решения (ЛПР), например проектировщик, эксперт, дающий оценки вариантам, и консультант.

Частным случаем общей задачи принятия решений является задача принятия решений в условиях неопределенности, возникающая, когда необходимо действовать в не полностью известной ситуации. Она часто формулируется как задача поиска одного наилучшего решения на заданном множестве допустимых решений.

Неизбежной платой за попытку получить решение в условиях неполной информации об объекте проектирования и его поведении является возможность ошибочных решений. Поэтому в такой ситуации ЛПР должно вырабатывать такую стратегию в отношении принятия решений, которая хотя и не исключает возможность принятия неправильных решений, но сводит к минимуму связанные с этим нежелательные последствия. Для уменьшения неопределенности и возможных потерь ЛПР может провести эксперимент. Это позволит сделать знания об исследуемом объекте сколь угодно полными и действовать уже в условиях определенности. Однако этому мешают два обстоятельства: 1) на проведение эксперимента требуется время, тогда как решение во многих случаях нужно принять быстро; 2) эксперимент требует затраты средств и может стоить дороже того

выигрыша, который дают добавочные знания, полученные в результате эксперимента.

Поэтому ЛПР должно принять решение о том, нужно ли проводить эксперимент, а если нужно, то на каком уровне его закончить и какие действия предпринять после окончания эксперимента.

Раздел математической теории принятия решений в условиях неполной определенности называют *теорией статистических решений*.

### 1.3. ВЫБОР КРИТЕРИЕВ ОПТИМАЛЬНОСТИ

На разных этапах проектирования технических объектов перед разработчиками встает задача выбора наилучшего варианта из множества допустимых проектных решений, удовлетворяющих предъявляемым требованиям.

Само по себе принятие решения есть компромисс. Принимая решение, необходимо взвешивать суждения о ценности, что включает рассмотрение многих факторов, в том числе экономических, технических, научных, социальных и чисто человеческих. Принять «правильное» решение — значит выбрать такую альтернативу из числа возможных, в которой с учетом всех разнообразных факторов будет оптимизирована общая ценность. Задача оптимального проектирования заключается в определении вектора  $\mathbf{X} = (x_1, \dots, x_m)$  оптимальных конструктивных параметров проектируемого объекта исходя из технических и технико-экономических критериев оптимальности и поставленных ограничений. Переменные проектирования  $\mathbf{X}$  являются внутренними переменными, допускающими варьирование. Использование рационального комплекса критериев представляет собой основной метод творческой технической деятельности при оптимальном проектировании. От того, как составлен комплекс критериев, зависит успех разработки. Процесс принятия решения при оптимальном проектировании характеризуют следующие основные черты: наличие цели (критериев оптимальности) и альтернативных вариантов проектируемого объекта и учет существенных факторов при проектировании.

Понятие «оптимальное решение» при проектировании имеет вполне определенное толкование — лучшее в том или ином смысле проектное решение, допускаемое обстоятельствами. В подавляющем большинстве случаев одна и та же техническая задача может быть решена несколькими спо-

собами, приводящими не только к различным выходным характеристикам, схемам и конструкциям, но даже и к физическим принципам, положенным в основу построения объекта. При этом одно из решений может превосходить другое по одним свойствам и уступать ему по другим. В этих условиях часто чрезвычайно трудно сказать не только какая из систем оптимальна, но даже какая из них предпочтительнее.

Если при проектировании технических объектов или систем можно выделить один параметр, которому отдается безусловное предпочтение и который наиболее полно характеризует свойства проектируемого объекта, то естественно этот параметр принять за целевую функцию. Такой выбор целевой функции лежит в основе критериев оптимальности, называемых частными критериями. При оптимизации по частным критериям задача проектирования сводится к задаче оптимизации выбранной целевой функции при условии соблюдения определенных ограничений. При этом одна часть параметров подпадает под категорию ограничений, а другая часть параметров, на которые не накладываются ограничения, принимается такой, какой получилась при оптимизации целевой функции.

Для решения однокритериальных задач создан и уже давно успешно применяется развитый математический аппарат, в том числе аппарат исследования операций.

Альтернативы, перед которыми оказываются разработчики новых технических систем, в большинстве случаев не могут быть отнесены к однокритериальным задачам. Любая техническая система, особенно система сложная, характеризуется многими параметрами, определяющими ее качество и ценность. Среди этих параметров есть такие, значения которых желательно всемерно увеличивать, есть и такие, которые желательно минимизировать.

Существующие взаимосвязи между параметрами любой технической системы и ограничения, накладываемые на параметры, не позволяют конструкторам системы увеличить, насколько это желательно, все те характеристики, возрастание которых повышает качество системы, и уменьшить до предела все те параметры, минимизация которых улучшает систему.

Таким образом, ограничения и связи между отдельными параметрами технической системы приводят к необходимости идти на компромисс и выбирать для каждой характеристики не максимально возможное в принципе значение, а



меньшее, но такое, при котором и другие важные характеристики тоже будут иметь приемлемые значения. Поэтому при выборе варианта технической системы нельзя ограничиваться сравнением по одной какой-либо характеристике, а необходимо принимать во внимание всю их совокупность. Задачи проектирования, проводимые по нескольким критериям оптимизации, носят название *многокритериальных*, или *задач векторной оптимизации*.

Все известные методы векторной оптимизации непосредственно или косвенно сводят решаемые задачи к задачам скалярной оптимизации. Иначе говоря, частные критерии  $F_i(\mathbf{X})$ ,  $i = \overline{1, n}$ , тем или иным способом объединяются в составной критерий  $F(\mathbf{X}) = \Phi(F_1(\mathbf{X}), \dots, F_n(\mathbf{X}))$ , который затем максимизируется (или минимизируется). Если составной критерий получается в результате проникновения в физическую суть функционирования системы и вскрытия объективно существующей взаимозависимости между частными критериями и составным критерием, то оптимальное решение является объективным. Однако отыскание подобной взаимозависимости чрезвычайно сложно, а может быть, и не всегда возможно. Поэтому на практике составной критерий обычно образуют путем формального объединения частных критериев, что неизбежно ведет к субъективности получаемого «оптимального» решения. Составной критерий иногда называют *обобщенным* или *интегральным критерием*.

В зависимости от того, каким образом частные критерии объединяются в обобщенный критерий, различают критерии аддитивные, мультипликативные и минимаксные (максиминные).

Если оптимизация ведется без учета статистического разброса характеристик, то соответствующий критерий оптимальности называют *детерминированным критерием*, если разброс параметров учитывается, то имеем *критерий статистический*. Статистические критерии оптимальности более полно отражают представление о качестве объектов проектирования, однако их использование, как правило, при автоматизированном проектировании ведет к значительному увеличению затрат машинного времени.

Рассмотрим наиболее часто встречающиеся на практике способы выбора критериев оптимальности.

**Частные критерии.** При проектировании по частным критериям в качестве целевой функции  $F(\mathbf{X})$  принимается наиболее важный выходной параметр проектируемого объ-

екта, все остальные параметры в виде соответствующих условий работоспособности относятся к ограничениям. В этом случае задача оптимального проектирования является однокритериальной задачей математического программирования: максимизировать (или минимизировать) значение целевой функции

$$F(\mathbf{X}) \rightarrow \max (\min)$$

при наличии системы ограничений на параметры проектируемого объекта.

Из постановки задачи математического программирования вытекает, что параметры, для которых выполняются ограничения в виде строгих неравенств, имеют определенный запас по сравнению с заданными техническими требованиями. Ряд параметров, для которых условия работоспособности имеют вид равенств, запасов вообще не имеет, и любые изменения технических требований для этих параметров приводят как к изменению характеристик и структуры проектируемого объекта, так и к изменению значения целевой функции.

Частные критерии довольно широко используют при проектировании технических объектов различного назначения.

**Пример 1.1.** Проектирование технологического оборудования. Переносной автомат для забивания стальных дюбелей в бетонные стены состоит из корпуса с магазином, содержащим запас дюбелей, подающе-спускового механизма с зарядами и ствола (рис. 1.1). Требуется опре-

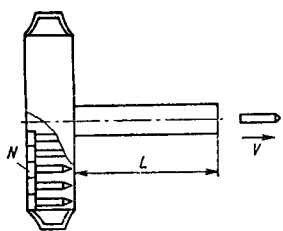


Рис. 1.1. Технологический автомат

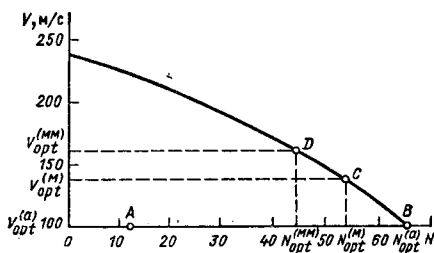


Рис. 1.2. Область решения задачи оптимизации технологического автомата

делить основной конструктивный параметр автомата — длину ствола  $L$  — при следующих исходных данных: число дюбелей, помещающихся в магазине,  $N \geq 12$ , масса одного дюбеля с расходуемым на него зарядом  $m = 50$  г, масса ствола  $1,6$  кг/м, масса корпуса  $2$  кг, критерий оптимальности — минимальная масса заряженного автомата.

При фиксированной величине заряда и заданной массе дробеля скорость  $V$  выбрасывания дробеля связана с длиной ствола  $L$  соотношением  $V = k\sqrt{L}$ , где  $k = 150 \text{ м}^{0,5}/\text{с}$ . Минимально допустимая скорость дробеля определяется экспериментально:  $V_{\min} = 100 \text{ м/с}$ . Масса автомата при минимально допустимом числе дробелей в магазине определяется как  $F(L) = 1,6L + 0,05N + 2 = 1,6L + 2,6$ .

Задача проектирования автомата сводится к минимизации целевой функции

$$F(L) = 1,6L + 2,6$$

при ограничении

$$150\sqrt{L} \geq 100.$$

Решение задачи оптимизации имеет вид:  $L = 0,445 \text{ м}$ ,  $F(L) = 3,31 \text{ кг}$ .

График области компромисса для массы автомата 6 кг показан на рис. 1.2. Здесь точка  $A$  соответствует оптимальному решению данной задачи по критерию минимума массы автомата.

**Аддитивные критерии.** В аддитивных критериях целевая функция образуется путем сложения нормированных значений частных критериев. Частные критерии имеют различную физическую природу и в соответствии с этим — различную размерность. Поэтому при образовании обобщенного критерия следует оперировать не с «натуральными» критериями, а с их нормированными значениями. Нормированные критерии представляют собой отношение «натурального» частного критерия к некоторой нормирующей величине, измеряемой в тех же единицах, что и сам критерий. При этом выбор нормирующего делителя должен быть логически обоснован. Возможны несколько подходов к выбору нормирующего делителя.

Первый подход предлагает принимать в качестве нормирующего делителя директивные значения параметров, заданные заказчиком. Логически слабым моментом такого подхода является негласное предположение того, что в ТЗ на проектируемый объект заданы оптимальные значения параметров объекта и что совокупность заданных значений критериев рассматривается как образцовая.

Второй подход предполагает выбор в качестве нормирующих делителей максимальных значений критериев, достигаемых в области существования проектных решений (в области компромисса). Возможен подход, при котором в качестве нормирующих делителей выбирают разность между максимальным и минимальным значениями критерия в области компромисса.

Выбор подхода к формированию безразмерной формы частных критериев в значительной степени носит субъективный характер и должен обосновываться в каждом конкретном случае.

Пусть при проектировании некоторого объекта существует  $n$  частных критериев. Тогда целевая функция задачи оптимизации при применении аддитивного критерия имеет вид

$$F(\mathbf{X}) = \sum_{i=1}^n c_i \frac{F_i(\mathbf{X})}{F_i^{(0)}(\mathbf{X})} = \sum_{i=1}^n c_i f_i(\mathbf{X}). \quad (1.1)$$

Здесь  $c_i$  — весовой коэффициент  $i$ -го частного критерия;  $F_i^{(0)}(\mathbf{X})$  —  $i$ -й нормирующий делитель;  $f_i(\mathbf{X})$  — нормированное значение  $i$ -го частного критерия.

Функция (1.1) позволяет осуществлять компромисс, при котором улучшение значения одного нормированного частного критерия компенсирует ухудшение значений других.

Введение весовых коэффициентов должно учитывать различную значимость частных критериев при формировании аддитивного критерия. Определение весовых коэффициентов сталкивается с серьезными трудностями и обычно сводится либо к использованию формальных процедур, либо к применению экспертных оценок.

С появлением обобщенного критерия исчезают логические проблемы, связанные с установлением взаимосвязей между частными критериями различной размерности и выбором наилучшего варианта проектируемого объекта, и остаются лишь вычислительные трудности. Но аддитивный критерий имеет ряд недостатков, главный из которых состоит в том, что он не вытекает из объективной роли частных критериев в функционировании объекта или системы и выступает поэтому как формальный математический прием, придающий задаче удобный для решения вид. Другой недостаток заключается в том, что в аддитивном критерии может происходить взаимная компенсация частных критериев. Это значит, что значительное уменьшение одного из критериев вплоть до нулевого значения может быть покрыто возрастанием другого критерия. Для ослабления этого недостатка следует вводить ограничения на минимальные значения частных критериев и их весовых коэффициентов. Несмотря на слабые стороны обобщенный аддитивный критерий позволяет в ряде случаев успешно решать многокритериальные задачи и получать полезные результаты.

**Пример 1.2.** По исходным данным примера 1.1 определить конструктивные параметры  $L$  и  $N$  переносного автомата при условии, что масса заряженного автомата не должна превышать 6 кг, а частными критериями эффективности автомата являются скорость выбрасывания дубеля  $V$  и число дубелей  $N$ , помещающихся в магазин. Выбор этих критериев объясняется тем, что чем выше  $V$ , тем надежнее дубеля проникают в бетон любой марки, а чем больше  $N$ , тем удобнее работать с автоматом. По мнению экспертов оба критерия  $V$  и  $N$  в нормированном виде имеют одинаковую важность.

Найдем оптимальное решение с помощью аддитивного критерия. Для нормирования найдем  $N_{\max}$  и  $V_{\max}$ . Величину  $N_{\max}$  определяют из условия,  $V_{\min} = 100$  м/с. Уравнение баланса масс имеет вид

$$1,6L + 0,05N + 2 = 6.$$

Из этого уравнения следует, что  $N_{\max} = 65$ . Для отыскания  $V_{\max}$  будем считать, что в автомате находится только один дубель. Тогда  $V_{\max} = 150 \sqrt{L_{\max}} = 236$  м/с. Нормированные частные критерии будут иметь вид

$$f_1(V) = V/236; f_2(N) = N/65.$$

Аддитивный критерий эффективности автомата

$$F(V, N) = f_1(V) + f_2(N) = V/236 + N/65.$$

Для определения максимального значения аддитивного критерия  $F(V, N)$  с учетом ограничения на массу автомата воспользуемся методом неопределенных множителей Лагранжа. В результате решения задачи оптимизации получаем  $V_{\text{opt}}^{(a)} = 100$  м/с,  $L_{\text{opt}}^{(a)} = 0,445$  м,  $N_{\text{opt}}^{(a)} = 65$ . На рис. 1.2 данному решению соответствует точка  $B$ .

**Мультипликативные критерии.** Аддитивные критерии основаны на использовании принципа справедливой компенсации абсолютных значений нормированных частных критериев. Но в ряде задач проектирования более целесообразным является оперирование не с абсолютными, а с относительными изменениями значений частных критериев.

Принцип справедливой относительной компенсации формулируется следующим образом: справедливым следует считать такой компромисс, когда суммарный уровень относительного снижения значений одного или нескольких критериев не превышает суммарного уровня относительного увеличения значений других критериев.

В математической форме условие оптимальности на основе принципа справедливой относительной компенсации имеет вид

$$\sum_{i=1}^n \frac{\Delta F_i(X)}{F_i(X)} = 0, \quad (1.2)$$

где  $\Delta F_i(X)$  — приращение величины  $i$ -го критерия;  $F_i(X)$  — первоначальная величина  $i$ -го критерия.

Полагая  $\Delta F_i(X) \ll F_i(X)$ , можно представить (1.2) как дифференциал натурального логарифма, тогда

$$\sum_{i=1}^n \frac{\Delta F_i(X)}{F_i(X)} = \sum_{i=1}^n d(\ln F_i(X)) = d\left(\ln \prod_{i=1}^n F_i(X)\right) = 0. \quad (1.3)$$

Из выражения (1.3) следует, что принцип справедливой относительной компенсации приводит к мультипликативному обобщенному критерию оптимальности

$$F(X) = \prod_{i=1}^n F_i(X).$$

Мультипликативный критерий образуется путем простого перемножения частных критериев в том случае, если все они имеют одинаковую важность. В случае неравноценности частных критериев вводятся весовые коэффициенты  $c_i$  и мультипликативный критерий принимает вид

$$F(X) = \prod_{i=1}^n F_i^{c_i}(X). \quad (1.4)$$

Достоинством мультипликативного критерия является то, что при его использовании не требуется нормировка частных критериев. Недостатки критерия: критерий компенсирует недостаточную величину одного частного критерия избыточной величиной другого и имеет тенденцию сглаживать уровни частных критериев за счет неравнозначных первоначальных значений частных критериев.

**Пример 1.3.** По исходным данным примера 1.2 определить конструктивные параметры технологического автомата по мультипликативному критерию вида

$$F(V, N) = VN.$$

Для отыскания максимума функции  $F(V, N)$  составляем функцию Лагранжа

$$\Phi(V, N, \lambda) = VN + \lambda(4 - 1,6L - 0,05N),$$

где  $\lambda$  — множитель Лагранжа.

Выражая длину ствола  $L$  через скорость  $V$  и решая задачу оптимизации, получаем  $V_{\text{opt}}^{(M)} = 137$  м/с,  $L_{\text{opt}}^{(M)} = 0,833$  м,  $N_{\text{opt}}^{(M)} = 53$ , что соответствует точке  $C$  на рис. 1.2.

Использование мультипликативного критерия в задаче оптимизации привело к другим значениям параметров технологического автомата по сравнению с решением задачи с аддитивным критерием оптимальности.

Это объясняется тем, что диапазоны взаимной компенсации абсолютных и относительных изменений частных критериев  $V$  и  $N$  неодинаковы. Поэтому в каждом конкретном случае технического проектирования следует тщательно анализировать и обосновывать целесообразность учета либо абсолютных, либо относительных изменений значений частных критериев и в зависимости от степени важности этих отклонений выбирать либо аддитивный, либо мультипликативный критерий оптимальности.

**Минимаксные критерии.** В теории векторной оптимизации особое место занимает принцип компромисса, основанный на идее равномерности. На базе этого принципа работают минимаксные (максиминные) критерии.

Сущность принципа максимина заключается в следующем. При проектировании сложных технических объектов при наличии большого числа частных критериев довольно трудно, а зачастую и невозможно установить аналитическую взаимосвязь между критериями. Поэтому, основываясь на идее равномерного компромисса, стараются найти такие значения переменных проектирования  $\mathbf{X} = (x_1, \dots, x_m)$ , при которых нормированные значения всех частных критериев становятся равными между собой, т. е.

$$f_i(\mathbf{X}) = K, i = \overline{1, n}. \quad (1.5)$$

С учетом весовых коэффициентов важности частных критериев выражения (1.5) трансформируются в соотношения вида

$$c_i f_i(\mathbf{X}) = K, i = \overline{1, n}. \quad (1.6)$$

При большом числе частных критериев из-за сложных взаимосвязей иногда чрезвычайно трудно добиться выполнения соотношений (1.5) и (1.6). В этом случае оказывается полезным применение принципа максимина, заключающегося в такой вариации значений переменных проектирования  $\mathbf{X}$ , при которой последовательно «подтягиваются» те нормированные критерии, численные значения которых в исходном решении оказались наименьшими. Вследствие того что операции производятся в области компромисса, подтягивание «отстающего» критерия неизбежно приводит к снижению значений части остальных критериев. Но при проведении ряда шагов можно добиться определенной степени уравнивания противоречивых (конфликтных) частных критериев, что и является целью принципа максимина.

Формально принцип максимина формулируется следующим образом: нужно выбрать такое  $\mathbf{X}^{(0)} \in \mathbf{X}$ , на котором

реализуется максимум из минимальных значений частных критериев, т. е.

$$F(\mathbf{X}^{(0)}) = \max_{\mathbf{X}} \min_i \{f_i(\mathbf{X})\}, \quad i = \overline{1, n}, \quad \mathbf{X} = (x_1, \dots, x_m).$$

Такой принцип выбора  $\mathbf{X}^{(0)}$  иногда носит название «принципа гарантированного результата». Он заимствован из теории игр, где, по существу, является основным принципом.

Если частные критерии  $f_i(\mathbf{X})$  следует минимизировать, то самым «отстающим» критерием является тот, который принимает максимальное значение. В этом случае принцип равномерной компенсации формулируется в виде минимаксной задачи:

$$F(\mathbf{X}^{(0)}) = \min_i \max_{\mathbf{X}} \{f_i(\mathbf{X})\}, \quad i = \overline{1, n}, \quad \mathbf{X} = (x_1, \dots, x_m). \quad (1.7)$$

Для обоснования геометрической интерпретации принципа минимакса приведем ряд определений из теории выпуклых множеств.

Пусть  $Q$  — некоторое множество, определенное в пространстве  $E^n$ . Множество  $Q$  называют *выпуклым*, если отрезок, соединяющий любые две точки этого множества, целиком принадлежит этому множеству. Другими словами,  $Q$  — выпуклое множество, если для любых  $x^{(i)}, x^{(j)} \in Q$  и любого  $0 \leq \lambda < 1$  справедливо

$$x = \lambda x^{(i)} + (1 - \lambda) x^{(j)} \in Q.$$

Величину  $x$  называют *средневзвешенной* по элементам  $x^{(i)}$  и  $x^{(j)}$  с весами  $\lambda$  и  $(1 - \lambda)$ .

Пусть  $A = \{A^{(1)}, \dots, A^{(k)}\}$  — конечное множество точек в пространстве  $E^n$ . Конечное множество точек (рис. 1.3, а) не является выпуклым,

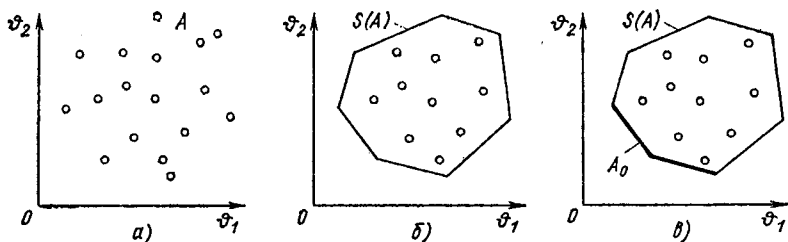


Рис. 1.3. Иллюстрация понятия выпуклой оболочки и области компромисса

однако может быть заключено в выпуклую оболочку  $S(A)$  (рис. 1.3, б). Выпуклой оболочкой  $S(A)$  конечного точечного множества  $A$  называют пересечение всех выпуклых множеств  $Q_i$ , подмножествами которых является  $A$ . В частности, если  $A \subset Q_1$  и  $A \subset Q_2$ , то  $S(A) \subset Q_1 \cap Q_2$ .



Из данного определения следует, что выпуклая оболочка  $S(A)$  является наименьшим выпуклым множеством, содержащим  $A$ . Выпуклой оболочкой конечного точечного множества  $A$  на плоскости является выпуклый многоугольник, вершинами которого являются крайние точки множества  $A$ , а выпуклой оболочкой конечного множества  $A$  в пространстве  $E^n$  — выпуклый многогранник. Точку  $x^*$  называют крайней точкой конечного множества  $A$ , если ни для каких  $A^{(i)}, A^{(j)} \in A$  она не может быть представлена в виде

$$x^* = \lambda A^{(i)} + (1 - \lambda) A^{(j)}, \quad 0 < \lambda < 1.$$

Заметим, что в этом определении  $\lambda$  не может принимать значений 0 и 1. Это означает, что крайняя точка не может лежать внутри отрезка, соединяющего любые две точки множества  $A$ , а может быть лишь концевой точкой этого отрезка. Выпуклая оболочка конечного множества  $A$  есть множество средневзвешенных по элементам множества  $A$ .

Геометрическая интерпретация принципа минимакса заключается в следующем.

Пусть проектируется некоторый объект по  $n$  частным критериям  $\phi_i = f_i(X)$ ,  $i = \overline{1, n}$ . Каждый вариант объекта может быть представлен в пространстве  $E^n$  в виде точки  $A^{(l)}$  с координатами  $A^{(l)} = (\phi_1^{(l)}, \dots, \phi_n^{(l)})$ , а множество вариантов может быть отображено в конечное множество точек  $A = \{A^{(1)}, \dots, A^{(k)}\}$ , заключенное в выпуклую оболочку  $S(A)$ .

Таким образом, область принятия решений при проектировании ограничена выпуклой оболочкой  $S(A)$  в пространстве  $E^n$ .

Пусть все частные критерии минимизируются. Тогда область компромисса является левая нижняя граница выпуклой оболочки  $S(A)$ , а решение должно находиться в области компромисса (рис. 1.3,  $\theta$ ). В общем случае при неравнозначных критериях  $\phi_i = f_i(X)$  решение на основе принципа равномерной компенсации будет соответствовать такой точке  $A^{(0)}$ , лежащей в области компромисса, для которой будут удовлетворяться соотношения

$$c_i \phi_i = K, \quad c_i \geq 0, \quad \sum_{i=1}^n c_i = 1, \quad i = \overline{1, n}. \quad (1.8)$$

Заметим, что направление, определяемое вектором  $C = (c_1, \dots, c_n)$ , задается в первом ортанте в пространстве  $E^n$ . Произвольный вектор весовых коэффициентов  $C$ , удовлетворяющий соотношениям (1.8), будем интерпретировать как предпочтение частных критериев  $\phi_i = f_i(X)$  друг перед другом, выраженное в количественной шкале.

Остановимся на определении направления, порождаемого вектором  $C$  в пространстве  $E^n$ . Это направление задается углами  $\beta_i$ ,  $i = \overline{1, n}$ , между осями координат и радиусом-вектором  $C$ . Тогда

$$\cos \beta_i = \frac{(\vartheta^*, e_i)}{\|\vartheta^*\| \|e_i\|} = \frac{\vartheta_i^*}{\sqrt{\sum_{i=1}^n \vartheta_i^{*2}}}, \quad i = \overline{1, n},$$

где  $e_i = (0, \dots, 0, 1, 0, \dots, 0)$  — орт оси  $\vartheta_i$ ;  $\vartheta^* = (\vartheta_1^*, \dots, \vartheta_n^*)$  — точка, находящаяся на луче  $C$ .

Исходя из отношений для различных пар углов  $\beta_i$  и  $\beta_j$  ( $i, j = \overline{1, n}$ ), запишем систему линейно независимых уравнений, из которой могут быть найдены неизвестные направляющие косинусы:

$$\cos \beta_i / \cos \beta_j = \vartheta_i^* / \vartheta_j^*, \quad i, j = \overline{1, n}, \quad \sum_{i=1}^n \cos^2 \beta_i = 1. \quad (1.9)$$

С другой стороны, в силу соотношений (1.8) для точки  $\vartheta^*$  справедливо

$$\vartheta_i^* / \vartheta_j^* = c_j / c_i, \quad i, j = \overline{1, n}, \quad i \neq j.$$

Учитывая эти соотношения, можно переписать (1.9) следующим образом:

$$\cos \beta_i / \cos \beta_j = c_j / c_i, \quad i, j = \overline{1, n}, \quad i \neq j, \quad \sum_{i=1}^n \cos^2 \beta_i = 1. \quad (1.10)$$

Решая (1.10), получим выражение для направляющих косинусов вектора  $C$ :

$$\cos \beta_i = \prod_{j=1}^n c_j / \sqrt{\sum_{q=1}^n \prod_{\substack{j=1 \\ j \neq q}}^n c_j^2}, \quad i = \overline{1, n}. \quad (1.11)$$

Если все частные критерии равноценны, т. е.  $c_i = 1/M$ ,  $i = \overline{1, n}$ , то

$$\cos \beta_i = 1/\sqrt{M}, \quad i = \overline{1, n}.$$

При наличии двух частных критериев для равноценных критериев направляющие косинусы имеют вид

$$\cos \beta_1 = \cos \beta_2 = 1/\sqrt{2},$$

что соответствует биссектрисе координатного угла  $\vartheta_1 0 \vartheta_2$  (рис. 1.4, а).

Если  $c_1 > c_2$ , что указывает на то, что частный критерий  $\vartheta_1$  предпочтительней второго, то  $\cos \beta_1 \ll \cos \beta_2$  в силу соотношения (1.11). Так как  $\beta_1 + \beta_2 = \pi/2$ , то вектор, определяемый этим предпочтением, прохо-

дит вблизи от  $0\theta_2$ , т. е. около оптимума (минимума) первого критерия (рис. 1.4, б).

**Пример 1.4.** По исходным данным примера 1.2 определить конструктивные параметры технологического аппарата по минимаксному критерию. Критериями эффективности являются длина ствола  $L$  и число дубелей  $N$  в магазине.

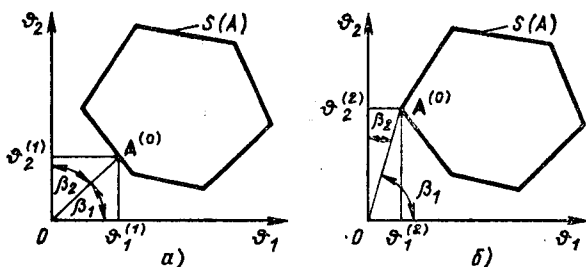


Рис. 1.4. Геометрическая интерпретация принципа минимакса

Желательно, чтобы оба критерия имели максимально возможное численное значение. В нормированном виде критерии эффективности определены в примере 1.2:

$$f_1(L) = k\sqrt{L}/V_{\max} = 150\sqrt{L}/236; f_2(N) = N/N_{\max} = N/65.$$

Решение по критерию минимакса имеет вид  $L_{\text{opt}}^{(\text{MM})} = 1,13$  м,  $N_{\text{opt}}^{(\text{MM})} = 44$ ,  $V_{\text{opt}}^{(\text{MM})} = 159,4$  м/с,  $f_1(L) = f_2(N) = 0,677$ , что соответствует точке D на рис. 1.2.

Сделаем краткие выводы по выбору критериев оптимальности при автоматизированном проектировании технических объектов. Выбор критерия оптимальности является сложной методологической проблемой и, как правило, может производиться неоднозначно. Источником сложности этой проблемы прежде всего служит противоречивость целей, преследуемых при проектировании любого нового технического объекта. Стоимость и надежность функционирования, энергоемкость и производительность, микроминиатюризация и массогабаритные параметры всегда находились и будут находиться в противоречии друг с другом. Поэтому если в ТЗ на проектирование сформулировано, что требуется оптимизировать один из параметров проектируемого объекта при соблюдении ограничительных требований на остальные параметры, то необходимо сформировать частный критерий  $F(X)$ . В этих случаях задача оптималь-

ного проектирования сводится к задаче оптимизации критерия  $F(\mathbf{X})$  с учетом заданных ограничений. При наличии нескольких критериев оптимальности аддитивный критерий выбирают тогда, когда существенное значение имеют абсолютные значения критериев при выбранном векторе параметров проектирования  $\mathbf{X}$ ; если же существенную роль играет изменение абсолютных значений частных критериев при вариации вектора переменных  $\mathbf{X}$ , то в этом случае целесообразно применять мультипликативный критерий оптимальности. Если перед разработчиком стоит задача достижения равенства нормированных значений конфликтных частных критериев, то оптимальное проектирование следует проводить по минимаксному критерию.

#### 1.4. МЕТОДЫ ЗАДАНИЯ ПРЕДПОЧТЕНИЯ НА МНОЖЕСТВЕ ЧАСТНЫХ КРИТЕРИЕВ

При автоматизированном проектировании новых технических объектов разработчик взаимодействует с техническими средствами САПР в интерактивном режиме. В процессе этого взаимодействия на основе анализа множества альтернативных вариантов проектных решений, получаемых с помощью технических и программных средств САПР, разработчик должен принять решение по выбору оптимального варианта проектируемого объекта, т. е. решить задачу выработки предпочтения среди некоторого множества альтернативных вариантов проектируемого объекта. Решение разработчик принимает на основе выбранных критериев. При существовании одного частного критерия принятие решения производится однозначно путем сравнения значений данного критерия для различных альтернативных вариантов.

В многокритериальных задачах оптимального проектирования возникает необходимость объективной оценки важности частных критериев, включаемых в аддитивный, мультипликативный или минимаксный критерий оптимальности. Оценивают важность частных критериев  $F_i(\mathbf{X})$ ,  $i = \overline{1, n}$ , с помощью весовых коэффициентов  $c_i$  согласно (1.1), (1.4) и (1.6), которые должны количественно отражать важность соответствующих частных критериев. Значения  $c_i$  выбирают исходя из анализа современного мирового уровня развития данной отрасли, из требований к проектируемому объекту и из существующих возможностей реализации этих требований. Открытие новых физических принципов и раз-

работка новых методов проектирования могут существенно влиять на значения весовых коэффициентов.

Рассмотрим основные подходы к решению задачи выработки предпочтения на множестве частных критериев.

**Экспертные оценки.** В теории экспертных оценок разработан ряд методов проведения экспертизы. Наиболее эффективными в проводимых исследованиях оказались методы ранжирования и приписывания баллов.

*Метод ранжирования* заключается в следующем. Пусть экспертиза проводится группой из  $l$  экспертов, которые являются квалифицированными специалистами в той области, где принимается решение. Метод ранжирования основан на том, что каждого эксперта просят расставить частные критерии  $F_i(\mathbf{X})$ ,  $i = \overline{1, n}$ , проектируемого объекта в порядке их важности. При этом цифрой 1 обозначают наиболее важный частный критерий (параметр), цифрой 2 — следующий по степени важности частный критерий и т. д. Эти ранги преобразовывают таким образом, что ранг 1 получает оценку  $n$ , ранг 2 — оценку  $(n-1)$  и т. д. до ранга  $n$ , которому присваивается оценка 1, где  $n$  — число частных критериев. Зная преобразованный ранг  $r_i^{(k)}$   $i$ -го критерия у  $k$ -го эксперта ( $k = \overline{1, l}$ ), весовые коэффициенты  $c_i$ ,  $i = \overline{1, n}$  определяют из следующего соотношения:

$$c_i = \frac{\sum_{k=1}^l r_i^{(k)}}{\sum_{i=1}^n \sum_{k=1}^l r_i^{(k)}}, \quad i = \overline{1, n}. \quad (1.12)$$

*Метод приписывания баллов* основан на том, что эксперты оценивают важность частного критерия по шкале 0—10. При этом разрешается оценивать важность дробными величинами или приписывать одну и ту же величину из выбранной шкалы нескольким критериям. Зная балл  $h_i^{(k)}$   $i$ -го критерия у  $k$ -го эксперта, весовые коэффициенты  $c_i$  можно найти из (1.12), заменив в нем  $r_i^{(k)}$  на

$$H_i^{(k)} = h_i^{(k)} / \sum_{i=1}^n h_i^{(k)}.$$

Последний называют весом, подсчитанным для  $i$ -го частного критерия  $F_i(\mathbf{X})$  на основе оценок  $k$ -го эксперта.

Важное место занимает обработка результатов экспертных оценок. Если рассматривать результаты оценок каждого из экспертов как реализации некоторой случайной ве-

личины, то к ним можно применять методы математической статистики.

В общем случае при определении степени важности частного критерия  $F_i(\mathbf{X})$  получают набор оценок  $r_i^{(k)}$ ,  $k = \overline{1, l}$ , подлежащих статистической обработке. Среднее значение оценки

$$r_i = \frac{\sum_{k=1}^l (\mu_k r_i^{(k)})}{\sum_{k=1}^l \mu_k},$$

где  $\mu_k$  — коэффициент авторитета  $k$ -го эксперта ( $0 \leq \mu \leq 1$ ).

Среднее значение оценки  $r_i$  выражает коллективное мнение группы экспертов. Степень согласованности мнений экспертов характеризуется величиной

$$\sigma^2 = \frac{1}{l} \sum_{k=1}^l (r_i^{(k)} - r_i)^2 / l,$$

называемой *дисперсией экспертных оценок*. Ясно, что чем меньше величины дисперсии, тем с большей уверенностью можно опираться на найденное значение  $r_i$  оценки степени важности частного критерия  $F_i(\mathbf{X})$ . Надежность экспертизы тем выше, чем меньшую долю среднего значения составляет среднеквадратический разброс оценок  $\sigma$ . Поэтому в качестве меры надежности приведенной экспертизы часто принимают  $\beta = \sigma / r_i$  и называют *вариацией*.

По среднему значению оценки  $r_i$  определяются весовые коэффициенты:

$$c_i = r_i / \sum_{i=1}^n r_i, \quad i = \overline{1, n}.$$

Статистическая обработка результатов экспертных оценок подобна статистической обработке результатов измерений. На достоверность экспертизы существенно влияют такие факторы, как численный состав экспертной группы, уровень компетентности экспертов, состав вопросов, предъявляемых экспертам, и т. д.

Индивидуальные экспертные оценки также носят на себе печать случайности: на суждения эксперта влияют не только такие стабильные факторы, как его знания и опыт, но и множество случайных факторов (настроение, самочувствие, обстановка и т. п.).

**Формальные процедуры.** Методика формального определения весовых коэффициентов основана на анализе современного уровня разви-

тия конкретной отрасли техники, в области которой производится проектирование. Каждому значению вектора параметров проектирования  $X_k$  соответствует альтернативный вариант  $S_k$  проектируемого объекта, качество (или эффективность) которого может быть оценено различными способами, и в частности в виде аддитивного критерия

$$F^{(k)}(X) = \sum_{i=1}^n c_i f_i^{(k)}(X), \quad k = \overline{1, m}, \quad (1.13)$$

где  $f_i^{(k)}(X)$  — нормированное значение  $i$ -го частного критерия для варианта  $S_k$ .

Для определения значений  $f_i^{(k)}(X)$  построим матрицу параметров  $\Theta = [\theta_i^{(j)}] \quad i = \overline{1, n}, j = \overline{1, m}$ , множества  $S = \{S_1, \dots, S_m\}$  альтернативных вариантов. В матрице  $\Theta$  вектор-строка  $\theta_j = (\theta_1^{(j)}, \dots, \theta_n^{(j)})$  описывает вариант  $S_j \in S$  проектируемого объекта. Для перехода от  $\theta_i^{(k)}$  к  $f_i^{(k)}(X) = f_i^k$  при фиксированном векторе переменных проектирования  $X_k$  введем совокупность директивных значений параметров  $\Theta_0 = (\theta_1^{(0)}, \dots, \theta_n^{(0)})$ , устанавливаемых в ТЗ на проектирование. Тогда нормированные (относительные) значения параметров определяются как

$$f_i^{(j)} = \theta_i^{(j)} / \theta_i^{(0)}, \quad i = \overline{1, n}, j = \overline{1, m}. \quad (1.14)$$

Определим в матрице  $\Theta$  величины  $\theta_i^*$  — экстремальные (наилучшие) значения всех параметров. Очевидно, что идеальный вариант объекта  $S_n$  должен описываться всеми  $\theta_i^*, i = \overline{1, n}$ .

Для оценки степени важности каждого параметра  $\theta_i$  (или каждого нормированного значения параметра  $f_i$ ) вводится система весов  $C = (c_1, \dots, c_n)$ , которая должна отражать усилия, необходимые для достижения экстремальных значений параметров (увеличить значения таких параметров, как производительность, надежность и другие, или уменьшить значения массогабаритных, стоимостных и энергетических параметров). Правильный выбор системы весов открывает возможность целенаправленно воздействовать на улучшение тех или иных параметров объекта путем увеличения соответствующих весов  $c_i$ . Конечно, для осуществления этой возможности система весов не должна быть застывшей, а должна быть гибкой и должна меняться в зависимости от назначения объекта и состояния развития данной отрасли техники в настоящий момент времени. В основу выбора системы весов положим принцип ограниченности общих затрат, необходимых для создания объекта. Это означает, что увеличение затрат на улучшение одних параметров неизбежно вызывает уменьшение затрат на улучшение других параметров.

Методика формального определения весовых коэффициентов базируется на выполнении последовательности процедур выработки предпочтения среди каждой пары показателей  $f_i$  и  $f_k$ .

Обозначим через  $f_{ik}$  значение показателя  $f_i$  в варианте объекта, в котором максимальные затраты сосредоточены на увеличении показателя  $f_h$ , а через  $f_i^*$  — наилучшее значение показателя  $f_i$  во множестве альтернативных вариантов  $S$ , т. е.

$$f_i^* = \text{ext}_{j=\overline{1, m}} \{ f_i^{(j)} \}; f_{ik} = f_i | f_k = f_k^*,$$

где  $f_{ik} = f_i | f_k = f_k^*$  — значение показателя  $f_i$  в варианте  $S_j$  для которого  $f_k^{(j)} = f_k^*$ .

При этом

$$\Delta f_i^* = \ln | f_i^* - f_i^{(0)} | = \ln | f_i^* - 1 | \quad (1.15)$$

определяет близость директивного значения показателя  $f_i^{(0)} = \phi_i^{(0)} / \phi_i^{(0)} = = 1$  к наилучшему  $f_i^*$ . Чем большее значение придается показателю  $f_i$ , тем меньше должно быть  $\Delta f_i^*$ . Следовательно, вес, придаваемый показателю  $f_i$ , должен быть обратно пропорционален величине  $\Delta f_i^*$ , что позволяет записать

$$c_i \approx 1 / \Delta f_i^*.$$

Величина

$$\Delta f_{ik} = \ln | f_i^* - f_{ik} | \quad (1.16)$$

определяет ухудшение показателя  $f_i$  в варианте объекта, в котором максимальные затраты сосредоточиваются на улучшении показателя  $f_k$ . Если величина  $\Delta f_{ik}$  мала, то это означает, что сосредоточение затрат на увеличении  $f_k$  не ухудшает существенно  $f_i^*$  и что, следовательно, поддержание  $f_i^*$  на высоком уровне не требует больших затрат и величина  $c_i$  должна быть взята малой, и наоборот. Следовательно, вес показателя  $f_i$  должен увеличиваться с увеличением  $\Delta f_{ik}$ . Это утверждение справедливо для любого  $k \in \{1, \dots, n\}$ , откуда следует, что

$$c_i \approx \sum_{k=1}^n \Delta f_{ik}. \quad (1.17)$$

С учетом (1.16) и (1.17) можно записать

$$c_i = \sum_{k=1}^n \Delta f_{ik} / \Delta f_i^* = \sum_{k=1}^n l_{ik}, \quad (1.18)$$

где  $l_{ik} = \Delta f_{ik} / \Delta f_i^*$  будут тем больше, чем большее значение придается показателю  $f_i$  и чем сильнее сказывается на снижении этого показателя сосредоточение усилий на показателе  $f_k$ .

Следовательно, величины  $l_{ik}$  могут рассматриваться как относительные веса, показывающие относительное превосходство (доминирование) показателя  $f_i$  над  $f_k$ .

Однако использование (1.18) для определения весов наталкивает-



ся на ряд трудностей. Во-первых, описанная методика определения  $l_{ik}$  оказывается неприемлемой при  $k=i$ , т. е. величины  $l_{ii}$ ,  $i=\overline{1, n}$ , остаются неопределенными. Ими, правда, можно было задаться произвольно, однако это вносит произвол в определение весов  $c_i$ . Во-вторых, величины  $l_{ik}$ , отражающие, как отмечалось, превосходство показателя  $f_i$  над показателем  $f_k$  и дающие соответствующие вклады в суммарный вес  $c_i$ , входят в (1.18) с коэффициентами, равными единице, т. е. не учитываются веса показателя  $f_k$ . В то же время превосходство  $l_{ik}$  параметра  $f_i$  над  $f_k$  может быть превосходством «сильного» над «слабым», поэтому значительность этого превосходства должна быть пропорциональна весу параметра  $f_k$ . Исходя из этого, следует заменить в (1.18) величины  $l_{ik}$  на  $c_k l_{ik}$ .

Такой подход приводит к необходимости использования для определения весов метода итерации. В нулевом приближении веса всех показателей принимаются одинаковыми и равными  $c_i^{(0)} = 1$ . Далее, если определены веса  $r$ -го порядка, то переход к весам  $r+1$ -го порядка будем осуществлять по формуле

$$c_i^{(r+1)} = \sum_{k=1}^n c_k^{(r)} l_{ik},$$

согласно которой веса первого, второго и т. д. порядков будут

$$c_i^{(1)} = \sum_{k=1}^n c_k^{(0)} l_{ik} = \sum_{k=1}^n l_{ik};$$

$$c_i^{(2)} = \sum_{k=1}^n c_k^{(1)} l_{ik};$$

.....

Данный процесс довольно быстро приводит к установившейся системе весов, не зависящих от последующих итераций и от величин  $l_{ii}$ . В связи с этим значения  $l_{ii}$  можно выбирать произвольно, например равными 0,5 или 1. Нормированные веса всех показателей после проведения  $t$  итераций определяются как

$$c_i = c_i^{(t)} / \sum_{i=1}^n c_i^{(t)}.$$

Если подмножество параметров  $\bar{\Theta}_j \subset \Theta_j$ , описывающее вариант объекта  $S_j \in \bar{S}$ , является подмножеством наилучших значений параметров  $\Theta_j = (\theta_1^{(j)} = \theta_1^*, \dots, \theta_q^{(j)} = \theta_q^*)$ , то возникает неопределенность при нахождении относительных весов  $c_i$  по формуле (1.18).

Для разрешения этой неопределенности перейдем от матрицы параметров  $\Theta = \|\theta_i^{(j)}\|$  к матрице относительных показателей  $A = [f_i^{(j)}]$ ,  $i=\overline{1, n}$ ,  $j=\overline{1, m}$ , элементы  $f_i^{(j)}$  которой образуются согласно (1.14).

Множество строк  $\bar{A} = \{A_1, \dots, A_n\}$  матрицы  $A$  расщепим на два подмножества  $\bar{A}_j$  и  $A'_j (\bar{A} = \bar{A}_j \cup A'_j)$ , причем подмножество  $\bar{A}_j$  (мощностью  $q$ ) является порождением множества  $\bar{\Theta}_j = (\vartheta_1^{(j)}, \dots, \vartheta_q^{(j)})$  посредством преобразования (1.14), а  $A'_j$  (мощностью  $n-q$ ) — порождением множества  $\Theta \setminus \bar{\Theta}_j = (\vartheta_{q+1}^{(j)}, \dots, \vartheta_n^{(j)})$ .

Сформируем  $q$  матриц  $A^{(k)}$  относительных параметров размерности  $(n-q+1)t$  таким образом, что множество строк  $\bar{A}^{(k)}$  матрицы  $A^{(k)}$  образуется из одного элемента  $A_k \in \bar{A}_j$  и всех элементов множества  $A'_j$ .

Для каждой матрицы  $A^{(k)}$ ,  $k = \overline{1, q}$ , определяют значения весовых коэффициентов  $\tilde{c}_k = (\tilde{c}_k, \tilde{c}_{k+1}, \dots, \tilde{c}_n)$  согласно описанной выше итерационной процедуре. Поскольку значения весовых коэффициентов  $c_k$  параметра  $f_k$  в матрицах  $A^{(k)}$ ,  $k = \overline{1, q}$  определены относительно одного и того же подмножества параметров  $A'_j$ , то это позволяет найти доминирование параметра  $f_i$  над параметром  $f_k (i, k \in A'_j)$  как  $l_{ik} = \tilde{c}_i / \tilde{c}_k$ , а затем все полученные  $l_{ik}$ ,  $i, k \in \{1, \dots, q\}$  использовать для определения весовых коэффициентов  $c_i$  более высоких порядков.

Проиллюстрируем формальную методику определения весовых коэффициентов на примере.

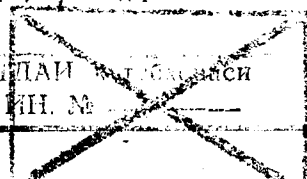
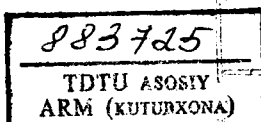
**Пример 1.5.** Выбор наилучшего варианта системы автоматического регулирования. При проектировании системы автоматического регулирования представлено три конкурирующих варианта, эквивалентных по функциональному назначению системы,  $S_1$ ,  $S_2$  и  $S_3$ , параметры которых приведены в табл. 1.1.

Отметим, что все  $\vartheta_i < \vartheta_i^{(0)}$ ,  $i = \overline{1, 3}$  и все параметры целесообразно минимизировать.

Таблица 1.1

Тип системы $S_k$	Параметры			Относительные показатели		
	$\vartheta_1^k$ , с	$\vartheta_2^k$ , Вт	$\vartheta_3^k$ , усл. ед.	$f_1^k$	$f_2^k$	$f_3^k$
$S_1$	0,1	10	15	0,17	0,5	1,0
$S_2$	0,3	13	5	0,5	0,65	0,33
$S_3$	0,6	7	10	1,0	0,35	0,67
Требования технического задания $\vartheta_i^{(0)}$	0,6	20	15	1,0	1,0	1,0

Примечание. Здесь  $\vartheta_1$  — время регулирования;  $\vartheta_2$  — энергопотребление;  $\vartheta_3$  — сложность аппаратурной реализации;  $\vartheta_i^{(0)}$  — директивные значения параметров.



В связи с тем что все требования ТЗ для всех систем выполнены и не требуется применять определенных усилий для достижения заданных директивных значений переменных, оказывается возможным вместо соотношений (1.15) и (1.16), необходимых для определения весовых коэффициентов параметров, использовать выражения вида

$$\Delta f_i^* = |f_i^* - f_i^{(0)}| = |f_i^* - 1|; \Delta f_{ik} = |f_i^* - f_{ik}|.$$

Из табл. 1.1 следует, что  $\min_k \{f_1^{(k)}\} = f_1^* = 0,17$ ;  $\min_k \{f_2^{(k)}\} = f_2^* = 0,35$ ;  
 $\min_k \{f_3^{(k)}\} = f_3^* = 0,33$ .

В табл. 1.2 приведены результаты расчетов величин  $f_{ik}$ ,  $c_i^{(1)}$  и  $c_i$  на 1-й итерации. Величины  $l_{ii}$  взяты равными 0,5.

Т а б л и ц а 1.2

$f_i$	$f_1$	$f_2$	$f_3$	$c_i^{(1)}$	$c_i$
$f_1$	0,5	1,0	0,397	1,897	0,372
$f_2$	0,231	0,5	0,461	1,192	0,234
$f_3$	1,0	0,507	0,5	2,007	0,394

Рассмотрим формирование элементов первой строки табл. 1.2:

$$l_{11} = 0,5; l_{12} = \Delta f_{12} / \Delta f_1^* = |f_1^* - f_{12}| / |f_1^* - 1| = \\ = |0,17 - 1| / |0,17 - 1| = 1,0;$$

$$l_{13} = |0,17 - 0,5| / |0,17 - 1| = 0,397; c_1^{(1)} \sum_{k=1}^3 l_{1k} = 1,897;$$

$$c_1 = c_1' / \sum_{k=1}^3 c_k^{(1)} = 1,897 / 5,096 = 0,372.$$

По формуле  $c_i^{(r+1)} = \sum_{k=1}^n c_i^{(r)} l_{ik}$  рассчитывают весовые коэффициенты более высоких порядков. Результаты расчетов сведены в табл. 1.3. Из таблицы видно, что значения весовых коэффициентов параметров стабилизировались к четвертой итерации.

Согласно выражению (1.13) определим значение аддитивного критерия для всех вариантов систем:

$$F^{(1)}(X) = 0,590; F^{(2)}(X) = 0,466; F^{(3)}(X) = 0,710.$$

Поскольку стремимся минимизировать значение аддитивного критерия, наиболее предпочтительным оказывается вариант  $S_2$ .

Таблица 1.3

№ итерации	$c_1$	$c_2$	$c_3$
1	0,372	0,234	0,394
2	0,349	0,234	0,417
3	0,350	0,238	0,412
4	0,351	0,237	0,412
5	0,351	0,237	0,412

В настоящее время получили распространение интерактивные методы решения многокритериальных задач, когда информация о важности и предпочтениях приходит как от инженера-разработчика, так и от ЭВМ. Уточнение обобщенных критериев и упорядочивание критериев по важности производится на основе диалога конструктора с ЭВМ. Часто для определения наилучшего решения конструктору приходится решать задачи структурной и параметрической оптимизации. При этом модель принятия решения описывается как задача многокритериальной оптимизации. В этом случае используют интерактивный режим оптимизации или диалоговой оптимизации. Разработчик может изменить процесс решения задачи на любом этапе, параметры, метод решения, математическое описание задачи. Проблемными здесь являются разработка эффективных пакетов прикладных программ, сценариев диалога, эвристических и точных алгоритмов проектирования с учетом расплывчатости и неопределенности интеллектуальной деятельности инженера-разработчика.

### 1.5. ВИДЫ ОБЕСПЕЧЕНИЯ И КЛАССИФИКАЦИЯ САПР

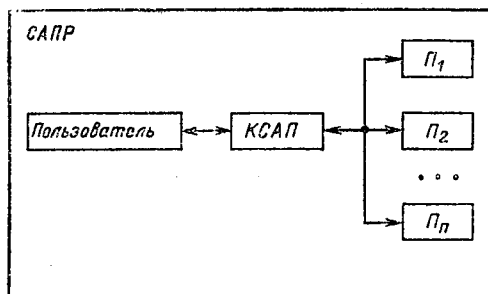
Все определения, приведенные ниже, даны в соответствии с существующими ГОСТ и стандартами по автоматизации проектирования.

*Система автоматизированного проектирования (САПР)* — организационно-техническая система, состоящая из комплекса средств автоматизации проектирования (КСАП), взаимосвязанного с необходимыми подразделениями проектной организации  $P_1, P_2, \dots, P_n$  или коллективом специалистов (пользователей системы) и выполняющая автоматизированное проектирование (рис. 1.5, а). Соответственно *система автоматического проектирования* выполняет автоматическое проектирование без участия человека.

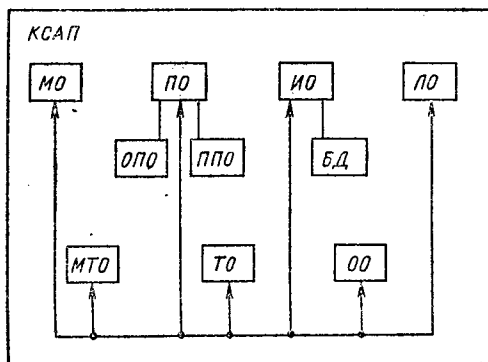
**Виды обеспечения САПР.** Комплекс средств автоматизации проектирования (КСАП) — это совокупность различных видов обеспечения автоматизированного (автоматического) проектирования (АП), необходимых для выполнения АП (рис. 1.5, б).

*Математическое обеспечение* (МО) АП — это совокупность математических методов (ММет), математических моделей (ММ) и алгоритмов проектирования (АлП), необходимых для выполнения АП, представленных в заданной форме (рис. 1.6, а).

*Техническое обеспечение* (ТО) АП — это совокупность взаимосвязанных и взаимодействующих технических средств, предназначенных для выполнения АП. Пример ТО АП ЭВА показан на рис. 1.6, б. Технические средства (ТС)



а)



б)

Рис. 1.5. Система автоматизированного (автоматического) проектирования (а) и комплекс средств автоматизированного (автоматического) проектирования (б)

выполняют определенную функцию в САПР и представляют собой компоненты ТО. Вообще говоря, компонент САПР — это элемент средства обеспечения, выполняющий определенную функцию. К ТС относятся устройства вычислительной и организационной техники, средства передачи данных, измерительные и другие устройства.

Различают следующие группы ТС:

**подготовка и ввод данных.** Группа предназначена для автоматизации подготовки и редактирования данных при вводе в ЭВМ алфавитно-цифровой и графической информации. Группа дает возможность кодирования информации, нанесения данных на машинные носители, ввода данных в ЭВМ, визуального контроля и редактирования данных при вводе информации;

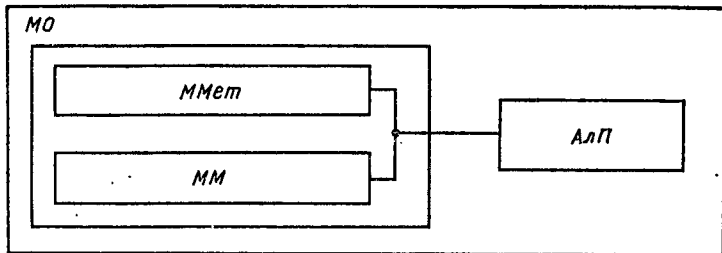
**передача данных.** Группа предназначена для обеспечения дистанционной связи технических средств по телефонным, телеграфным и специальным каналам связи;

**программная обработка данных.** Группа включает в себя универсальные или специализированные ЭВМ, обеспечивающие прием цифровых данных с устройства ввода или каналов связи, их программной обработки, накопления и вывода на машинные носители, устройства отображения и вывода связи. Позволяет изменить производительность путем наращивания ЭВМ, использовать мультипрограммный и диалоговый режимы работы;

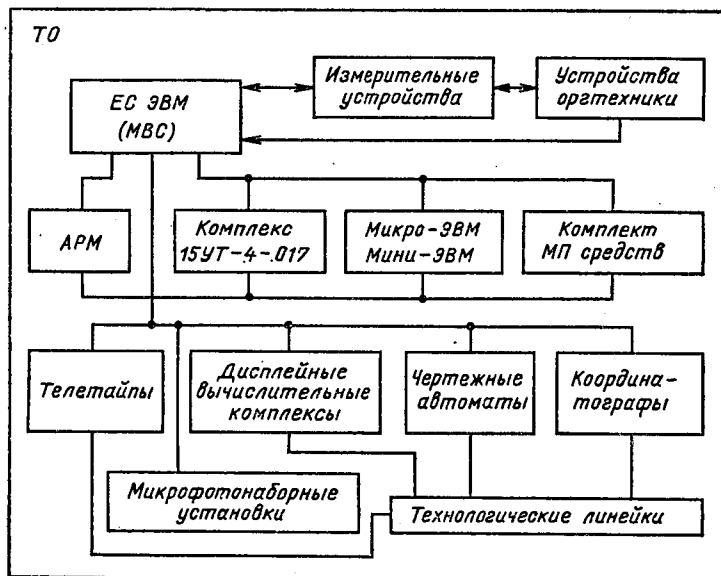
**отображение и документирование данных.** Группа предназначена для оперативного представления и документирования проектных решений. Здесь используют печатающие устройства и графопостроители, микрофильмы, микрофиши и устройства отображения визуальной информации;

**архив проектных решений.** Группа предназначена для обеспечения хранения, контроля, восстановления и размножения данных о проектных решениях и справочных данных.

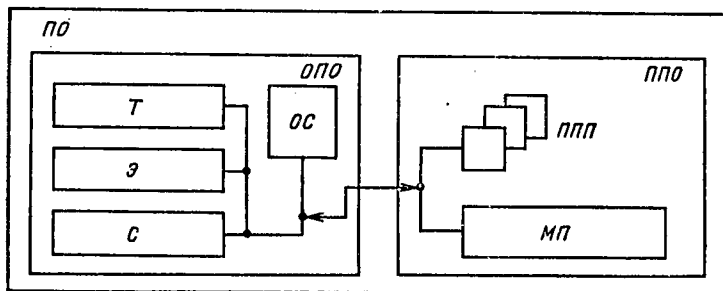
Компоненты ТО создаются на базе серийных средств вычислительной техники общего назначения и специализированных технических средств. В настоящее время преимущественно используют двухуровневую иерархическую структуру комплекса ТС САПР. Структура включает в себя компоненты центрального вычислительного комплекса (ЦВК) и компоненты терминального комплекса (ТК). Центральный ВК строят на основе ЭВМ, вычислительных систем и сетей ЭВМ коллективного пользования. Терми-



а)



б)



в)

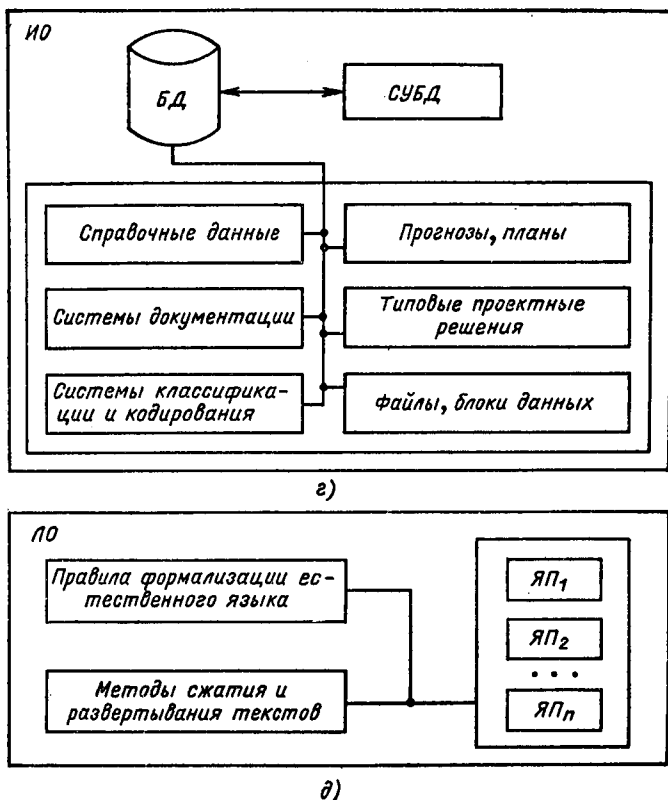


Рис. 1.6. Математическое (а), техническое (б), программное (в), информационное (г) и лингвистическое (д) обеспечения

нальный комплекс САПР строят на основе автоматизированных рабочих мест (АРМ) терминальных станций с использованием микропроцессоров, мини- и микро-ЭВМ.

*Программное обеспечение (ПО) АП* — совокупность машинных программ, необходимых для выполнения АП, представленных в заданной форме. Часть ПО АП, предназначенную для управления проектированием, называют *операционной системой (ОС) АП*.

Совокупность машинных программ (МП), необходимых для выполнения какой-либо проектной процедуры и представленных в заданной форме, называют *пакетом прикладных программ (ППП)*.



Компонентами ПО являются документы с текстами программ, программы на всех видах носителей, эксплуатационные документы. Программное обеспечение разделяют на общесистемное (ОПО) и прикладное (ППО). Компонентами ОПО являются трансляторы (Т) с алгоритмических языков, эмуляторы (Э), супервизоры (С) и др. Компонентами ППО являются программы (МП) и пакеты прикладных программ для АП (рис. 1.6, в).

*Информационное обеспечение (ИО) АП* — совокупность сведений, необходимых для выполнения АП, представленных в заданной форме. Основной частью ИО являются *автоматизированные банки данных*, которые состоят из баз данных (БД) САПР и систем управления базами данных (СУБД). В ИО входят нормативно-справочные документы, задания государственных планов, прогнозы технического развития, типовые проектные решения, системы классификации и кодирования технико-экономической информации, системы документации типа ЕСКД, ЕСТД, файлы и блоки данных на машинных носителях, фонды нормативные, плановые, прогнозные, типовых решений, алгоритмов и программ и т. п. (рис. 1.6, г).

Управление автоматизированным банком данных осуществляют проектировщики, при этом необходимо обеспечить целостность, правильность данных, эффективность и функциональные возможности СУБД. Проектировщик организует и формирует БД, определяет вопросы использования и реорганизации. База данных составляется с учетом характеристик объектов проектирования, процесса проектирования, действующих нормативов и справочных данных. При создании автоматизированных банков данных одним из основных является принцип информационного единства, заключающийся в использовании единой терминологии, условных обозначений, символов, единых проблемно-ориентированных языков, способов представления информации, единой размерности данных физических величин, хранящихся в БД. Автоматизированные банки данных должны обладать гибкостью, надежностью, наглядностью и экономичностью. Гибкость заключается в возможности адаптации, наращивания и изменения средств СУБД и структуры БД. Реорганизация БД не должна приводить к изменению прикладных программ. Для одновременного обслуживания пользователей должен быть организован параллельный доступ к данным. При использовании интерактивных методов проектирования необходимо использовать режим диалога.

Все версии СУБД должны генерироваться в соответствии с имеющимся комплексом ТС САПР. Доступ к информации БД должен обеспечиваться пользователями различных уровней.

Для надежности необходимо иметь возможность восстановления информации и ПО в случае разрушения, обеспечивать стандартные реакции на ошибочный запрос. Для наглядности информация должна представляться в обычной и удобной для пользователя форме. Для обеспечения экономичности необходимо исключить неоправданное дублирование данных, обеспечить автоматизацию сбора статистических данных о содержании и использовании информации банка для организации эффективного распределения памяти, обеспечить наличие средств для тиражирования баз данных.

*Лингвистическое обеспечение (ЛО) АП* — совокупность языков проектирования (ЯП), включая термины и определения, правила формализации естественного языка и методы сжатия и развертывания текстов, необходимых для выполнения АП, представленных в заданной форме (рис. 1.6, д).

*Методическое обеспечение (МО)* — совокупность документов, устанавливающих состав и правила отбора и эксплуатации средств обеспечения АП, необходимых для выполнения АП. Отметим, что в некоторых работах и документах методическое обеспечение понимается более широко: в качестве компонентов включает МО и ЛО.

*Организационное обеспечение (ОО) АП* — совокупность документов, устанавливающих состав проектной организации и ее подразделений, связи между ними, их функции, а также форму представления результата проектирования и порядок рассмотрения проектных документов, необходимых для выполнения АП. Компонентами ОО САПР являются методические и руководящие материалы, положения, инструкции, приказы и другие документы, обеспечивающие взаимодействие подразделений проектной организации при создании и эксплуатации САПР.

В настоящее время при создании ЭВМ пятого поколения с перестраиваемой архитектурой и коммутацией, способных принимать решения в условиях расплывчатости и неопределенности, перспективными являются САПР, позволяющие адаптироваться к внешним условиям проектирования и имеющие возможность настраиваться на заданный класс создаваемых объектов.

*Интегрированной называют САПР, имеющую альтернативное ПО и ОС АП и позволяющую выбирать совокупность машинных программ применительно к заданному объекту или классу объектов проектирования.*

*Функционирование САПР — выполнение проектирования в САПР в соответствии с заданным алгоритмом проектирования. Функционирование САПР должно обеспечивать получение проектных решений, т. е. промежуточных или конечных описаний объекта проектирования, необходимых для его окончания. Результатом проектирования в САПР считают совокупность законченных проектных решений, удовлетворяющих ТЗ и необходимых для создания объекта проектирования.*

*Алгоритм функционирования САПР — совокупность предписаний, необходимых для функционирования САПР.*

*Управление САПР — совокупность воздействий извне, предусмотренных алгоритмом функционирования САПР.*

**Классификация САПР.** В общем смысле классификация — система соподчиненных понятий, часто представляемая в виде различных схем, таблиц и используемая как средство для установления связей между этими понятиями или классами объектов, а также для точной ориентировки в многообразии понятий или соответствующих объектов. Классификация фиксирует место объекта в системе, которая указывает на его свойства. В связи с этим она служит средством хранения и поиска информации, содержащейся в ней самой. Классификация создает условия для разработки технически обоснованных норм обеспечения процесса создания, функционирования и стандартизации в области САПР.

Системы автоматизированного проектирования классифицируются по типу, разновидности и сложности объекта проектирования; уровню и комплектности автоматизации проектирования; характеру и числу выпускаемых проектных документов; числу уровней в структуре технического обеспечения (рис. 1.7). На этом классификация первого уровня закончена. Рассмотрим классификацию второго иерархического уровня.

По типу объекта проектирования различают САПР (рис. 1.8, а): изделий машиностроения и приборостроения; технологических процессов в машиностроении и приборостроении; объектов строительства; организационных систем.

Под *организационной системой* понимают совокупность,

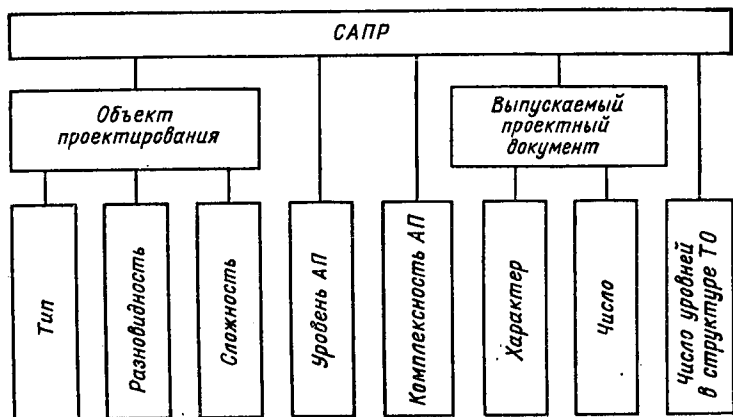


Рис. 1.7. Классификация САПР

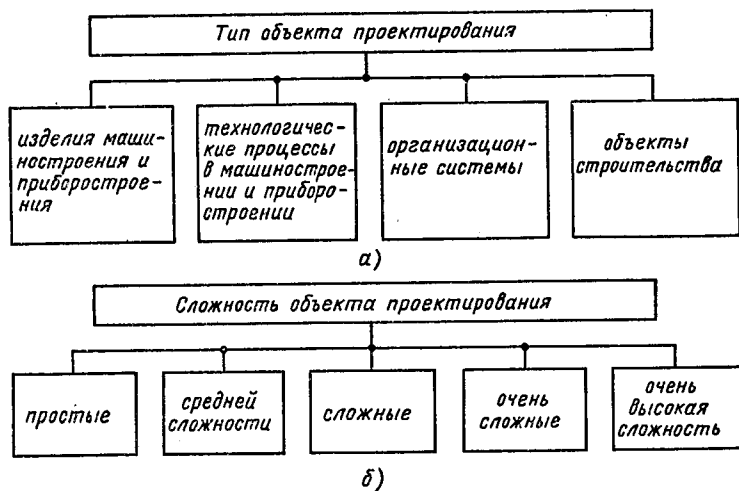


Рис. 1.8. Классификация САПР по типу (а) и сложности (б) объекта проектирования

состоящую из коллектива специалистов и комплекса средств, взаимодействий и взаимосвязей между ними и с внешней средой и алгоритма процесса проектирования, необходимого для выполнения АП.

По разновидности объекта проектирования можно раз-

лчать САПР ЭВА, САПР РЭА, САПР атомной станции, САПР ракетной системы и т. п.

По сложности объекта проектирования различают САПР (рис. 1.8, б): простых объектов, содержащих до  $10^2$  составных частей; объектов средней сложности, содержащих от  $10^2$  до  $10^3$  составных частей; сложных объектов, содержащих от  $10^3$  до  $10^4$  составных частей; очень сложных объектов, содержащих от  $10^4$  до  $10^6$  составных частей; объектов очень высокой сложности, содержащих  $10^6$  и более составных частей.

Следовательно, САПР ТЭЗ современных ЕСЭВМ относится к САПР простых объектов, а САПР многопроцессорных вычислительных систем на сверхбольших интегральных микросхемах — к САПР очень сложных объектов.

По уровню автоматизации проектирования различают САПР (рис. 1.9, а) низкоавтоматизированные, в которых

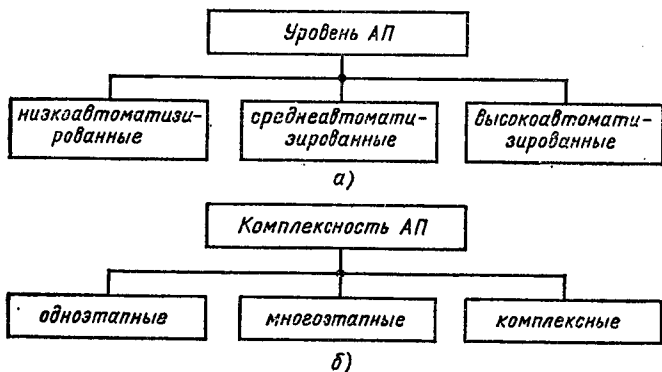


Рис. 1.9. Классификация САПР по уровню (а) и сложности (б) автоматизации проектирования

число автоматизированных проектных процедур (АПП) составляет 25 % общего числа проектных процедур; среднеавтоматизированные, — от 25 до 50 % общего числа проектных процедур, высокоавтоматизированные от 50 до 75 %. В этих системах применяют методы многовариантного оптимального проектирования. Например, в гибких автоматизированных производствах (ГАП) для эффективности результатов необходимо использовать САПР средне- и высокоавтоматизированного проектирования.

По сложности САПР классифицируют так

(рис. 1.9, б): одноэтапные, выполняющие один этап проектирования из всех установленных для объекта; многоэтапные, выполняющие несколько этапов проектирования из всех установленных для объекта; комплексные, выполняющие все этапы проектирования, установленные для объекта.

По характеру выпускаемых проектных документов различают САПР (рис. 1.10, а): текстовые, выполняющие только текстовые, документы на бумажной ленте или листе; текстовые и графические, выполняющие текстовые и графические документы на бумажной ленте или листе; на машинных носителях, выполняющих документы на перфоно-

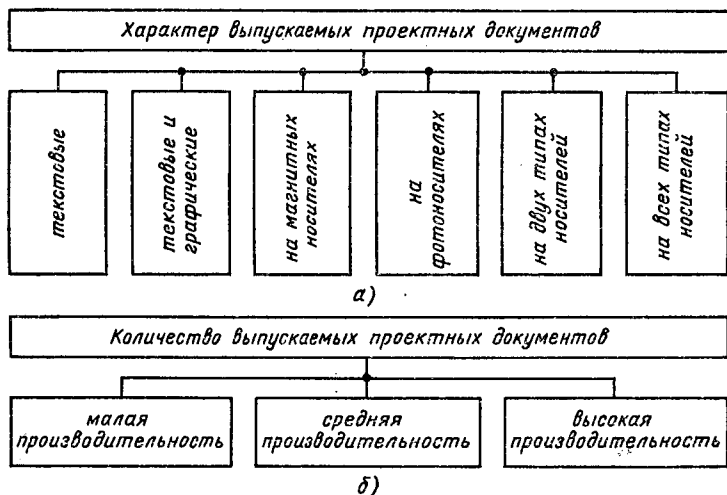


Рис. 1.10. Классификация САПР по характеру (а) и числу (б) выпускаемых проектных документов

носителях (перфокарты, перфоленты) и на магнитных носителях (магнитных лентах, дисках и барабанах); на фотоносителях, выполняющих документы на микрофильмах, микрофишах, фотошаблонах и т. п.; на двух типах носителей; на всех типах носителей.

По количеству выпускаемых проектных документов различают САПР (рис. 1.10, б): малой производительности, выпускающих до  $10^5$  проектных документов в пересчете на формат 2 за год; средней производительности, выпускающих от  $10^5$  до  $10^6$  проектных документов; высокой произво-

длительности, выпускающих свыше  $10^6$  проектных документов.

В различных отраслях вводятся различные количественные характеристики информации, определяющие производительность САПР.

По числу уровней в структуре технического обеспечения различают САПР (рис. 1.11): одноуровневые, построенные

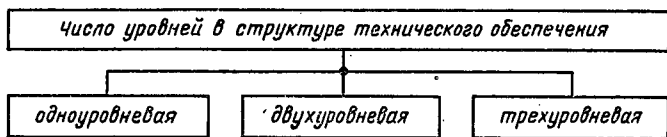


Рис. 1.11. Классификация САПР по числу уровней в структуре технического обеспечения

на основе ЭВМ среднего или высокого класса со штатным набором периферийных устройств, который может быть дополнен средствами обработки графической информации; двухуровневые, построенные на основе ЭВМ среднего или высокого класса и одного или нескольких АРМ, включающих мини-ЭВМ; трехуровневые, построенные на основе ЭВМ высокого класса, АРМ и периферийного программно-управляемого оборудования.

Данная классификация — не догма. Теория автоматизации проектирования непрерывно развивается. Появляются новые технические и программные средства ЭВМ, комплексные САПР, поэтому существующие схемы классификации САПР будут видоизменяться и совершенствоваться.

#### 1.6. СИСТЕМНЫЙ ПОДХОД К АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ И ПРИНЦИПЫ ОРГАНИЗАЦИИ САПР

Рассмотрим основные принципы организации САПР. Целью создания САПР является повышение качества и технико-экономического уровня проектируемых объектов при их создании и применении, повышение производительности труда, сокращение сроков, уменьшение стоимости и трудоемкости проектирования.

Очевидно, что создание САПР в первую очередь должно быть обусловлено технико-экономической целесообразностью.

**Состав типовой САПР.** Составными структурными частями САПР являются подсистемы, обладающие всеми свойствами систем и создаваемые как самостоятельные. *Подсистемой* САПР называют выделенную по некоторым признакам часть САПР, обеспечивающую получение законченных проектных решений.

По назначению подсистемы САПР разделяют на проектирующие и обслуживающие. К проектирующим относят подсистемы, выполняющие проектные процедуры и операции, например подсистема логического проектирования, подсистема конструкторского проектирования, подсистема технологического проектирования, подсистема проектирования деталей и сборочных единиц и т. п. К обслуживающим относят подсистемы, предназначенные для поддержания работоспособности проектирующих подсистем, например подсистема информационного поиска, подсистема документирования, подсистема графического отображения объектов проектирования и т. п.

По отношению к объекту проектирования различают объектно-ориентированные (объектные) и объектно-независимые (инвариантные). К объектным относят подсистемы, выполняющие одну или несколько проектных процедур или операций, непосредственно зависимых от конкретного объекта проектирования. К инвариантным относят подсистемы, выполняющие унифицированные проектные процедуры и операции, например функции обработки, не зависящие от особенностей проектируемого объекта.

Подсистемы состоят из компонентов, объединенных общей для данной подсистемы целевой функцией и обеспечивающих функционирование этой подсистемы.

Структурное единство подсистемы обеспечивается связями между компонентами различных обеспечений САПР.

**Принципы создания САПР.** К ним относят принципы включения, системного единства, развития, комплексности, информационного единства, совместимости, стандартизации.

Принцип включения обеспечивает разработку САПР на основе требований, позволяющих включать эту САПР в САПР более высокого уровня.

Принцип системного единства состоит в том, что при создании, функционировании и развитии САПР связи между подсистемами должны обеспечивать целостность системы.

Согласно принципу развития САПР должна создавать-



ся и функционировать с учетом пополнения, совершенствования и обновления подсистем и компонентов.

Принцип комплексности обеспечивает связность проектирования элементов и всего объекта на всех стадиях, позволяет осуществлять согласование и контроль характеристик элементов и объекта в целом.

Принцип информационного единства состоит в использовании в подсистемах, компонентах и средствах обеспечения САПР единых условных обозначений, терминов, символов, проблемно-ориентированных языков, способов представления информации, соответствующих принятым нормативным документам.

Согласно принципу совместимости языки, символы, коды, информационные и технические характеристики структурных связей между подсистемами, средствами обеспечения и компонентами должны обеспечивать совместное функционирование подсистем и сохранять открытую структуру системы в целом.

Принцип стандартизации заключается в проведении унификации, типизации и стандартизации подсистем и компонентов, инвариантных к проектируемым объектам и отраслевой специфике, а также в установлении правил с целью упорядочения деятельности в области создания и развития САПР.

Подсистемы должны вводиться в действие и функционировать независимо от других подсистем. Единство общесистемных требований обеспечивает проектная служба САПР.

**Стадии создания САПР.** Различают внешнее и внутреннее проектирование (рис. 1.12). К внешнему проектированию относятся следующие стадии:

*предпроектные исследования* — производится обследование проектной организации, оформление технического отчета, а также его согласование и утверждение;

разработка, согласование и утверждение *технического задания* — выполняются совместно разработчиком и заказчиком.

К внутреннему проектированию относятся следующие стадии:

разработка *технического предложения* — выбирают и обосновывают оптимальный вариант САПР. Производится разработка комплекта документации, а также согласование и утверждение технического предложения;

*эскизный проект* — разрабатывают принципиальные ре-

шения по созданию САПР и документации, согласовывают и утверждают эскизный проект;

*технический проект* — разрабатывают окончательные решения по созданию САПР и документации, согласовывают и утверждают технический проект;

*рабочий проект* — разрабатывают рабочую документацию по САПР, согласовывают и утверждают рабочий проект;

*изготовление, отладка и испытания* — изготавливают и отлаживают компоненты САПР, производят монтаж, налад-

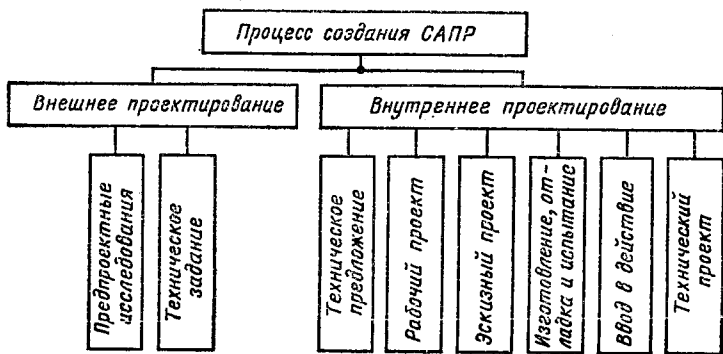


Рис. 1.12. Стадии создания САПР

ку и испытание КСАП, осуществляют подготовку организации к вводу системы в действие;

*ввод в действие* — производят опытное функционирование и приемочные испытания САПР.

Работы по подготовке организации к вводу в действие САПР, подсистем и компонентов должны производиться на всех стадиях ее создания. Перечисленные стадии САПР не всегда являются обязательными. Иногда они отсутствуют или объединяются в одну или несколько стадий в зависимости от конкретных требований на проектируемую САПР.

Рассмотрим более подробно основные стадии создания САПР. К работам на стадии предпроектных исследований (ПИ) относят обследование проектной организации, оценку возможности создания САПР, сбор данных, описание существующих САПР и их анализ, сбор предложений по созданию САПР, составу подсистем, разработке компонентов САПР, формированию технических требований к функциям и структуре САПР, видов обеспечения, общесистем-

ных принципов создания САПР. Общее руководство ПИ осуществляет главный конструктор САПР. По окончании ПИ оформляется и утверждается научно-технический отчет.

Техническое задание (ТЗ) является исходным и обязательным документом для создания, приемки или сдачи системы и должно содержать все исходные данные и требования, необходимые для создания САПР. Техническое задание на создание САПР разрабатывает организация — разработчик системы с учетом выполнения предпроектных исследований на основе требований заказчика. Оно согласовывается с организацией — пользователем САПР и по необходимости с другими заинтересованными организациями. Техническое задание на создание САПР утверждает заказчик. Изменяют и уточняют утвержденное ТЗ в процессе создания подсистемы с помощью дополнений в порядке, установленном в ТЗ. Полное ТЗ на создание САПР ЭВА должно содержать следующие разделы:

наименование и область применения — указаны полное или условное наименование (индекс) САПР, краткая характеристика области ее применения;

основание для создания — указаны полное наименование директивных документов, на основании которых создают САПР, организация, утвердившая эти документы, и дата утверждения, а также наименование или условное обозначение темы, в рамках которой создается САПР;

характеристика объектов проектирования — даны сведения о назначении, составе, условиях применения разрабатываемого объекта;

цель и назначение — указаны цель создания САПР, ее назначение и критерий эффективности ее функционирования;

характеристика процесса проектирования — приведены общее описание процесса проектирования, требования к входным и выходным данным, а также требования по разделению проектных процедур, выполняемых при автоматизированном и неавтоматизированном проектировании;

требования к САПР — приведены требования к САПР в целом, к составу ее подсистем, к включению в САПР ранее созданных подсистем и компонентов, требования к взаимосвязи САПР с другими системами (например, с АСУ, АСУТП), видам обеспечения и возможности развития САПР;

техничко-экономические показатели — оценены затраты

на создание САПР, источники экономии, ожидаемая эффективность от применения САПР, требования к технико-экономическим показателям объекта, которые будут достигнуты в результате функционирования САПР;

стадии и этапы — установлены необходимые стадии создания, очередность ввода в действие САПР, этапы по стадиям, сроки выполнения работ и исполнители, перечень документации, предъявляемой по окончании стадии, объемы работ в соответствии с действующими нормативно-техническими и методическими документами;

порядок испытаний и ввода в действие — определены требования к проведению испытаний и вводу в действие САПР и их подсистем;

источники разработки — даны перечень научно-исследовательских, опытно-конструкторских и экспериментальных работ, нормативных документов, методических материалов и т. п., используемых при создании САПР;

приложения — включают таблицы, схемы, расчеты, библиографию, листинги программ, обоснования, доказательства и т. п.

Техническое задание на создание САПР оформляют в соответствии с общими требованиями к текстовым документам.

Целью технического предложения (ТП) является выбор рациональных вариантов САПР, учитывающих требования ТЗ. При разработке ТП выполняются следующие работы: анализ процессов автоматизированного проектирования; выявление возможных вариантов структуры для САПР на основе разработки модулей, реализующих процессы проектирования; выбор рациональных вариантов (варианта) структуры САПР; технико-экономическое обоснование выбранного варианта; составление дополнительных по сравнению с ТЗ требований к САПР; уточнение требований к содержанию работ на последующих стадиях САПР.

Техническое предложение разрабатывает организация, являющаяся главным разработчиком системы, его согласовывают с заказчиком и со всеми заинтересованными организациями. Техническое предложение на создание САПР должно содержать следующие разделы:

общие положения — цель разработки ТП, наименование и дата утверждения ТЗ; назначение и область применения — назначение, характеристика, область и условия применения создаваемой САПР; описание объекта проектирования в САПР — основные составные элементы объектов

проектирования, их взаимосвязь, схема деления; описание процесса автоматизированного проектирования — результаты анализа процесса АП; характеристика и анализ вариантов структуры САПР — результаты синтеза и анализа структуры САПР с выделением подсистем и компонентов и связей между ними, предложения по использованию существующих подсистем и компонентов САПР;

обоснование и описание выбранного варианта — по результатам рассмотрения вариантов структуры САПР обосновывается и описывается выбранный вариант САПР с указанием взаимодействия подсистем и компонентов и выполняемых ими функций;

технико-экономическое обоснование — основные технико-экономические показатели создаваемой САПР;

предложения по содержанию и организации работ на последующих стадиях — уточненные по сравнению с ТЗ данные по очередности и содержанию работ.

Техническое предложение может иметь по необходимости дополнительные разделы и приложения. В последних приводят схемы, описания, расчеты, необходимые для качественного обоснования выбранного варианта САПР.

На стадии эскизного проекта (ЭП) выполняют:

принятие основных решений по взаимодействию САПР с другими системами;

принятие основных технических решений по структуре подсистем САПР, разработке вопросов функционирования подсистем на уровне компонентов;

описание исходных данных и требований на разработку языков проектирования, алгоритмов, компонентов ИО по подсистемам;

оформление и утверждение ЭП.

На стадии технического проекта выполняют: принятие решений по новому процессу проектирования с обеспечением взаимодействия и совместимости автоматических и автоматизированных процедур, получение окончательной схемы функционирования САПР в целом; разработку структуры и состава подсистем САПР; получение окончательной структуры всех видов обеспечений САПР; выбор математических моделей объекта проектирования и его элементов; разработку алгоритмов проектных операций; разработку требований на создание программ реализации процедур проектирования; разработку алгоритмов, языков проектирования, компонентов ИО, формирование общесистемного программного обеспечения; расчет производительности и

выбор режимов работы средств технического обеспечения; определение требований к представлению исходных данных, результатов проектирования и проектных документов; оформление и утверждение совокупности документов, составляющих технический проект.

Технический проект включает в себя следующие основные документы: ведомость; пояснительную записку; схемы процессов проектирования, подсистем и средств обеспечения, спецификации компонентов всех видов обеспечений; смету затрат на создание САПР; ТЗ на разработку исполнителями по отдельным компонентам; расчет ожидаемых технико-экономических показателей.

Рабочий проект (РП) создается на основании технического проекта организацией — разработчиком САПР.

На стадии РП проводят: разработку детальной структуры САПР, ее подсистем, взаимосвязи с другими системами и ее уточнение; построение алгоритмов и структурных схем автоматизированных процессов проектирования; формирование МО, ПО, ИО, ОО; разработку документации для монтажа, настройки и эксплуатации КСАП; создание проектов программ и методик испытаний и опытной эксплуатации; оформление и утверждение.

Результатом работ на этой стадии являются документ «Рабочий проект» и комплект эксплуатационных документов.

Рабочий проект содержит:

ведомость РП; пояснительную записку; спецификацию — перечислены подсистемы, спецификации видов обеспечений; документацию технического обеспечения — приведены спецификация, комплект конструкторских документов;

документацию ИО — включены спецификация, описание базы данных, инструкция по ее заполнению, инструкция по введению массивов данных;

документацию ПО — приведены спецификация, тексты программ, описание программ, порядок и методика испытаний;

документацию МО, МТО, ЛО — содержит спецификацию, пояснительную записку, описание языка, подсистемы управления САПР;

документацию ОО — даны спецификация, программа подготовки специалистов — пользователей САПР, положение о службе САПР;

программу и методику испытаний КСАП — содержит технические данные, подлежащие проверке при испытании

компонентов САПР, порядок испытаний и методы их контроля;

программу и методику опытного функционирования системы и подсистем — приведены данные, обеспечивающие получение и проверку проектных решений, выявление причин сбоев, показателей качества функционирования системы и подсистем;

комплект эксплуатационных документов — содержит МТО, ТО, ПО и ведомость эксплуатационных документов.

Изготовление, отладку и испытания САПР проводят на основании документации РП. Они должны обеспечивать бесперебойную работу подсистем действующей САПР.

На стадии изготовления, отладки и испытаний выполняют: подготовку организации к вводу в действие САПР, включая обучение пользователей; изготовление и отладку компонентов ТО, ПО и ИО на машинных носителях; монтаж, отладку и испытания КСАП для САПР и всех ее подсистем; организацию работ по оформлению и утверждению актов сдачи компонентов и КСАП.

Работы на стадии ввода в действие САПР предназначены для обеспечения возможности промышленного функционирования САПР и определения технико-экономических показателей системы. На этой стадии проводят опытное функционирование, приемочные испытания, корректировку, доработку и выпуск эксплуатационных документов, промышленную эксплуатацию и развитие САПР.

Опытное функционирование САПР предполагает проверку работоспособности и взаимодействия подразделений организации с подсистемами САПР, выявление неисправностей и отказов, определение фактических технико-экономических показателей системы и корректировку документации, устранение дефектов, оформление протокола.

Приемочные испытания включают в себя оценку качества разработки КСАП, проверку готовности организации к промышленному функционированию САПР, оценку качества опытного функционирования, проверку документации на САПР. Работа оканчивается вводом в действие системы, оформлением и утверждением протокола опытного функционирования и акта приемки САПР в промышленное функционирование.

Промышленную эксплуатацию выполняет служба САПР. Проектная организация для определения соответствия научно-техническому уровню проводит периодические испытания САПР. При этом оформляют заключение о соответ-

ствии подсистем и компонентов действующей САПР современным требованиям, определяют оценку эффективности и качества продукции, полученной с помощью САПР, записывают рекомендации о разработке или модернизации подсистем и компонентов проверяемой САПР.

Развитие САПР ведут путем модернизации подсистем, компонентов и средств обеспечения, а также ввода в эксплуатацию новых подсистем и компонентов.

Рассмотрим основные функции и рекомендации по структуре служб САПР в организациях. Основными функциями службы САПР являются:

- изучение, анализ и обобщение отечественного и зарубежного опыта создания и функционирования САПР;

- исследование и разработка системных вопросов создания САПР в отрасли, организация создания отраслевой системы, разработка отраслевых инвариантных подсистем и компонентов САПР для отрасли;

- анализ ввода в действие и опытного функционирования САПР, обеспечение разработки научно-технических прогнозов развития САПР в отрасли;

- формирование и контроль исполнения отраслевых координационных (перспективных и годовых) планов НИР и ОКР по созданию САПР в отрасли;

- организация внедрения типовых систем и компонентов САПР;

- координация и контроль за ходом создания САПР в отрасли;

- обеспечение проведения экспертизы и согласование технической документации на создаваемые САПР;

- участие в работе комиссии по вводу в действие САПР в организациях отрасли, участие в создании конкретных систем САПР;

- обеспечение создания и развития отраслевого фонда компонентов САПР;

- разработка отраслевых стандартов, руководящих документов в области создания САПР, функционирования и развития САПР;

- организация отраслевых мероприятий по обмену опытом и технической информации в области САПР;

- разработка предложений по вопросам подготовки кадров по САПР.

Структура службы САПР в головной организации отрасли определяется положением о службе САПР, утверждаемым руководством ведомства. В состав службы САПР



в общем случае входят подразделения: управления созданием и развитием САПР; общесистемных разработок в области МТО, МО, ПО, ЛО, ИО и ТО САПР общепромышленного применения; отраслевого фонда алгоритмов и программ (ОФАП) САПР; информации с архивом технической документации; разработки типовых подсистем и компонентов САПР общепромышленного назначения; технической эксплуатации отдельных видов средств обеспечения.

Порядок взаимоотношений службы САПР с другими подразделениями определяет положение о службе САПР, Руководство деятельностью службы САПР осуществляет главный конструктор по САПР в отрасли.

Основные функции службы САПР в организациях следующие: проведение предпроектных обследований в организации; согласование ТЗ на создание САПР; выделение в подразделениях — пользователях САПР группы специалистов, обеспечивающих функционирование и развитие подсистем и компонентов САПР в соответствии со специализацией подразделений; обеспечение создания САПР; реализация привязки типовых подсистем и компонентов САПР к условиям организаций; участие в разработке индивидуальных компонентов САПР с учетом специфики конкретного объекта проектирования; организация обучения и работы специалистов-пользователей с КСАП; обеспечение проведения приемочных испытаний САПР; организация изготовления комплекса нестандартных компонентов САПР; организация работ по реконструкции помещений; осуществление подготовки и перевод на машинные носители необходимой информации для заполнения базы данных проектирования; организация монтажных и наладочных работ, необходимых для функционирования САПР; осуществление планомерного развития САПР; обеспечение использования типовых подсистем и компонентов САПР.

В состав службы САПР в общем случае входят следующие подразделения: организационно-методического обеспечения; разработки и развития отдельных компонентов САПР с учетом спецификации предприятия; технического обеспечения и эксплуатации средств ТО; ведение архива технической документации САПР, средств ИО, ПО на магнитных носителях, а также организации и диспетчеризации работ.

Руководство деятельностью службы САПР осуществляет главный конструктор САПР в организации. Служба САПР обеспечивает промышленную эксплуатацию САПР.

Развитие САПР ведут путем ввода новых подсистем и компонентов в эксплуатацию, а также на основе модернизации существующих подсистем и компонентов.

Мониторная система САПР, т. е. подсистема управления САПР, является обслуживающей подсистемой САПР и предназначена для организации и оптимизации управления процессом при выполнении проектных процедур и взаимодействия подсистем САПР. Мониторная система (МС) в общем случае включает в себя компоненты математического (МО), программного (ПО) и информационного (ИО) обеспечения (рис. 1.13).

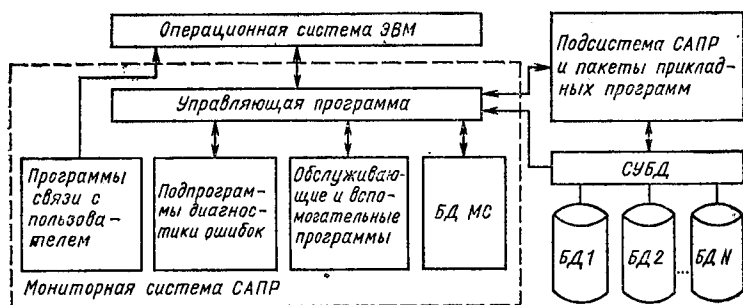


Рис. 1.13. Вариант организации мониторинг системы САПР

В состав МС САПР входят процедурно-ориентированные языки (метаязыки) проектирования и управления процессом проектирования, языки описаний функций МС, т. е. распределения ресурсов САПР, формирования плановых заданий и другое, языки описания структуры данных и преобразования базы данных, программы, обеспечивающие реализацию функций МС, базы данных МС.

Основные функции МС САПР: управление процессом реализации проектных процедур и операций; организация воздействия подсистем САПР; интерпретация языковых форм заданий на выполнение проектных процедур и операций; распределение ресурсов САПР в процессе проектирования; защита ресурсов системы и баз данных САПР от неразрешенного доступа; обеспечение диалоговых и интерактивных режимов работы при проектировании в условиях параллельной работы нескольких подсистем САПР.

Программное обеспечение МС САПР должно функционировать под управлением операционной системы (ОС)

ЭВМ. Мониторная система должна обеспечивать генерацию или настройку, а также контроль и восстановление процесса функционирования программ, обладать свойствами информационной совместимости с ППП и подсистемами САПР на уровне управляющих параметров, констант и интерфейсов обмена информации.

**Диалоговые средства.** Рассмотрим общие требования к диалоговым средствам (ДС), которые приобретают все больший вес в современных САПР. К *диалоговым средствам* САПР относят средства, обеспечивающие прямое (интерактивное) взаимодействие пользователя с КСАП, осуществляемое в реальном масштабе времени.

*Диалоговый режим* — это проектирование с применением ДС, при котором пользователь, взаимодействуя с КСАП, выдает инструкции управления проектированием. Диалоговые средства составляют часть КСАП и обеспечивают перевод САПР из пакетного режима в диалоговый. *Пакетный режим* — это проектирование при отсутствии ДС и непосредственного воздействия пользователя на процесс проектирования. В зависимости от вида представления данных ДС подразделяют на символьные, графические и смешанные. К символьным относят ДС, обеспечивающие представление данных в символьном (алфавитно-цифровом) виде, к графическим — ДС в графическом виде, а к смешанным — ДС как в графическом, так и в символьном виде. В зависимости от режима применения ДС САПР подразделяют на средства автономного, неавтономного и смешанного применения.

В состав ДС САПР входят: МО и ЛО, содержащие инструкции и методики ведения диалогового режима, язык диалогового взаимодействия и язык внутреннего представления данных;

ТО, в которое входят устройства ввода/вывода данных на электронных и других типах дисплеев, процессоры ЭВМ, устройства обмена данными;

ПО, имеющие программы диалогового ввода и вывода данных, программы интерпретатора языка диалогового взаимодействия, программы управления процессом диалога, программы формирования результатов проектирования, программы вывода результатов на дисплей и их редактирование, программы взаимодействия с другими обслуживающими подсистемами САПР, программы регистрации процесса диалога и выдачи пользователю справок, инструкций, сообщений об ошибках;

ИО, в которое входят БД для процесса диалога, БД для регистрации и корректировки результатов проектирования, библиотека типовых графических элементов, технических данных и параметров типовых элементов проектирования.

Рекомендуется предусматривать в составе ДС средства обучения и самообучения пользователя, а также адаптацию ДС по отношению к возможностям пользователя.

Математическое обеспечение ДС САПР должно содержать оригинальные и типовые методы проектирования. Лингвистическое обеспечение ДС САПР базируется на естественных языках, общепринятых символьных и графических образах языков, оно должно быть инвариантным по отношению к ИО ДС САПР.

Технические ДС САПР обеспечивают ввод/вывод данных с использованием символьных и графических образов естественных языков. Технические ДС строят на основе серийно изготавливаемого периферийного оборудования ЭВМ, обеспечивающего ввод/вывод данных.

Программное обеспечение ДС САПР создают с использованием базового ПО средств вычислительной техники. Программное обеспечение ДС подразделяют на общее и специализированное.

Общее ПО ДС обеспечивает: ввод и вывод информации на диалоговый терминал; выдачу пользователю справок, инструкций и сообщений об ошибках; регистрацию процесса диалога и управление им; адаптацию структуры диалога с учетом особенностей конкретного пользователя; управление базами данных и их защиту; редактирование и манипулирование данными; формально-логический контроль диалога; восстановление процесса диалога.

Специализированное ПО ДС обеспечивает: трансляцию или интерпретацию входных сообщений; формирование информации пользователю о результатах проектирования; лексический контроль входных сообщений.

Программное обеспечение ДС должно иметь модульную структуру и адаптироваться к изменению структуры и средств ведения диалога. Информационное обеспечение ДС САПР должно быть совместимым с ИО недиалоговых средств САПР. В него должны входить БД для управления процессом диалога и обеспечения режимов обучения и адаптации.

**Принципы системного подхода.** Рассмотрим вопросы системного подхода при создании ЭВМ. Считается, что

исследование объектов проектирования с помощью их математических моделей составляют основную суть системного подхода. Выделяют такие принципы системного подхода как:

иерархичность — каждая подсистема или элемент может рассматриваться как система;

структурность — возможность описания системы с помощью коммутационных связей между ее элементами;

взаимозависимость — проявление свойств системы только при взаимодействии с окружающей средой;

множественность описания — описание системы на основе множества математических моделей;

целостность изучаемой системы — изучение свойств целой системы на основе анализа и знаний частей этого целого.

В основе системного подхода лежит исследование объекта как системы, направленное на поиск механизмов целостности объектов и выявление всех его связей. Системный подход обосновывает общую оптимизацию разработки, проектирования, конструирования, производства, эксплуатации объекта. Одна из важнейших задач системного подхода — выбор вида, числа, уровня сложности, формы представления математических моделей. В общем случае системный подход при проектировании — это учет всех факторов, которые влияют на процесс создания объекта. Другими словами, системный подход — это решение технической задачи для части с учетом целого.

Например, задача проектирования ЭВМ не может в настоящее время рассматриваться как локальная, она должна решаться с учетом всех системных критериев, а также с учетом состояния развития вычислительной техники. В качестве системных критериев при проектировании ЭВМ используют такие показатели, как сложность, живучесть, надежность, стоимость, эффективность, производительность и др. Использование системных критериев позволяет осуществлять комплексную оптимизацию при проектировании ЭВМ.

Известные принципы системного подхода в применении к проектированию ЭВА можно сформулировать следующим образом:

ЭВА, состоящая из проектируемых отдельно оптимальным образом частей, не является в общем случае оптимальной (значение оптимума может быть только относительным), поэтому должна оптимизироваться в целом как единый объект с заданным целевым назначением;

ЭВА должна оптимизироваться по критерию, отражающему цель оптимизации;

ЭВА оптимизируется в условиях количественно определенных ограничений на оптимизируемые параметры.

Выше были описаны задачи синтеза. Задачи анализа при проектировании являются задачами исследования моделей создаваемых объектов. Выделяют физические (макеты, стенды, блоки и т. п.) и математические модели. Математические модели (ММ) — это совокупность математических объектов с заданными отношениями между ними. Математические модели бывают функциональные, структурные и коммутационные. Функциональные ММ отображают физические и информационные процессы, происходящие в моделируемом объекте; структурные ММ — геометрические свойства объектов; коммутационные ММ — соединения в моделируемых объектах. При проектировании объекта обычно используют совокупность описанных моделей. На каждом этапе проектирования могут применять различные модификации ММ.

Величины, фигурирующие в ММ объектов, называют параметрами. Различают параметры внешние, внутренние и выходные. Внешние параметры характеризуют свойства внешней по отношению к проектируемому объекту среды, внутренние параметры — свойства элементов объекта, а выходные параметры — свойства самого объекта.

### Контрольные вопросы и задания

1. Опишите основные способы проектирования объектов.
2. Определите понятие «проектирование».
3. Охарактеризуйте содержание задач на различных иерархических уровнях проектирования ЭВМ.
4. Поясните различия между понятиями «стадия», «этап», «процедура проектирования».
5. Поясните сущность блочно-иерархического подхода, приведите примеры декомпозиции описаний сложных технических систем.
6. Каким образом задача векторной оптимизации сводится к задаче скалярной оптимизации?
7. Что понимается под внешними, внутренними и выходными параметрами?
8. Каким образом задачи принятия решений используются в САПР? Что такое задача выбора?
9. Дайте определения критерия, однокритериальных и многокритериальных задач.

10. Опишите методы решения многокритериальных задач, приведите примеры.

11. Дайте определение САПР и КСАП.

12. Охарактеризуйте математическое, техническое, информационное, программное, лингвистическое, методическое и организационное обеспечения САПР.

13. Дайте определение интегрированной САПР, приведите примеры.

14. Приведите основные виды классификации САПР и поясните, какие цели она преследует.

15. Опишите основные принципы создания САПР.

16. Перечислите и охарактеризуйте основные стадии создания САПР.

17. Дайте характеристику предпроектных исследований при создании САПР.

18. Опишите порядок работ при создании ТЗ на САПР.

19. Что такое техническое предложение?

20. Опишите основные требования при создании эскизного проекта на САПР.

21. Охарактеризуйте порядок работ при разработке технического проекта САПР.

22. Что такое рабочий проект на САПР?

23. Дайте характеристику стадий изготовления, отладки, испытания и ввода в действие САПР.

24. Какие цели преследует разработка проектирующих и обслуживающих подсистем САПР?

25. Что такое мониторинговая система САПР, каковы ее состав и функции?

26. Приведите основные рекомендации по созданию служб САПР в организациях.

27. Опишите основные принципы разработки автоматизированных банков данных.

28. Приведите определение и основные сведения о диалоговом режиме и диалоговых средствах САПР.

29. Опишите принципы восходящего проектирования. Приведите примеры.

30. Что такое нисходящее проектирование?

31. Приведите основные принципы системного подхода при разработке САПР.

## ГЛАВА 2

### ТИПОВЫЕ СТРУКТУРЫ САПР

#### 2.1. ОРГАНИЗАЦИЯ ТЕХНИЧЕСКИХ СРЕДСТВ САПР

Техническое обеспечение САПР совместно с ПО является инструментальной базой САПР, в среде которой реализуются другие виды обеспечения САПР.

Компоненты технического обеспечения САПР включают в себя средства вычислительной и организационной техники, средства передачи данных, измерительные и другие устройства или их сочетания, обеспечивающие функционирование САПР. Совокупность компонентов ТО образует комплекс технических средств САПР.

Требования к составу и структуре ТС формируются исходя из: общих требований к структуре САПР; эффективного решения выделенного класса задач проектирования, активного включения пользователя в процесс проектирования; возможности работы с графическим материалом, включая процессы как ввода и обработки, так и вывода информации.

Классификация ТС САПР приведена в гл. 1. Различают ТС подготовки и ввода данных; передачи данных; программы обработки данных; архива проектных решений.

Логическая структура технических средств САПР приведена на рис. 2.1. Технические средства САПР объе-

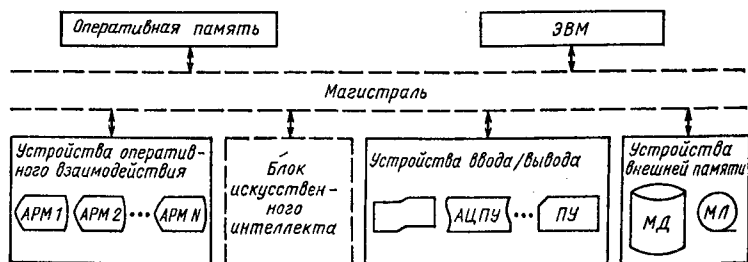


Рис. 2.1. Логическая структура технических средств САПР

динены в группы взаимодействующего оборудования. Среди этих групп выделяют группы базовой конфигурации. *Базовая конфигурация* — это минимальный состав, позволяющий решать задачи определенного класса.



Существует ряд требований, которые необходимо учитывать при разработке базовых конфигураций: унификация проектных решений; построение развивающейся системы, предусматривающее наращивание и совершенствование компонентов технических средств; физическая совместимость, предусматривающая совместное функционирование всех компонентов комплекса; модульность конфигурации, требующая, чтобы компоненты системы были универсальными и типовыми; минимизация стоимости; согласованность основных параметров компонентов системы.

Одним из основных компонентов базовых конфигураций является автоматизированное рабочее место (АРМ).

В настоящее время серийно выпускаются комплексы АРМ-Р, АРМ-М, 15УТ-4-017 и др.

Комплекс АРМ-Р представляет собой совокупность технических и программных средств для организации проектирования радиоаппаратуры, а также для выполнения различных работ, связанных с вводом/выводом, редактированием графической и текстовой информации. Центральным элементом в комплексе является мини-ЭВМ СМ-4 (рис. 2.2), процессор которой связан с общей шиной с пери-

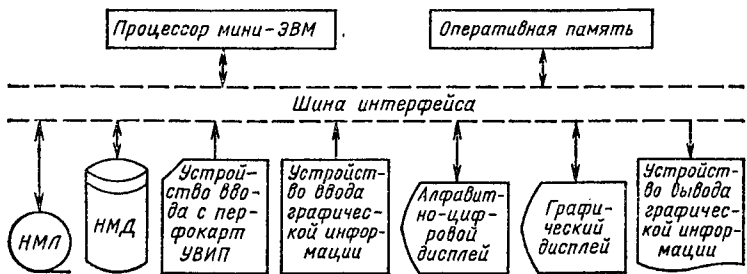


Рис. 2.2. Схема АРМ

ферийным оборудованием и оперативной памятью. Общесистемное программное обеспечение АРМ-Р в мультипрограммном режиме поддерживает различные конфигурации аппаратных средств комплекса, ввод/вывод графических данных, подготовку информации для дальнейшей передачи.

Для приближения пользователя к средствам САПР созданы диалоговые комплексы АРМ-2. Разработано несколько модификаций, где для программной обработки

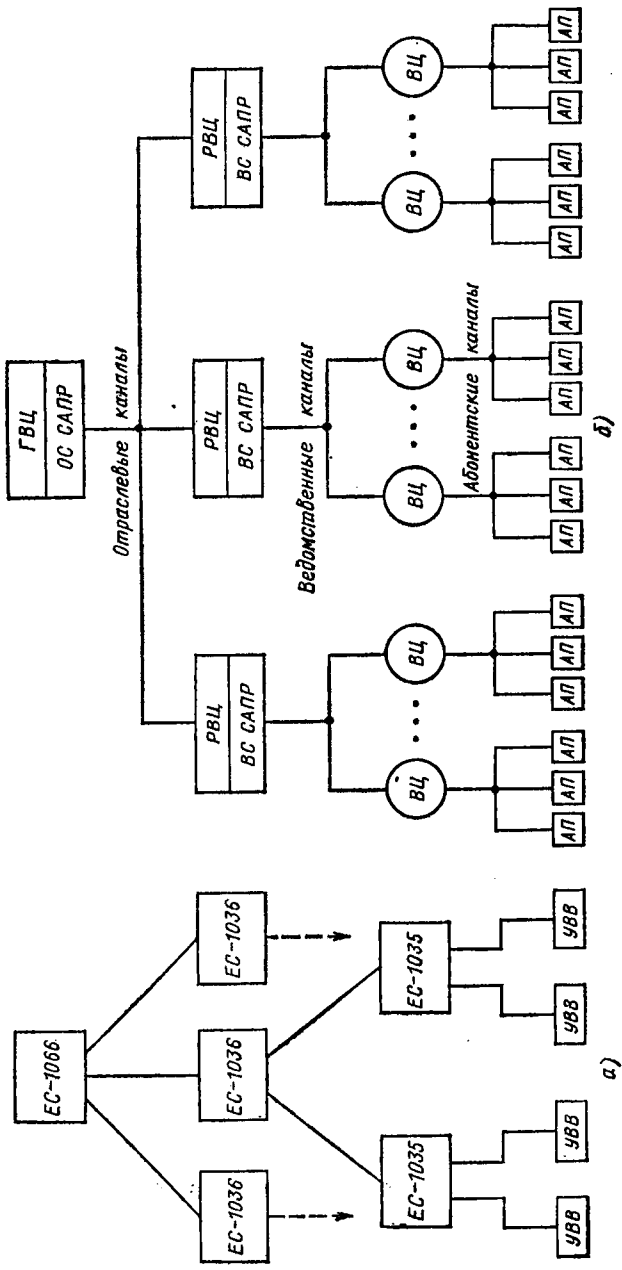
данных используют мини-ЭВМ СМ-1407.01 и СМ-1420. Комплекс АРМ-2.01 предназначен для разработки радиоэлектронной аппаратуры и машиностроительных объектов, позволяет осуществлять ввод графической и алфавитно-цифровой информации, ее редактирование, отображение и выпуск документации на машинных носителях. Одновременно могут работать восемь пользователей. Технические средства АРМ2-01 включают мини-ЭВМ СМ-1407, терминалы подготовки и ввода графических данных СМ-6404, экранные пульты СМ-7316, графопостроители. Базовое программное обеспечение позволяет на основе СМ-6404 описывать на проблемно-ориентированном языке, вводить в ЭВМ и редактировать графическую и текстовую информацию, вести диалог с ЭВМ, создавать на магнитных дисках архив результатов проектирования, выводить информацию на специальные устройства.

Комплексы АРМ-2.02 используют для проектирования деталей машиностроения, технологических процессов их изготовления; АРМ-2.03 — для создания управляющих программ для станков с числовым программным управлением и проектирования цифровой РЭА; АРМ-2.04 — для создания и редактирования программ и выпуска текстовой документации; АРМ2-05 — для отладки управляющих программ и микропрограмм, исследования их в реальном масштабе времени, занесения программ в постоянные запоминающие устройства ЭВМ с программируемой логикой работы.

В состав системы 15УТ-4-017 входят мини-ЭВМ «Электроника 100/25», рабочие места, оборудованные кодировщиками, алфавитно-цифровые и графические дисплеи, координатографы. Система позволяет проектировать топологию БИС и редактировать информацию в интерактивном режиме. При этом анализируют и редактируют эскиз кристалла и топологию компонентов и кристалла, преобразовывают топологию кристалла в информацию для управления микрофотонаборной установкой. Программное обеспечение системы состоит из следующих блоков:

системного обеспечения — включает в себя диспетчер системы и набор вспомогательных программ;

программного обеспечения логического анализа и схематехнического проектирования — позволяет описывать информацию о схеме и получать эскизы логических и принципиальных схем, автоматически формировать массивы связей и контроль содержащейся в них информации;



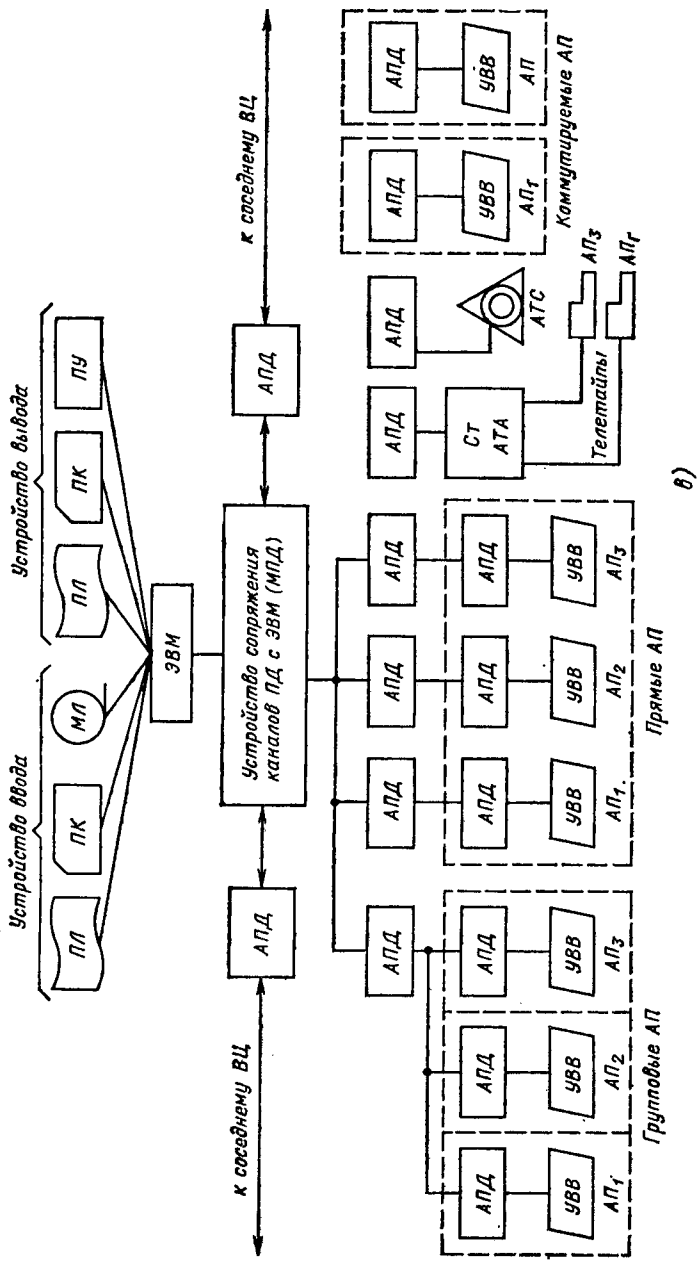


Рис. 2.3. Схема САПР трехуровневая (а), коллективного пользования (б) и сверхбольшая (в):

ПЛ, ПК — перфокарты; МЛ — магнитные ленты; ПУ — печатающее устройство; УВВ — устройство ввода/вывода информации

программного обеспечения проектирования топологии и фотошаблонов — позволяет описывать совмещенный топологический чертеж, проектировать топологию в интерактивном режиме, проверять конструкторско-технологические ограничения, а также выводить фрагменты топологии на графопостроитель;

программного обеспечения общего назначения — выполняет функции реализации директив пользователей и выводит конструкторско-технологическую документацию;

средств развития программного обеспечения — позволяет совершенствовать программное обеспечение комплекса.

Пример структуры ТС САПР для проектирования объектов средней сложности представлен на рис. 2.3, а. Состав ТС включает одну или несколько ЭВМ с возможностью создания иерархической структуры подключения технических средств нижнего уровня.

Организация ТС САПР, предусматривающая использование несколькими конструкторско-проектными организациями, представлена на рис. 2.3, б. В состав ТС входят высокопроизводительная аппаратура передачи данных (АПД) между частями комплекса и мощные вычислительные средства.

Для САПР крупных научно-производственных объединений предполагается применение сверхбольших систем ТС и сетей САПР коллективного пользования. Комплекс ТС таких САПР представляет собой единую вычислительную сеть, объединяющую территориально распределенные центры. Пользователь любого из функциональных уровней может увеличить вычислительную мощность своего уровня ТС за счет подключения его к одному из мощных вычислительных комплексов, содержащихся в этой сети, получить информацию, накопленную другими пользователями, использовать вычислительную технику сети для решения задач (рис. 2.3, в).

Развитие таких систем предъявляет повышенные требования к техническим средствам. Необходимо существенное увеличение емкости и уменьшение габаритов внешних накопителей, уменьшение времени выборки информации. Переход на оптические диски доведет емкость до 200 Гбайт на одну поверхность. Необходимо улучшать характеристики терминалов. Намечается переход на графические терминалы со встроенными функциями обработки изображений, имеющие достаточно большую буферную память, модули для подключения к сетям передачи данных.

Для получения возможности подсоединения к сети различных машинных комплексов на всех уровнях необходимо включить в состав технических средств сетей специальные коммутаторы, комплексирующие разнотипные ЭВМ.

## 2.2. РЕЖИМЫ РАБОТЫ ТЕХНИЧЕСКИХ СРЕДСТВ САПР

Состав технических средств базовых конфигураций САПР различных уровней в значительной степени определяется характером проектных задач. Существует взаимосвязь между классом решаемых задач и режимом использования ЭВМ. Рассмотрим задачи, решаемые в САПР, с целью выделения характеристик, определяющих выбор различных режимов работы ЭВМ.

По характеру вычислительного процесса решаемые задачи можно разделить на две основные группы: задачи, решаемые без участия пользователя, и задачи, в процессе решения которых необходимо участие пользователя.

По сложности вычисления задачи бывают:

задачи первой группы — задачи, на решение которых требуется более нескольких минут; задачи, время счета которых измеряется секундами;

задачи второй группы — время взаимодействия с пользователем соизмеримо с временем счета задачи: время решения велико по сравнению со временем диалога.

По объему информации задачи, решаемые в САПР, можно разделить на монопольно использующие основную память ЭВМ и частично использующие основную память ЭВМ.

Исходя из этой классификации решаемых задач САПР, можно выделить следующие необходимые режимы работы технических средств:

I — однопрограммный режим, при котором решаемой задаче доступны все ресурсы ЭВМ;

II — мультипрограммный режим с фиксированным количеством задач; при таком режиме ОП ЭВМ делится на фиксированное число разделов, которые определены для выполнения одной задачи в каждом; некоторые внешние устройства (ВУ) могут быть назначены для использования несколькими задачами;

III — мультипрограммный режим с переменным числом задач, все ресурсы ЭВМ общие.

Режим работы технических средств можно классифи-

цировать по удалению проектировщика от основного компонента технических средств:

местный режим, при котором пользователь работает непосредственно у ЭВМ;

дистанционный режим, при котором часть периферийного оборудования связана с процессором канала связи.

Режим работы технических средств можно классифицировать по степени участия пользователя в процессе решения задач:

*пакетный режим*, когда пользователь составляет задание на выполнение программы, которое в составе пакета заданий запускается для обработки на ЭВМ. Обработка задач производится по очереди. После решения пользователю требуется проанализировать результаты обработки своего задания и подготовить новый вариант задания, что замедляет отладку и увеличивает время получения окончательных результатов;

*режим разделения времени (РРВ)*, при котором каждой решаемой задаче поочередно выделяется определенный квант времени работы процессора. Пользователь во время сеанса работы за абонентским пунктом, используя средства системы разделения времени (СРВ), может составить, протранслировать, отредактировать программу и приступить к ее выполнению, непосредственно контролируя происходящий процесс. Степень готовности программы зависит от подготовленности пользователя к работе с СРВ.

От выбора правильного режима использования технических средств САПР зависит эффективность эксплуатации технических средств. Поэтому при создании конкретной САПР определенного уровня необходимо провести четкий анализ решаемых задач.

Пакетный режим обработки информации предпочтительнее для задач с большим временем счета и задач, не требующих вмешательства в процесс решения пользователя.

Режим разделения времени удобнее для задач, время счета у которых соизмеримо с временем отклика пользователя на запрос ЭВМ, а также в том числе, когда необходимо вмешательство пользователя в процесс решения.

### 2.3. ТЕХНИЧЕСКИЕ СРЕДСТВА МАШИННОЙ ГРАФИКИ

Технические средства машинной графики можно классифицировать по следующим отличительным признакам: назначению, степени автоматизации, методу обработки ин-

формации, способу отсчета текущих координат. Указанная схема классификации показана на рис. 2.4.

Рассмотрим устройства графического ввода информации. Они позволяют увеличить производительность труда и повысить достоверность кодируемой информации. В полуавтоматических устройствах кодирование входной информа-

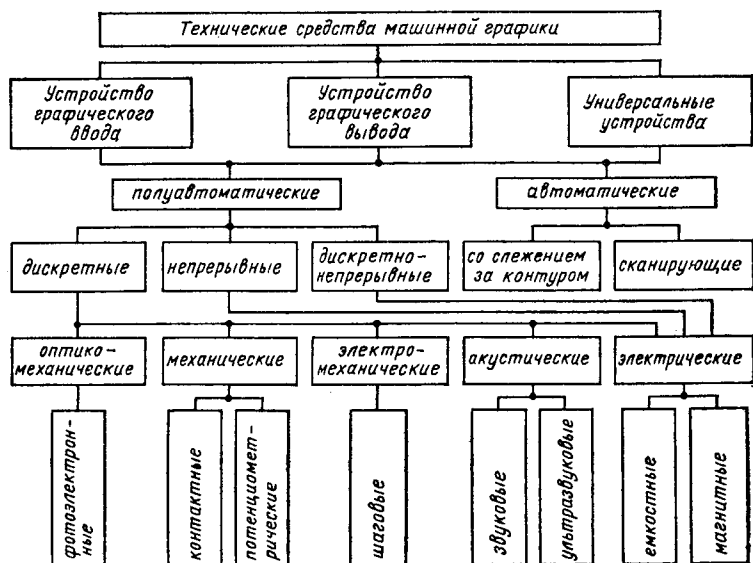


Рис. 2.4. Технические средства машинной графики

ции происходит при участии человека, который, перемещая регистрирующий орган, обходит последовательно все элементы графического документа. Перемещение регистрирующего органа фиксируется. Существуют дополнительные панели для ввода вспомогательной информации (клавиатура печатающего устройства). Представителями таких устройств являются устройства ЭМ-709, ГАРНИ, ПАСТИ, Calcomp, Venson, Aristo и др.

Полуавтоматические системы ввода графической информации могут сопрягаться с ЭВМ, которые выполняют автоматический контроль ошибок, допущенных оператором при считывании, позволяют выбирать фрагменты из библиотеки элементов и ранее введенных участков изображения, осуществляют оперативный вывод закодированной информа-



ции. Примерами таких систем служат системы QED и IED фирмы *Quest Automation* и системы *Aristigrid* фирмы *Aristo*.

Автоматические устройства ввода графической информации характеризуются значительным быстродействием и отсутствием несистематических ошибок, но для их работы требуются графические документы высокого качества и точности.

По методу считывания различают устройства непрерывного, дискретного и непрерывно-дискретного действия. Для графической информации большого объема желательно использовать дискретные устройства, обладающие более жесткими требованиями к точности.

В зависимости от способа отсчета текущих координат устройства ввода графической информации подразделяют на электромеханические, оптико-механические, магнитные, контактные и звуковые.

Примеры устройств ввода:

1) устройство магнитного типа, полуавтомат кодирования графической информации (ПКГИ—0), входящий в состав АРМ—М. Полуавтомат позволяет кодировать элементы чертежа с указанием типа линий, символы 3 алфавитов числом 256 и произвольные кривые, чертежи с учетом смены системы координат;

2) устройство графического ввода FRED фирмы *Quest Automation*, состоящего из планшета с матрицей проводов, расположенных по координатным осям, устройства для цифрового отображения информации, передвижного пульта и ЭВМ NOVA. Задание положения точки определяется при касании щупом поверхности планшета.

Представителями группы устройств вывода графической информации являются чертежные автоматы, позволяющие получать документацию в виде чертежей, графиков, схем, диаграмм. Классификацию чертежных автоматов можно провести по следующим признакам: способу программного управления; методу обработки данных; принципу действия исполнительного механизма.

Классификация чертежных автоматов по этим признакам показана на рис. 2.5. Чертежные автоматы работают как в автономном режиме от машинных носителей информации, так и непосредственно от ЭВМ. Наиболее совершенные чертежные автоматы: ЕС-7051, ЕС-7052, ЕС-7053, ЕС-7054, АП-7251, АП-7252, АП-7253, Calcomp моделей 7000, 925/1036, 905/936, Aristomat моделей 8320, 8350, Benson серии 100, 200, 300 и др.

Чертежным автоматам присущи недостатки: невысокое

быстродействие, недостаточная надежность. Поэтому прямое сопряжение их с ЭВМ не приводит к желательным результатам. Чертежные автоматы входят в комплекс периферийных устройств со своей малой ЭВМ, которая сопрягается с решающими устройствами.

Одними из лучших представителей группы чертежных автоматов являются автоматы «Итекан-2М», «Итекан-3», «Итекан-4», «Итекан-6», различные по назначению, про-

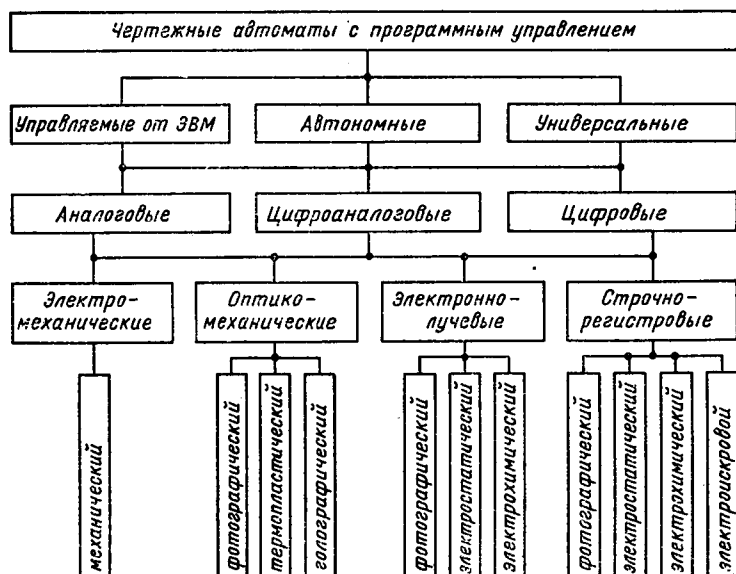


Рис. 2.5. Классификация чертежных автоматов

граммному управлению и конструктивному исполнению. Изображение вычерчивается цветными чернилами, тушью или карандашом на обычной бумаге или кальке линиями различной толщины. Размеры рабочего поля и максимальная скорость вычерчивания для автомата «Итекан-4» составляют  $818 \times 40\,000$  мм и 40 мм/с.

К устройствам графического вывода информации можно отнести программно-управляемые координатографы, предназначенные для изготовления прецизионных фотооригиналов печатных плат, полосковых, микрополосковых линий и др. От чертежных автоматов координатографы отличаются

ся повышенной точностью и способом нанесения изображений. Изображение наносится экспонированием, вырезанием, гравированием, скрайбированием и другими способами.

Программно-управляемые координатографы отличаются большими размерами рабочего поля. Управление осуществляется непосредственно через каналы связи или с помощью промежуточных носителей информации. Наиболее распространенные типы координатографов «Минск», модели М2004, М2005, а также ЭМ-703 и КПА-1200.

Координатограф КПА-1200 предназначен для изготовления фотошаблонов микросхем и печатных плат. В состав координатографа входят: FS-1501, пульт управления, координатный стол с размерами рабочего поля  $1200 \times 1200$  мм, устройство управления с блоками ввода информации, операционное устройство, интерполятор, блоки задания скоростей, обработки информации, ориентации инструмента, технологических операций, управления приводом, цифровой индикации, а также центрального управления. Максимальная скорость перемещений на прямолинейных участках 90 мм/с, на дугах окружностей 25 мм/с.

Наиболее перспективный класс устройств ввода/вывода графической информации — алфавитно-цифровые и графические дисплеи. Скорость обмена информации этих устройств сравнима со скоростью обработки информации в ЭВМ, это обусловило их широкое применение.

Все дисплейные устройства можно классифицировать по следующим основным признакам: назначению и виду отображаемой информации; методу формирования изображения; физическим принципам создания информации. Один из возможных типов классификации дисплеев приведен на рис. 2.6.

Алфавитно-цифровые дисплеи предназначены для обработки символьной информации. Их отличают высокая надежность и экономичность. Рассмотрим основные характеристики наиболее известных алфавитно-цифровых дисплеев.

Экранный пульт ЕС-7925.01 предназначен для оперативной связи пользователя с ЕС ЭВМ в составе комплекса ЕС-7920: емкость экрана 1920 символов; число строк 24; число знаков в строке 80.

Изображение знаков осуществляется векторным обходом луча по контуру знака. Графические дисплеи позволяют выводить на экран графические изображения, имеют широкий набор встроенных функций преобразования инфор-

мации: сдвиг, масштабирование, поворот, редактирование и др.

Рассмотрим технические характеристики наиболее распространенных графических дисплеев.

Дисплей ЕС-7064 предназначен для совместной работы с ЕС ЭВМ. Обеспечивает визуальное отображение на экране электронно-лучевой трубки (ЭЛТ) произвольной графичес-

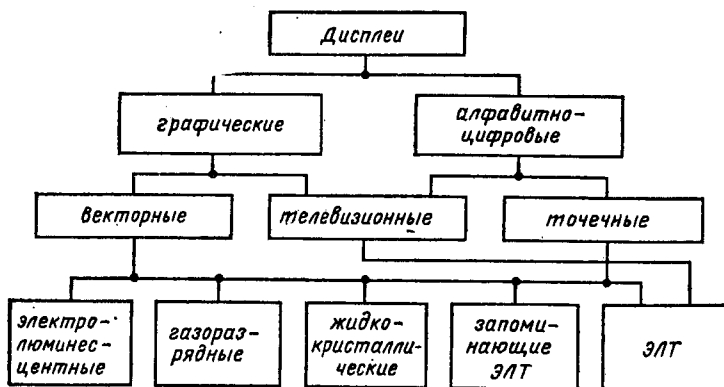


Рис. 2.6. Классификация дисплеев

ской и алфавитно-цифровой информации, хранимой в буферном запоминающем устройстве (ЗУ). Взаимодействие с ЭВМ осуществляется через блок сопряжения с каналом ЭВМ. Размеры рабочего поля дисплея  $250 \times 250$  мм; число адресуемых точек  $1024 \times 1024$ , знаков в строке 74, строк для знаков 52.

Экранный пульт проектирования (ЭПП) А-5433 предназначен для оперативной двусторонней связи (диалога) проектировщика с ЭВМ на графическом языке. Изображение на экране ЭЛТ формируется мозаичным, векторным или текстовым методом.

Экранный пульт проектирования может работать в трех режимах: ЭВМ — непосредственная логическая связь с ЭВМ, АВТОНОМ — автономное функционирование, КОНТРОЛЬ — вспомогательный контрольный режим. Размеры рабочей площади экрана  $384 \times 256$  мм; формат точечного раstra  $192 \times 128$  точек.

В заключение отметим, что тенденции развития графических средств САПР определены в сторону повышения бы-

стросдействия обмена информацией между конструктором и ЭВМ. Поэтому в последнее время отмечается динамичный рост разработок в области экранных пультов, обладающих большим набором встроенных функций обработки изображений, возможностью его идентификации и обработки на уровне макрофрагментов.

#### 2.4. ОРГАНИЗАЦИЯ ВЗАИМОДЕЙСТВИЯ КОНСТРУКТОРА С ЭВМ

Основные тенденции развития САПР определяют преимущественное развитие интерактивных систем автоматизированного проектирования. Центральное место в таких системах занимает диалог конструктор — ЭВМ. Организация диалога обеспечивается информационными, программными и техническими средствами САПР.

При выборе технических средств САПР, обеспечивающих интерактивное взаимодействие конструктора с процессом решения, среди возможных альтернатив следует руководствоваться следующим.

Используемая ЭВМ должна позволять организовать работу определенного числа пользователей (т. е. удовлетворять требованиям по производительности) в режиме разделения времени.

Используемое терминальное оборудование должно отвечать эргономическим эстетическим требованиям и требованиям эффективности работы.

К терминальному оборудованию можно отнести следующие устройства: пультовую пишущую машинку, телетайп, алфавитно-цифровой и графический дисплей, акустические устройства.

Применение этих устройств обусловлено определенным классом решаемых задач в САПР. В настоящее время практически во всех применениях становится экономически невыгодным использование телетайпов и терминалов с посимвольным вводом/выводом информации. Дисплеи со страничной обработкой информации позволяют увеличить производительность работы проектировщика.

Применение графических дисплеев идет в направлении представления терминалов как автономных систем со специальными операционными системами управления работой отдельных аппаратных и программных компонентов терминала и взаимосвязью терминала с основной ЭВМ. Появление супермикро-ЭВМ, базиса автономного комплекса, прибли-

зило комплексы по своим возможностям к мини-ЭВМ. Развитие комплексов идет по двум направлениям разработки однопрограммных систем, рассчитанных только на одного пользователя, и мультипрограммных систем, рассчитанных на одновременную работу нескольких пользователей, с реализацией виртуальной памяти.

Для достижения высокой производительности систем, необходимой для одновременной работы многих пользователей и соединения функциональных блоков системы, используют различные каналы, объединяющие процессор, оперативную память, матричный процессор и различные источники информации.

Схема одного из автономных терминальных комплексов показана на рис. 2.7. В графических терминалах для уп-

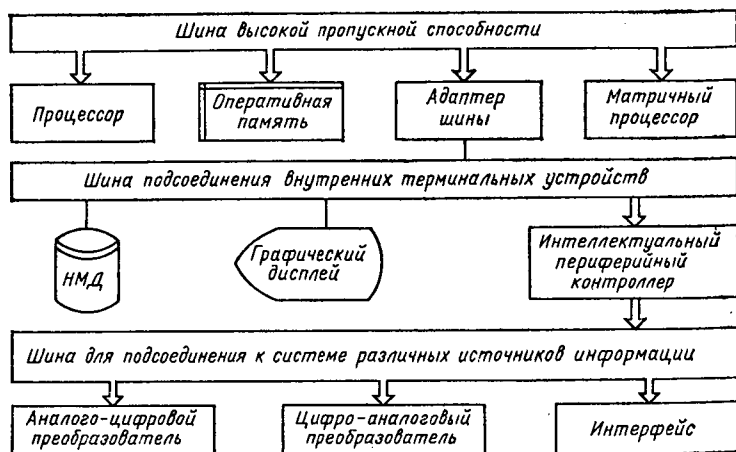


Рис. 2.7. Схема автономного терминального комплекса

равления функционированием всех компонент применяют микропроцессоры; для обработки графической информации необходимы специализированные процессоры; к терминалу могут подключаться устройства памяти.

В целом тенденции развития терминалов могут быть охарактеризованы как движение в сторону многофункциональных систем с распределенной обработкой информации, основанных на широком использовании БИС.

В качестве примера рассмотрим вариант организации взаимодействия конструктора с ЭВМ при реализации на вычислительных центрах.

Комплекс технических средств представляет собой базовый комплект ЭВМ типа ЕС и включает в себя оперативное запоминающее устройство (ОЗУ) емкостью 512 К; накопители на магнитных дисках ЕС-5061, 29 Мбайт — 4 шт.; устройство ЕС-8410 — базовый комплект с четырьмя абонентскими пунктами ЕС-7920.

Операционная система ОС ЕС ЭВМ версии 6.1 работает в режиме мультипрограммной обработки заданий. Система предназначена для организации диалоговых систем коллективного доступа на базе ЕС ЭВМ. Ее основные функции заключаются в обеспечении одновременной и независимой работы группы пользователей, получающих доступ к ресурсам вычислительной системы через алфавитно-цифровые дисплеи.

Структурный состав комплекса можно варьировать в зависимости от уровня САПР, но даже в одном уровне его конфигурация определяется только в условиях конкретной разработки на основе анализа состава решаемых задач.

## 2.5. ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ САПР

Эволюция развития комплекса технических средств САПР характеризуется созданием территориально рассредоточенных многомашинных систем сбора, хранения и обработки информации, реализованных в виде вычислительных сетей. Последние, рассредоточенные на небольших территориях предприятий и объединяющие в единую информационную систему автоматизированные рабочие места пользователей, ЭВМ и микро-ЭВМ, графопостроители, терминальные станции и другую специализированную аппаратуру, называют локальными вычислительными сетями (ЛВС). Локальные ВС имеют открытую архитектуру, обеспечивающую возможность подключения к сети любых других ЛВС, в том числе и крупных сетей ЭВМ. Основное достоинство ЛВС — низкая стоимость системы передачи данных.

Локальные вычислительные сети САПР должны обеспечивать: использование режимов пакетной и диалоговой обработки, разделения времени, виртуальной памяти; экономичную обработку информации по принципу «наиболее важные процессы САПР выполняются техническими средствами с развитым программным обеспечением и высокой производительностью, наименее ответственные — на дешевых мини- и микро-ЭВМ»; высокую надежность и достоверность функционирования, высокую производительность; применение разнообразного проблемно-ориентированного ПО, централизованных и локальных БД с необходимым объемом памяти; работу с автоматизированными рабочими

При использовании в ЛВС нескольких методов управления средой передачи данных существенно увеличивается сложность схемных решений контроллеров, с помощью которых станции ЛВС подключаются к среде передачи данных.

*Принцип информационной и программной совместимости* предусматривает совместимость операционных систем, программ и систем управления базами данных (СУБД), распродоточенных в рамках ЛВС.

Особенность этого принципа — возможность адаптации процессов к видам пересылаемой информации и применение единых систем кодирования и контроля информации.

*Принцип гибкой модульной организации* предусматривает проектирование СПД ЛВС на основе набора гибких конструктивно законченных модулей.

Локальные вычислительные сети классифицируют:

по топологическим признакам — иерархической, кольцевой и звездообразной конфигурации, конфигурации типа «общая шина»;

по методам управления ресурсами среды передачи данных — с детерминированным и случайным доступом к мультисектору;

по программному обеспечению — с единой операционной поддержкой и едиными методами теледоступа, ориентированными на конкретную ЛВС и ЛВС с различными наборами тех и других компонентов операционной поддержки;

по методу передачи данных — сети с коммутацией каналов, с коммутацией сообщений и коммутацией пакетов, причем в современных ЛВС характерно использование коммутации пакетов;

по техническому обеспечению — гомогенные и гетерогенные ЛВС. Первые предусматривают применение в станциях однотипного оборудования, например только комплексов в машинной графике. Вторые дают возможность подключения любых абонентских комплексов — от устройств выдачи конструкторской документации до высокопроизводительных вычислительных комплексов с развитой терминальной сетью (рис. 2.9).

Анализируя способы реализации технического обеспечения САПР на базе стандартных многоуровневых структур вычислительных центров коллективного пользования и на базе ЛВС, можно сделать следующие выводы. Сетевая архитектура по сравнению со стандартной многоуровневой имеет много преимуществ: возможность взаимодействия с одним и того же терминала с ресурсами всех рабочих



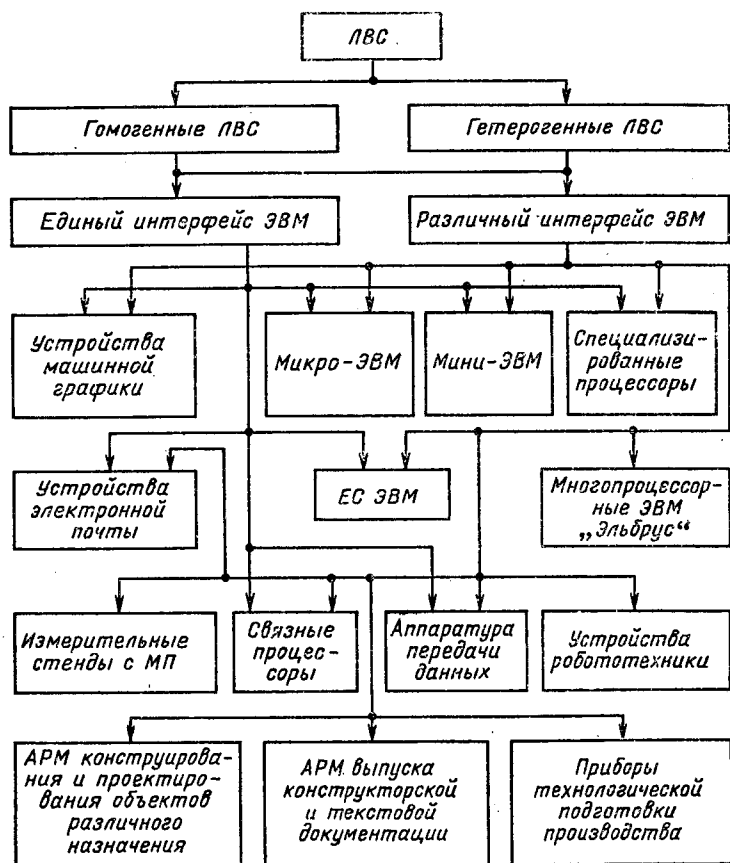


Рис. 2.9. Классификация ЛВС по техническому обеспечению

и терминальных машин ЛВС; обеспечение высокой надежности обработки путем замены вышедшей из строя рабочей машины резервной; повышение эффективности функционирования ЭВМ за счет их специализации на выполнение определенных функций хранения и управления данными, геометрического моделирования, подготовки управляющей информации для программного управляемого оборудования и т. д.

На рис. 2.10 показан пример реализации иерархической ЛВС. Центральную рабочую систему (ЦРС), реализованную на базе мощных ЭВМ типа «Эльбрус» или старших мо-

делей ЕС ЭВМ, и коммуникационно-административную систему (КАС) территориально располагают в одном здании в непосредственной близости друг от друга, а терминальные системы могут быть удалены на расстояние до 10 км. Одна из мини-ЭВМ КАС кроме функций основного управления, связанных с коммутацией и маршрутизацией пакетов сообщений, осуществляет функции административного управления, которые заключаются в сборе статистики и учете работы систем и каналов, производит изменение конфигурации

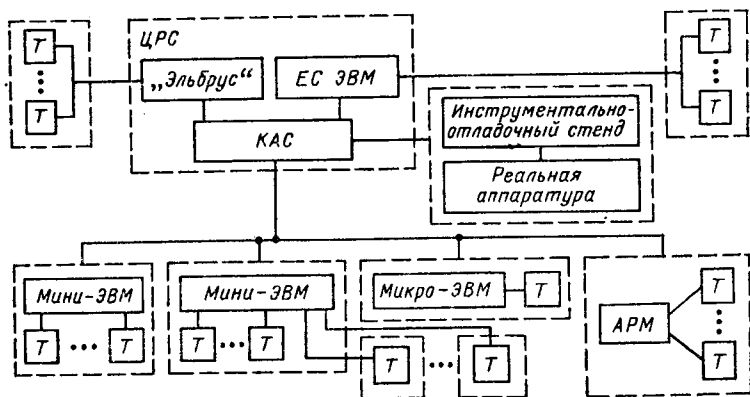


Рис. 2.10. Структура иерархической ЛВС

сети и выполняет процедуры управления потоками данных. Программное обеспечение ЛВС (рис. 2.10) осуществляет обмен файлами, пересылку данных, запуск заданий с терминалов, контроль их выполнения, вывод результатов на одно из устройств вывода или просмотр их с терминала *T*.

Пример структуры ЛВС, реализующей логико-информационные связи организационных элементов САПР сложных изделий электронной техники — больших (БИС) и сверхбольших (СБИС) интегральных схем с большой степенью интеграции, приведен на рис. 2.11. С созданием таких САПР разработчик получает возможность учитывать логику работы, электрические параметры наборов типовых элементов БИС и СБИС с различной геометрией на кристалле, контролировать различные этапы разработки, так чтобы несколько разработчиков, проектирующих различные фрагменты изделий, могли использовать их правильные версии,

объединять графическую информацию с текстовой документацией и т. д. В такой системе информационные связи должны пронизывать весь маршрут проектирования БИС и СБИС — от составления ТЗ до выпуска соответствующей документации и изготовления опытной партии микросхем.

Система состоит из двух частей: функциональной и обеспечивающей. В функциональную часть САПР для компоновки логических схем кристаллов БИС входят следующие ав-

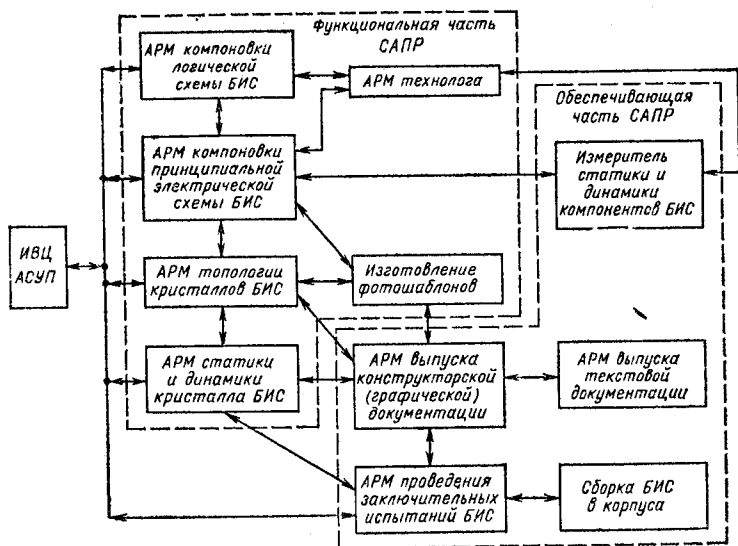


Рис. 2.11. Пример организации САПР БИС на базе ЛВС

томатизированные рабочие места (АРМ): для компоновки логической и принципиальной электрической схемы кристалла БИС; для компоновки отдельных топологических заготовок топологии кристалла; для измерения параметров статике и динамики кристалла БИС; для изготовления технологической документации, сопровождающей производство БИС; для изготовления фотошаблонов с заданной точностью топологических элементов и их запоминания в цифровом коде на магнитных носителях.

В обеспечивающую часть САПР входят АРМ: для проведения измерений статических и динамических параметров компонентов БИС; для проведения заключительных испы-

таний после сборки БИС в корпуса; для выпуска конструкторской графической и текстовой документации.

Все АРМ снабжены накопителями, в которых хранятся соответствующие файлы информации: описание типовых логических схем компонентов БИС, описание и характеристики электронных элементов БИС (транзисторов, диодов и т. д.), данные для планирования технологии изготовления БИС и другие справочные и архивные данные. Все файлы через внешние связи подключены к центральному БД информационно-вычислительного центра (ИВЦ) предприятия.

Связь между элементами ЛВС осуществляется с помощью общей шины моноканала с высокой пропускной способностью. Структура информационных связей САПР (рис. 2.11) представляет собой исходную информацию для разработки логической структуры ЛВС. Расстояния между различными узлами сети зависят от их территориального расположения на предприятии, а интенсивность обмена между узлами — от сложности проектируемого объекта. В зависимости от расстояний между узлами ЛВС и планируемой интенсивности обмена информацией выбирается аппаратура передачи данных.

Следует отметить, что структура и состав технических средств, входящих в ЛВС предприятия, целиком зависят от номенклатуры задач автоматизированного проектирования, интенсивности поступления и трудоемкости их решения.

## 2.6. АППАРАТУРА ПЕРЕДАЧИ ДАННЫХ

Основными составляющими сети ВК САПР для обмена информацией являются: каналы передачи данных, состоящие из каналов связи, оборудованных аппаратурой передачи данных (АПД); устройства сопряжения каналов ПД с ЭВМ, называемые мультиплексорами передачи данных (МПД), и оборудование абонентских пунктов. Последние состоят из средств передачи данных и внешних устройств: устройства ввода/вывода данных информации в (из) ЭВМ и печатающих устройств.

Каналы и сети связи, используемые для передачи данных, разделяют на первичные и вторичные. Под *первичной сетью связи* понимают совокупность технических средств связи, используемых для каналообразования. В состав такой сети входят воздушные, кабельные, радиорелейные линии, линии коротковолновой УКВ и радиосвязи, а также линии связи через искусственные спутники Земли. Испол-

зую каналы первичной сети, можно создавать вторичные каналы и сети, предназначенные для передачи какого-либо одного вида информации. Каналы САПР относятся ко вторичным сетям.

В современных сетях САПР применяют оперативный способ коммутации абонентских пунктов, при котором обеспечивается соединение входящих и исходящих каналов только на время передачи сообщения. После передачи сообщения соединение нарушается. При оперативной коммутации используют способ образования транзитивных трактов, построенный на сквозном принципе, при котором до начала сеанса передачи сообщения абонент устанавливает сквозное соединение каналов. Только после этого он приступает к передаче сообщений. При занятости требуемого абонентского пункта или канала связи информацию записывают на запоминающее устройство (ЗУ) ближайшего центра коммутации (ЦК) и хранят там до освобождения канала ПД.

Аппаратура передачи данных состоит из устройств преобразования сигналов (УПС), обеспечивающих прямое и обратное преобразование сигналов в вид, пригодный для передачи по каналу связи, устройства защиты от ошибок, входного кодера и выходного декодера канала ПД.

Главной характеристикой канала является вид сопряжения, который обеспечивается внешнему абоненту, а также внешнему устройству или комплексу устройств. Типовыми примерами являются пословный, посимвольный и последовательный (разрядный) интерфейсы. В каждом из этих случаев канал будет производить преобразование данных из формата, получаемого от устройства, в формат канала или из формата канала в формат, воспринимаемый устройством. Так, последовательный поток двоичных разрядов собирается в слова и при необходимости запоминается в буфере. При передаче же данных из памяти во внешнее устройство слово, получаемое из памяти в параллельном коде, преобразуется в последовательный поток двоичных разрядов, который после этого преобразования может быть принят данным внешним устройством. Аналогичным способом будут разбираться слова на символы, а символы на слова. Преобразование формата может включать в себя такую операцию, как удаление или добавление двоичных разрядов контроля.

Для некоторых областей применения может потребоваться специальный канал, который выполняет преобразование некоторого внешнего кода во внутренний. Для каж-

дого слова, пересылаемого в ходе данной передачи, должен быть известен адрес. Этот адрес обновляется после каждого цикла памяти и проверяется на предмет определения конца передачи. Функция управления передачами в современном развитии аппаратных средств САПР возложена на канал. Такой подход позволяет сосредоточить все специализированные средства сопряжения в одном определенном модуле, что исключает необходимость использования разных вариантов базового комплекса или специальных коммутаторов памяти в разных установках. Другими функциями, относящимися к управлению передачами, являются выработка сигналов прерывания, передач и выработка сложной системы сигналов, которые могут потребоваться для взаимодействия абонентского пункта и коммутатора памяти.

Во многих каналах выполняются контроль и коррекция ошибок. Типовой способ обнаружения ошибок сводится к некоторому виду контроля по четности.

Требование совместимости скорости различного абонентского пункта не всегда выполняется, поэтому в современных каналах применяют буфер в виде нескольких каскадов сборочно-разборочных регистров.

Канал может быть реализован в виде специализированного устройства, предназначенного для выполнения только канальных функций. В то же время (и эта тенденция все больше усиливается) канальные функции берут на себя малые сравнительно быстрые ЭВМ. При использовании их в таком качестве пропускная способность их ЗУ должна быть существенно выше, чем пропускные способности внешних устройств, которыми она управляет.

## 2.7. ПРИМЕРЫ СТРУКТУР САПР ЭВА И РЭА

Система автоматизированного проектирования БИС имеет трехуровневую структуру. Верхний уровень составляет центральный вычислительный комплекс (ЦВК). Технические средства ЦВК представлены тремя ЭВМ БЭСМ-6, которые связаны друг с другом с помощью специальных адаптеров, эти ЭВМ имеют общее поле внешней памяти на магнитных дисках. В ЦВК входят внешняя память на магнитных барабанах, лентах, дисках, стандартный набор устройств ввода/вывода, возможно подключенные до 16 алфавитно-цифровых дисплеев и их использование в режиме разделения времени. Общее программное обеспечение представлено операционной системой ДИСПАК, мониторинговой системой МОНИТОР-80, включающей трансляторы с ряда языков программирования, диалоговой системой общего назначения КРАБ. Система КРАБ

позволяет создавать и редактировать файлы, защищает их от несанкционированного доступа.

Развитие ЦВК в САПР БИС осуществляется в направлении перехода от использования ЭВМ БЭСМ-6 к применению более производительного многопроцессорного вычислительного комплекса (МВК) ЭЛЬБРУС. Наличие в МВК ЭЛЬБРУС специализированного процессора СВС, являющегося аппаратным эмулятором системы команд БЭСМ-6, обеспечивает преемственность в использовании ранее созданного программного обеспечения. Центральный вычислительный комплекс предназначен для выполнения проектных процедур, требующих значительных вычислительных ресурсов.

Второй уровень САПР БИС составляют автоматизированные рабочие места (АРМ) проектировщиков. Главное назначение АРМ — обеспечение интерактивного режима работы проектировщика в САПР. Значительное место во взаимодействии проектировщика с ЭВМ занимает обмен графической информацией. Это обуславливает наличие в АРМ развитых средств машинной графики и объясняет другое название второго уровня — интерактивный графический комплекс (ИГК). Для управления функционированием периферийных устройств, входящих в АРМ, и выполнения проектных процедур, не требующих больших объемов вычислений, в состав каждого АРМ входит мини-ЭВМ «Электроника—100/25» или «Электроника-79».

Третий уровень САПР БИС представляет собой технологический комплекс (ТК), предназначенный для оформления проектной документации и управления фотонаборными установками, служащими для изготовления фотошаблонов.

В САПР БИС выделено несколько подсистем, имеющих свое программное обеспечение в ЦВК и в ИГК. Основными проектирующими подсистемами САПР БИС являются подсистемы структурно-приборного, схемотехнического, логического и топологического проектирования.

Система автоматизированного проектирования высокопроизводительных ЭВМ КОМПАС-82 является развитием более ранней версии — системы КАСПИ. Система базируется на использовании ЭВМ БЭСМ-6 с переходом на МВК ЭЛЬБРУС. Общее программное обеспечение составляют операционная система ДИСПАК и мониторная система МОНИТОР-80. Система КОМПАС-82 состоит из четырех основных подсистем: логического и конструкторского проектирования, выдачи документации и генерации тестов.

Подсистема логического проектирования ПУЛЬС предназначена для моделирования распространения сигналов и получения временных диаграмм. В подсистеме конструкторского проектирования выполняют процедуры компоновки и трассировки межсоединений в БИС. В подсистеме выдачи документации производятся оформление конструкторской документации и изготовление управляющих лент для технологи-

ческих автоматов. В подсистеме генерации тестов синтезируются как контролирующие, так и диагностические тесты. В КОМПАС-82 имеется единая БД со специализированной СУБД. В базе данных различают постоянную часть, содержащую каталоги унифицированных элементов, сведения из руководящих технических материалов и другую справочную информацию, а также логическую и техническую части, содержащие описания соответственно функционально-логических схем и конструкций ЭВМ.

Система автоматизации проектирования (ЕСАП ЭВТ) предназначена для автоматизации проектирования технических средств «Ряд-3», базирующихся на использовании комплектов микропроцессорных БИС, сверхбыстродействующих устройств памяти и интерфейса, прецизионных многослойных печатных плат, перспективных внешних устройств ЕС ЭВМ.

Система АП ЭВТ позволяет автоматизировать этапы структурного, логического и технического проектирования сложных вычислительных систем в конструкциях с пятиуровневой иерархией с учетом жестких ограничений, накладываемых быстродействующей элементной базой.

Средства настройки ЕСАП ЭВТ обеспечивают проектирование изделий с различными элементными и конструктивными базами. Система обладает диалоговыми средствами для обеспечения оперативного обнаружения, локализации и устранения синтаксических, семантических и структурно-алгоритмических ошибок.

Система АП ЭВТ использует технические средства и операционную систему ЕС ЭВМ, минимальная конфигурация технических средств должна включать в себя ЭВМ с производительностью не менее 500 тыс. операций/с, емкостью оперативной памяти до двух мегабайт и внешней памятью до 500 Мбайт.

Система АП ЭВТ обеспечивает независимость входных информационных потоков от конфигурации используемых программных и технических средств, единство структурного представления данных для всех уровней проектирования, имеет агрегативный принцип организации программных и аппаратных средств и модульную структуру.

Архитектурные принципы построения ЕСАП определяются в основном неизменностью в целом существующей конструктивной иерархии проектирования ЭВМ. Отсюда следует перечень основных подсистем ЕСАП: автоматизированного управления проектом; генерации версий ЕСАП; информационно-справочная подсистема разработчика; информационного обеспечения ЕСАП ЭВТ; проектирования ТЭЗ; проектирования панелей, рам (блока) и стойки (устройства); ведения документации ЕСАП ЭВТ; испытаний ЕСАП ЭВТ; синтеза логики; автоматизированного теплового проектирования; средства связи ЕСАП с АСУТП и автоматизированным оборудованием.

Стремление обеспечить высокие показатели по производительности,



гибкости и настраиваемости ЕСАП приводит к образованию таких режимов эксплуатации системы, когда в ЕСАП одновременно проводится проектирование многих конструктивных единиц по нескольким проектам. Это обуславливает необходимость включения в состав ЕСАП специальной подсистемы автоматизированного управления проектом с целью сокращения сроков проектирования и более оптимального использования возможностей ЕСАП.

Информация о состоянии проектирования конструктивных единиц, накапливаемая в БД системы в процессе проведения проектирования, может быть использована пользователем для обработки статистических данных о процессе проектирования.

Подсистема проектирования ТЭЗ предназначена для автоматизации всего цикла работ по проектированию, контролю и изготовлению ТЭЗ, выполненных на базе двустороннего и многослойного монтажа, а также тонкопроводного монтажа. Программной реализацией методов решения этих проблем в подсистеме являются три соответствующих пакета программ. Стыковка с АСУТП осуществляется как через машинные носители для программно-технологического оборудования, так и через БД ЕСАП.

Подсистема проектирования панели и рамы (блока) предназначена для обеспечения монтажно-коммутационного проектирования панели и рамы прямоугольной конструкции, допускающей планарное представление монтажного поля. В проектируемом объемном монтаже могут быть использованы различные типы плоского кабеля, дискретных кабельных изделий и соединителей. Подсистема рассчитана на проектирование блока, содержащего до 45 тыс. электрических контактов (типовая рама ЕС ЭВМ с 6 панелями и 135 контактными соединителями). Минимальной моделью ЭВМ для эксплуатации подсистемы можно считать ЕС-1022 с объемом дисковой памяти до 150 и оперативной памяти — 512 кбайт.

Конструктивная сложность объекта проектирования, измеряемая числом микросхем, распределена в диапазоне от  $10^3$  до  $10^5$ . Соответственно логическая (или информационная) сложность имеет диапазон изменения от  $10^3$  до  $10^8$  вентилях на объект проектирования. Число серий микросхем, с которыми приходится иметь дело в ЕСАП, более 10. При проведении анализа возможных конструктивных решений за основу берется следующая иерархия:

- |                        |                       |
|------------------------|-----------------------|
| 1-й уровень — ИС, БИС; | 4-й уровень — рама;   |
| 2-й уровень — ТЭЗ;     | 5-й уровень — стойка. |
| 3-й уровень — панель;  |                       |

Анализ практики проектирования с применением средств ЕСАП показывает, что при тех же нормах ежемесячного выпуска спроектированных комплектов документации, что и при неавтоматизированном про-

ектировании, удалось улучшить ряд показателей, а именно: а) трудоемкость разработки одного конструктивного узла снижена на 40—60 %; б) время, затрачиваемое на разработку одного конструктивного узла, сократилось в 1,8 раза при автоматизированном способе проектирования и в 3,2 раза при автоматическом проектировании; в) число дефектов в проектной документации сократилось на 15 %; г) за счет автоматизации процессов изготовления и контроля ошибки производства сократились в среднем более чем на 90 %.

В настоящее время широко распространены системы РАПИРА, используемые для функционального и конструкторского проектирования РЭА и ЭВА, СВЧ устройств, микросборок, плоских конструктивов, управляющих перфолент для станков с ЧПУ и др. Одна из модификаций этой системы проектирования РАПИРА—5.3—82 представляет собой комплекс пакетов прикладных программ, предназначенный для автоматизации проектирования РЭА и ЭВА на ЕС ЭВМ и выполняющий конструкторское проектирование двусторонних печатных плат, тонкопленочных и толстопленочных микросборок. В состав системы входят программные средства: базовое программно-информационное обеспечение (БПИО), подсистема конструкторского проектирования микросборок, подсистема конструкторского проектирования двусторонних печатных плат (ДПП). Система функционирует на ЕС ЭВМ модели не ниже ЕС-1022 стандартной конфигурации (ОЗУ-512к). Для функционирования системы дополнительно используют координатографы, графопостроители, сверлильные станки.

Базовое ПИО системы содержит следующие средства: управления процессом автоматизированного проектирования объектов на основании директив пользователей, управления и реализации обмена информацией между ОП и внешними носителями данных ЭВМ, реализации унифицированных для всех подсистем вычислительных процессов, управления и ведения базы данных и архивов системы.

Подсистема конструкторского проектирования тонко- и толстопленочных микросборок рассчитывает геометрию резисторов, размещение разногабаритных элементов, тепловой режим элементов микросборок, трассировку соединений, выпуск конструкторско-технологических документов. Работа подсистемы на ЭВМ происходит в пакетном режиме. В комплект выходных документов входят фотооригиналы, послойные чертежи, сборочный чертеж, таблица координат и цепей, перечень элементов, спецификация.

Подсистема конструкторского проектирования ДПП в

пакетном режиме выполняет размещение разногабаритных элементов, трассировку соединений и выпуск конструкторско-технологических документов. Подсистема позволяет проектировать ДПП с переходными металлизированными отверстиями. Выходными документами подсистемы являются фотооригиналы, сборочный чертеж, таблицы цепей, перечень элементов, спецификация.

Развитием системы РАПИРА является ПРАМ, содержащая подсистемы формирования технологических процессов, моделирования аналоговых схем, формирования и выдачи текстовой конструкторско-технологической документации, проектирования механических конструкций РЭА и др.

### **Контрольные вопросы и задания**

1. Приведите основные требования к составу и структуре технических средств САПР.
2. Перечислите основные компоненты комплекса технических средств САПР, их задачи и характеристики.
3. Постройте общую логическую структуру технических средств САПР и поясните взаимодействие ее звеньев.
4. Что такое базовая конфигурация технических средств, и какие требования необходимо учитывать при ее разработке?
5. Поясните преимущества объединения технических средств САПР в вычислительные сети.
6. Что называют протоколом связи в вычислительных сетях?
7. Поясните состав и функции аппаратуры передачи данных.
8. Какие режимы работы технических средств Вы знаете? Приведите их достоинства и недостатки.
9. Приведите схему классификации средств машинной графики.
10. Опишите устройства графического ввода информации.
11. Опишите основные характеристики устройств вывода графической информации, приведите примеры.
12. Приведите характеристики алфавитно-цифровых дисплеев.
13. Приведите характеристики графических дисплеев.
14. Как организовано взаимодействие конструктора с ЭВМ в САПР?

## **ГЛАВА 3**

### **ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ САПР**

#### **3.1. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ**

Основу информационного обеспечения подсистем САПР составляет совокупность данных, которые необходимы для выполнения процесса проектирования. Совокупность компо-

ентов ИО образует информационную базу, называемую базой данных (БД) САПР.

*Данные* — это сведения о некоторых фактах, позволяющие делать определенные выводы. Взаимосвязанные данные часто называют *системой данных*, и хранимые данные называют информационным фондом. Основное назначение информационного обеспечения — предоставлять пользователям САПР достоверную информацию в определенном виде.

Объектом считают любой предмет, событие, понятие и т. п., о которых приводятся данные. Все объекты характеризуются атрибутами. Например, объект ЭВМ можно характеризовать такими атрибутами: скоростью вычислений, объемом оперативной памяти, числом элементарных операций, числом процессоров, габаритами, количеством мультиплексных каналов и т. п. Сведения, содержащиеся в каждом атрибуте, называют *значениями данных*.

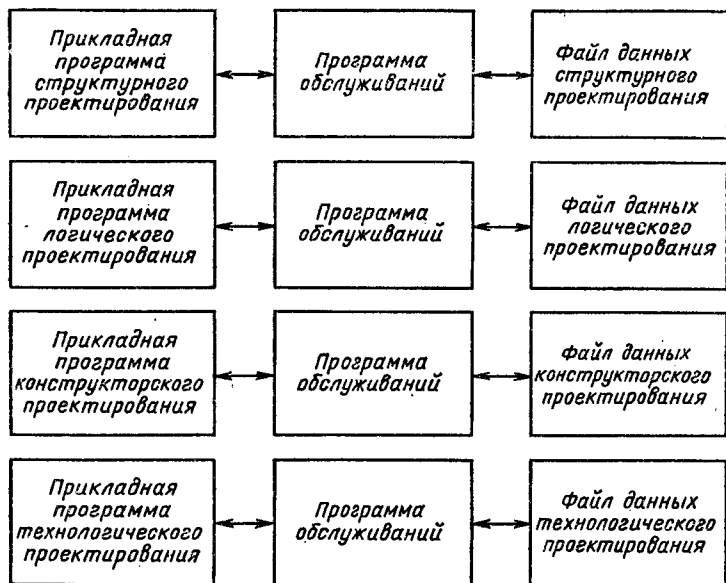
Ниже приведены значения данных для ЭВМ ЕС-1050:

Скорость вычислений . . . . .	5 · 10 <sup>5</sup> оп/с
Емкость оперативной памяти . . . . .	128—1024 кбайт
Скорость работы печатающего устройства . . . . .	900 строк/мин
Заводской номер . . . . .	785

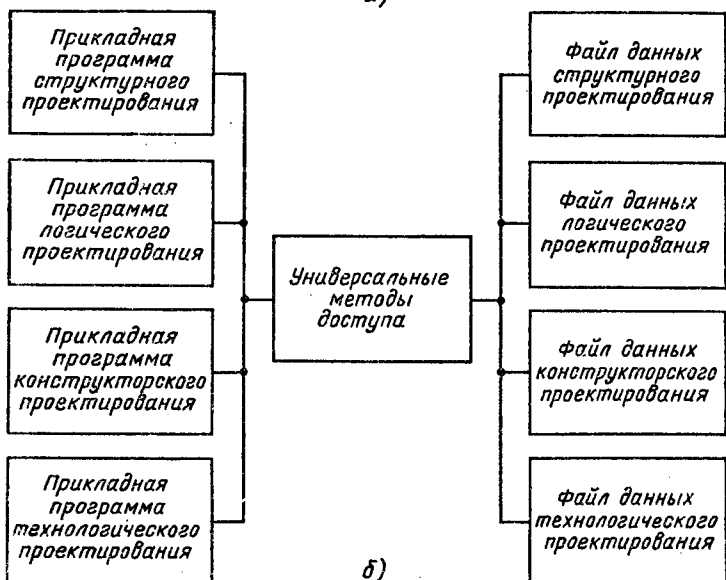
Среди атрибутов имеются такие, по значениям которых возможна идентификация объекта. Например, если известен заводской номер ЭВМ 785, то можно определить, что это ЕС-1050, ее скорость вычислений и т. д.

Атрибуты, по значениям которых определяют значения других атрибутов, называют *идентификаторами объекта* или *ключевыми элементами данных*. Отметим, что один и тот же объект могут идентифицировать несколько элементов данных. Их тогда считают кандидатами в идентификаторы. Проблему выбора идентификатора из нескольких кандидатов решает пользователь САПР.

Объединение значений связанных атрибутов называют *записью данных*. Например, 785 ЕС-1050 — запись данных. Упорядоченную совокупность записей данных называют *файлом данных* или *набором данных*. В настоящее время разработаны универсальные подпрограммы, реализующие методы доступа к файлам, их обработку и обслуживание. Эти подпрограммы являются частью ОС ЭВМ. На рис. 3.1, а показано взаимно-однозначное соответствие между прикладными программами и файлами данных в САПР печатных плат. На рис. 3.1, б показано использование универсальных методов доступа вместо единичных подпрограмм обслуживания файлов данных.



а)



б)

Рис. 3.1. Взаимно-однозначное соответствие между прикладными программами и файлом данных (а) и пример использования универсальных методов доступа (б)

При обработке данных с расположением одних и тех же элементов данных в нескольких файлах возникают трудности, связанные с избыточностью данных, что требует нескольких процедур ввода, обновления и формирования; опасностью нарушения непротиворечивости данных, что связано с хранением одной и той же информации в нескольких местах; ограниченностью разделения данных; ограничениями по доступности данных; негибкостью к изменениям; сложностью управления процессом решения задачи.

Существует много адекватных и в то же время неформальных определений базы данных. Приведем те из них, которые получили наибольшее распространение.

*База данных* — это совокупность специально организованных данных, рассчитанных на применение в большом количестве прикладных программ. *База данных* — это набор файлов, работа с которыми обеспечивается специальным пакетом прикладных программ — системой управления базой данных — с целью создания массивов данных, их обновления и получения справок. *База данных* — это совокупность данных, отображающая состояние объектов и их отношений в рассматриваемой предметной области. *База данных* — совокупность всех тех данных, которые обрабатываются более чем в одной программе (модуле).

Приведем основное различие между БД и файлом данных. Файл данных имеет несколько назначений, но соответствует одному представлению хранимых данных. База данных имеет также несколько назначений, но соответствует различным представлениям о хранимых данных.

Сформулируем основные требования к БД:

1. Целостность данных — их непротиворечивость и достоверность.

2. Организация БД должна обеспечивать согласование времени выборки данных прикладными программами с частотами их использования прикладными программами САПР.

3. Универсальность, т. е. наличие в БД всех необходимых данных и возможности доступа к ним в процессе решения проектной задачи.

4. Открытость БД для внесения в нее новой информации.

5. Наличие языков высокого уровня взаимодействия пользователей с БД.

6. Секретность, т. е. невозможность несанкционированного доступа к информации и ее изменений.

7. Оптимизация организации БД — минимизация избыточности данных.

Одним из принципов построения САПР является информационная согласованность частей ее программного обеспечения, т. е. пригодность результатов выполнения одной проектной процедуры для использования другой проектной процедурой без их трудоемкого ручного преобразования пользователем. Отсюда вытекают следующие условия информационной согласованности:

использование программами одной и той же подсистемы САПР единой БД;

использование единого внутреннего языка для представления данных.

Комплексная автоматизация процесса проектирования объекта предполагает информационную согласованность не только отдельных программ подсистем САПР, но и самих подсистем между собой. Способом достижения этой согласованности является единство информационного обеспечения.

Основные способы информационного согласования подсистем САПР достигаются либо созданием единой БД, либо сопряжением нескольких БД с помощью специальных программ, которые перекодируют информацию, приводя ее к требуемому виду.

Части ПО и методы, осуществляющие управление базами данных, составляют *систему управления базами данных* (СУБД). На рис. 3.2 показан пример использования СУБД и БД для проектирования ЭВА. Система управления БД позволяет получить доступ к интегрированным данным и

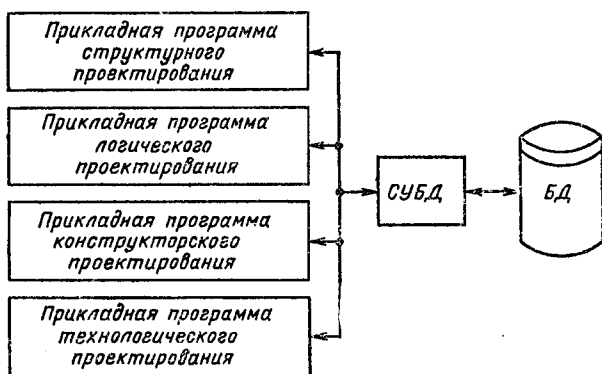


Рис. 3.2. Пример использования СУБД и БД для проектирования ЭВА

допускает множество различных представлений о хранимых данных. Приведем основные определения СУБД, сформулируем требования, которым они должны удовлетворять, и опишем функции, выполняемые СУБД.

Программное обеспечение, которое позволяет прикладным программам работать с БД без знания конкретного способа размещения данных в памяти ЭВМ, называют СУБД. Система управления БД выступает как совокупность программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Система управления БД должна обеспечивать простоту физической реализации БД; возможность централизованного и децентрализованного управления БД; минимизацию избыточности хранимых данных; предоставление пользователю по запросам непротиворечивой информации; простоту разработки, ведение и совершенствование прикладных программ; выполнение различных функций.

Система управления БД реализует два интерфейса: 1) между логическими структурами данных в программах и в БД; 2) между логической и физической структурами БД.

Опишем последовательность работы СУБД в одном из режимов: 1) программа запрашивает возможность чтения данных у СУБД, она передает необходимую информацию о программисте, типе записи и т. п.; 2) программа осуществляет поиск описания данных, на которые выдан запрос; 3) определяет, какого типа логические и физические записи необходимы; 4) выдает ОС запрос на чтение требуемой записи; 5) ОС взаимодействует с физической памятью; 6) записывает запрошенные данные в системные буферы; 7) выделяет требуемую логическую запись, выполняя необходимые преобразования; 8) передает данные из системных буферов в программу пользователя, а затем программе пользователя информацию о результатах выполнения запроса); 9) прикладная программа обрабатывает полученные данные.

*Распределенная БД* — это БД, в которой данные размещены по месту возникновения или наиболее эффективного использования. Она предполагает, что на каждой ЭВМ данные управляются локальными СУБД.

Для работы с распределенными БД создаются специальные системы управления распределенными БД (СУРБД). Основное требование к СУРБД такое же, как и



к СУБД: обеспечение максимальной независимости прикладных программ САПР от локализации данных в сети. Основой СУРБД являются каталоги, в которых хранится информация о структуре сети, описание локальных СУРБД и т. п. Особенность управления распределенными БД определяется разнотипностью локальных СУРБД, которые желательно иметь одного типа. Если же разные части распределенной БД управляются разными СУБД, то используют некоторую промежуточную интерфейсную СУРБД, через которую на основе соответствующих отображений обеспечивается взаимодействие локальных СУРБД.

### 3.2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ. ЯЗЫКИ БАЗЫ ДАННЫХ

Процесс разработки структуры БД на основании требований пользователя называют проектированием БД (ПБД). Результатом ПБД являются структура БД, состоящая из логических и физических компонент, и руководство для прикладных программистов.

Развитие системы БД во времени называют жизненным циклом. Последний делится на стадии анализа, проектирования и эксплуатации.

Первая стадия включает в себя этапы формулирования и анализа требований, концептуального проектирования, проектирования реализации, физического проектирования.

Анализ требований является полностью неформализованным этапом. Его основная цель — обеспечить согласованность целей пользователей и представлений об информационных потоках.

Вторая стадия состоит из этапов реализации БД, анализа функционирования и поддержки, модификации и адаптации.

Концептуальное проектирование обеспечивает выбор структуры организации информации на основе объединения информационных требований пользователей.

Проектирование реализации (логическое проектирование) разделяют на две части: проектирование базы данных и проектирование программ. Результатом первой части является логическая структура БД. Результатом второй части считают функциональные описания программных модулей и наборы запросов к БД.

Физическое проектирование разделяют также на две части: выбор физической структуры БД и отладку про-

граммных модулей, полученных при проектировании программ. Результатом этапа является подготовка к эксплуатации БД.

На этапе реализации БД ставится задача разработки программ доступа к БД.

Этап анализа функционирования и поддержки обеспечивает статистическую обработку данных о функционировании системы. Восстановление БД и ее целостность после сбоев обеспечивает поддержка БД.

Этап модернизации и адаптации позволяет производить изменения, оптимизацию функционирования, модификацию программ.

Языки, используемые в БД, делят на языки описания данных (ЯОД) и языки манипулирования данными (ЯМД).

В общем случае ЯОД описывает различные типы записей, их имена и форматы, служит для определения: типов элементов данных, которые нужны в качестве ключей; отношений между записями или их частями и именования этих отношений; типа данных, которые используются в записях; диапазона их значений; числа элементов, их порядка и т. п.; секретности частей данных и режимов доступа к ним.

Различают три уровня абстракции для описания данных:

концептуальный (с позиции администратора);

реализации (с позиции прикладного программиста и пользователя);

физический (с позиции системного программиста).

На концептуальном уровне описывают объекты, атрибуты и значения данных. На уровне реализации имеют дело с записями, элементами данных и связями между записями. На физическом уровне оперируют блоками, указателями, данными переполнения, группировкой данных.

Обычно ЯМД дают возможность манипулирования данными без знания несущественных для программиста подробностей. Они могут реализоваться как расширение языков программирования общего назначения путем введения в них специальных операторов или путем реализации специального языка (например, DL/1, ЯМД CODASYL).

Например, рассмотрим список операторов ЯМД CODASYL:

OPEN — объявляет открытие файла данных, делая возможным доступ к нему для записи, чтения и т. д.;

CLOSE — объявляет закрытие файла данных;  
FIND — находит указанную запись в файле;  
GET — передает указанные элементы записи в программу;

MODIFY — изменяет значения определенных элементов записи;

INSERT — вставляет в файл новую запись;

DELETE — удаляет из файла указанную запись и разрушает ее связи;

STORE — устанавливает место для новой записи.

При работе с БД используются несколько типов языков: манипулирования данными; программирования; описания физической организации данных.

Языки программирования, используемые в БД, представляют собой распространенные языки типа ФОРТРАН, КОБОЛ и др.

Языки описания логических схем пользователя реализуются средствами описания данных языка прикладного программирования, средствами СУБД, специальным языком.

Наиболее широко распространен первый способ описания. Он имеет в основе операторы объявления (например, DECLARE в языке PL/I, STRUCT в СИ, type в ADA).

Например,  
type DATA is

record

D: день

M: месяц

Y: год

end record;

определяют логическую запись ДАТА, состоящую из трех полей: ДЕНЬ, МЕСЯЦ и ГОД.

Языки описания схем БД предназначены для администратора БД. С их помощью определяют глобальные описания данных.

Языки описания физической организации данных описывают физическую структуру размещения схемы на машинных носителях. С их помощью определяют методы доступа, предписывающие размещение данных на тех или иных носителях, и т. п.

Наибольшими преимуществами обладают специальные языки, так как не зависят от используемых языков программирования или технических средств. Следовательно, при переносе БД на другое ТО или смене языка програм-

мирования большинство описаний БД останется без изменения.

Процесс проектирования БД начинают с построения *концептуальной модели (КМ)*. Концептуальная модель состоит из описания объектов и их взаимосвязей без указания способов физического хранения. Построение КМ начинается с анализа данных об объектах и связях между ними, сбора информации о данных в существующих и возможных прикладных программах. Другими словами, КМ — это модель предметной области. Версия КМ, обеспечиваемая СУБД, называется *логической моделью (ЛМ)*. Подмножества ЛМ, которые выделяются для пользователей, называются *внешними моделями (подсхемами)*. Логическая модель отображается в физическую, которая отображает размещение данных и методы доступа. Физическую модель называют еще *внутренней*.

Внешние модели не связаны с используемыми ТС и методами доступа к БД. Они определяют первый уровень независимости данных. Второй уровень независимости данных связан с отсутствием изменений внешних моделей при изменении КМ. На рис. 3.3 показаны два уровня не-

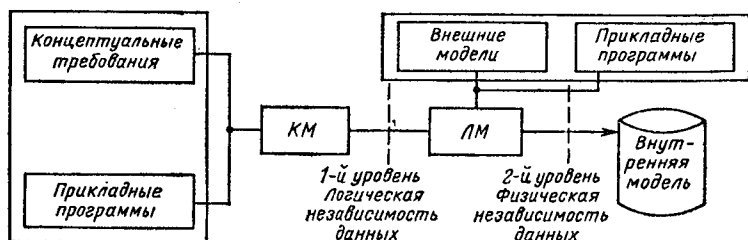


Рис. 3.3. Логическая и физическая независимость данных

зависимости данных в ИО САПР. В основном степень независимости данных определяется формализованностью проектирования БД.

Важным инструментом при разработке и проектировании БД является *словарь данных (СД)*, предназначенный для хранения сведений об объектах, атрибутах, значениях данных, взаимосвязях между ними, их источниках, значениях, форматах представления. Словарь данных позволяет получить однообразную и формализованную информацию

обо всех ресурсах данных. Главное назначение СД состоит в документировании данных. Он должен обеспечивать централизованное введение и управления данными, взаимодействие между разработчиками любого проекта, например, САПР. Существуют *интегрированные и независимые СД*. В первом случае СД — это часть пакета программ СУБД, а во втором — отдельный пакет программ в виде дополнения к СУБД.

В настоящее время СД рассматривают как связующее звено в системе ПО обработки данных, включающей в себя процессор, СУБД, языки запросов, монитор телеобработки. На рис. 3.4 показаны интерфейсы СД в гипотетической

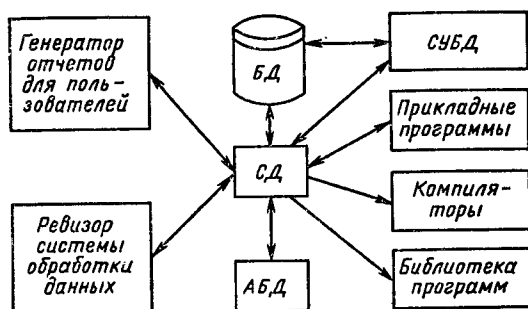


Рис. 3.4. Интерфейсные СД в гипотетической системе с БД

ской системе с БД. В полном объеме СД обязан: поддерживать КМ, логическую, внутреннюю и внешнюю модели; быть интегрированным с СУБД, поддерживать тестовые и рабочие версии хранимых описаний; обеспечивать эффективный обмен информацией с СУБД и процесс изменения рабочей версии при изменении БД. Словарь данных должен иметь свою БД. Приведем основные составляющие БД словаря данных. Последний включает в себя атрибут; объект; групповой элемент данных; выводимый объект данных; синонимы, т. е. атрибуты, имеющие одинаковое назначение, но различные идентификаторы; омонимы, т. е. атрибуты с различным назначением, но с одинаковыми идентификаторами; описание КМ, ЛМ, внешних и внутренних моделей; описание, позволяющее пользователям формально и однозначно выбирать атрибуты для решения задач.

По аналогии с СД создают информационный фонд САПР. Отметим, что он может являться частью СД. Содержимое информационного фонда САПР можно разделить на две основные части.

Первая часть содержит нормативно-справочную информацию, сведения о ГОСТ, ОСТ и т. п., а также различные модели компонентов (тепловые, электрические, коммутационные). Эта часть характеризуется многократным считыванием, сравнительно редкой обновляемостью. Иногда ее называют «постоянной» частью БД.

Например, информация БД САПР РАПИРА—5.3—82, относящаяся к постоянной части, включает в себя описания одиночных базовых элементов (ОБЭ), групповых базовых элементов (ГБЭ), таблиц технологических материалов, тепловых моделей элементов (ТМ).

Вторая часть содержит сведения об объектах, которые находятся в процессе проектирования. Они включают описания объектов после выполнения различных проектных операций (например, размещения, трассировки), описание конструкторско-технологической документации во внутренней форме САПР и т. п.

В САПР РАПИРА—5.3—82 к этой части относится информация, хранящаяся во временных архивах данных (ВАД). Можно выделить третью часть информации, в которой хранятся сведения о спроектированных изделиях (в САПР РАПИРА—5.3—82 это информация, занесенная в архив спроектированных изделий—АСИ).

При проектировании БД необходимо провести идентификацию основных объектов предметной области и прикладных программ, подлежащих использованию, определить объекты и их взаимосвязи, построить СД, КМ, ЛМ, ФМ с проведением анализа и оценок (рис. 3.5).

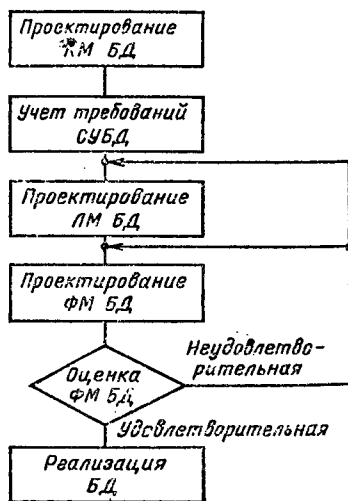


Рис. 3.5. Упрощенная структурная схема проектирования БД

Информацию, используемую в процессе проектирования, делят условно на две группы. К первой группе относят информацию, описывающую концептуальное структурное представление. Ее называют *ISP*-информацией. Она не связана с конкретными способами обработки и приложениями и описывает концептуальные связи в базе данных. Пример записи *ISP*-информации имеет вид:

1. Описание сущности	
Наименование	ЭВМ
Количество	10
2. Описание атрибута	
Наименование	Номер
Скорость вычислений	$5 \cdot 10^6$ оп/с
Объем оперативной памяти	1024 кбайт
Вероятность существования	1,0
3. Описание связи	
Наименование	—
Сущность	Работает на ВЦ универсальная ЭВМ
Отображение	1 : 1
Вероятность существования	1,0

Ко второй группе относят информацию, описывающую концептуальное представление. Ее называют *UP*-информацией. Она определяет требование организации к обработке данных и описывает данные и связи, используемые в приложениях.

Пример записи *UP*-информации имеет вид:

1. Описание процесса	
Наименование	Компоновка
Частота применения	Еженедельно
Вероятность применения	1,0
Приоритет	Высший
Требуемые данные	Схема, правила проектирования, ГОСТ
Объем данных	100 схем
2. Оператор	
Оператор	Разбиение
Критерий поиска	Минимум внешних связей
Вероятность события	1,0

Считают, что *ISP*-информация должна обеспечивать гибкость и адаптивность, *UP*-информация — эффективность проектирования. *ISP*-Информацию в основном используют для построения начальной информационной структуры, *UP*-информацию — для ее улучшения и уточнения в конкретном приложении.

После определения физической структуры ее описание передается СУБД. До использования БД реализуют ее

макет или строят прототип, с помощью которых получают оценку характеристик БД и прогнозируют ее развитие. Если прототип не отвечает заданным требованиям, производят перепроектирование БД, в положительном случае — загрузку БД. Отметим, что построение прототипов, макетов и моделей реальной БД является сложной задачей из-за отсутствия математических методов оценки правильности модели.

При конвертировании и интеграции данных обеспечиваются возможность изменения физической структуры БД и поддержка разработанных прикладных программ управления БД.

При эксплуатации обеспечение безопасности, секретности, разграничения доступа к БД поддерживается определенными процедурами. Кроме того, вводят процедуры восстановления и повторного запуска.

### 3.3. МОДЕЛИ ДАННЫХ

Современные СУБД основываются на использовании моделей данных (МД), позволяющих описывать объекты предметных областей и взаимосвязи между ними. Существуют три основные МД и их комбинации, на которых основываются СУБД: *реляционная модель данных* (РМД), *сетевая модель данных* (СМД), *иерархическая модель данных* (ИМД).

Основное различие между этими МД состоит в способах описания взаимодействий между объектами и атрибутами. Взаимосвязь выражает отношение между множествами данных. Используют взаимосвязи «один к другому», «один ко многим» и «многие ко многим». «Один к одному» — это взаимно однозначное соответствие, которое устанавливается между одним объектом и одним атрибутом. Например, в определенный момент времени в одной ЭВМ используется один определенный процессор. Номеру выбранной ЭВМ соответствует номер выбранного процессора. «Один ко многим» — это соответствие между одним объектом и многими атрибутами. «Многие ко многим» — это соответствие между многими объектами и многими атрибутами. Например, на множество ЭВМ может одновременно работать множество пользователей. Взаимосвязи между объектами и атрибутами удобно представлять в виде графов и гиперграфов.

**Реляционная модель данных.** В РМД объекты и взаи-



мосвязи между ними представляют в виде таблиц. Таблица, состоящая из строк и столбцов, называется отношением. Каждый столбец в таблице является атрибутом. Строки таблицы являются кортежами, т. е. упорядоченными множествами. Значения в столбце определяют из множества значений, которые принимает атрибут. Столбцы таблицы — это элементы данных, а строки — записи. Построим, например, таблицу для представления данных с помощью РМД (табл. 3.1).

Таблица 3.1

Номер	Этапы проектирования (имя)	Атрибуты	
		Сложность	Время реализации
1	Покрытие	$O(n^3)$	минуты
2	Типизация	$O(n^2)$	секунды
3	Компоновка	$O(n^3)$	минуты
4	Размещение	$O(n^3)$	минуты
5	Трассировка	$O(n^3 - n^4)$	часы
6	Контроль	$O(n^3)$	минуты

Первичным ключом в таблице является номер этапа проектирования. Таблица имеет два атрибута и шесть кортежей.

Приведем еще один пример представления данных с помощью таблицы ПОЛЬЗОВАТЕЛЬ (табл. 3.2).

Таблица 3.2

Идентификационный номер	Фамилия
236	Сидоров
185	Иванов
349	Петров
499	Кочетков
777	Харламов

Эта таблица представляет данные с помощью реляционной модели. Первичным ключом является идентификационный номер пользователя.

Основное достоинство реляционного подхода — его простота и доступность. Пользователи абстрагированы от физической структуры памяти. Это позволяет эксплуатиро-

вать БД без знания методов и способов ее построения. Основные достоинства РМД следующие: простота, независимость данных; гибкость; непроцедурные запросы, теоретическое обоснование на основе теории отношений.

Основные недостатки РМД: низкая производительность по сравнению с ИМД и СМД, сложность ПО, избыточность. Кроме того, проблематично обеспечение полного набора функциональных возможностей и необходимых операционных характеристик при обработке больших БД.

**Иерархическая модель данных.** Она основана на понятии деревьев, состоящих из вершин и ребер. Вершина дерева ставится в соответствие совокупности атрибутов данных, характеризующих некоторый объект. Вершины и ребра дерева как бы образуют иерархическую древовидную структуру (ИДС), состоящую из  $n$  уровней (рис. 3.6).

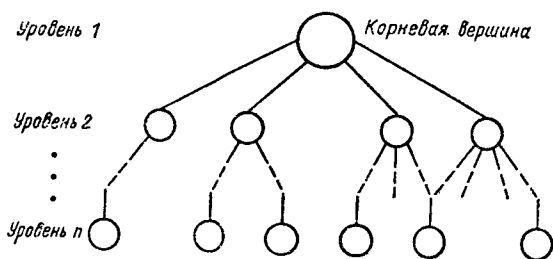


Рис. 3.6. Иерархическая древовидная структура

Первую вершину в дереве называют корневой вершиной ИДС. Она удовлетворяет семи условиям: 1. Иерархия начинается с корневой вершины. 2. Каждая вершина соответствует одному или нескольким атрибутам. 3. На уровнях с большим номером находятся зависимые вершины. Вершина предшествующего уровня является начальной для новых зависимых вершин. 4. Каждая вершина, находящаяся на уровне  $i$ , соединена с одной и только одной вершиной уровня  $i-1$ , за исключением корневой вершины. 5. Корневая вершина может быть связана с одной или несколькими зависимыми вершинами. 6. Доступ к каждой вершине происходит через корневую по единственному пути. 7. Существует произвольное количество вершин каждого уровня.

Иерархическая модель данных состоит из нескольких деревьев, т. е. является лесом. Каждая корневая вершина

образует начало записи логической базы данных. В ИМД вершины, находящиеся на уровне  $i$ , называют порожденными вершинами на уровне  $i-1$ . Ниже приведен пример представления информации в ИМД, реализующей отношение «один ко многим»:

	ПОЛЬЗОВАТЕЛЬ	Корневая вершина							
Уровень 1	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; border-right: 1px solid black; padding: 5px;">номер</td> <td style="width: 33%; border-right: 1px solid black; padding: 5px;">фамилия</td> <td style="padding: 5px;">отдел</td> </tr> </table>	номер	фамилия	отдел					
номер	фамилия	отдел							
	ЭВМ	ОПЕРАЦИЯ+РЕЗУЛЬТАТ	Порожденная вершина						
Уровень 2	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 16.6%; border-right: 1px solid black; padding: 5px;">номер ЭВМ</td> <td style="width: 16.6%; border-right: 1px solid black; padding: 5px;">название</td> <td style="width: 16.6%; border-right: 1px solid black; padding: 5px;">дата операции</td> <td style="width: 16.6%; border-right: 1px solid black; padding: 5px;">опреде- ление</td> <td style="width: 16.6%; border-right: 1px solid black; padding: 5px;">результ- тат</td> <td style="padding: 5px;">рекомен- дации</td> </tr> </table>	номер ЭВМ	название	дата операции	опреде- ление	результ- тат	рекомен- дации		
номер ЭВМ	название	дата операции	опреде- ление	результ- тат	рекомен- дации				

Для каждого пользователя может иметься экземпляр корневой вершины. Иерархическая модель данных позволяет для каждого пользователя получать представление о нескольких операциях и нескольких ЭВМ. ПОЛЬЗОВАТЕЛЬ соответствует корневой вершине и находится на более высоком уровне иерархии, чем ЭВМ, ОПЕРАЦИЯ и РЕЗУЛЬТАТ.

Отметим, что выбор ИМД осуществляет администратор БД на основе операционных характеристик. Введение двух ИМД, связанных между собой, позволяет решать вопросы включения и удаления данных. Основные достоинства ИМД — простота построения и использования, обеспечение определенного уровня независимости данных, наличие существующих СУБД, простота оценки операционных характеристик. Основные недостатки: отношение «многие ко многим» реализуется очень сложно, дает громоздкую структуру и требует хранения избыточных данных, что особенно нежелательно на физическом уровне; иерархическая упорядоченность усложняет операции удаления и включения; доступ к любой вершине возможен только через корневую, что увеличивает время доступа.

**Сетевые модели данных.** В СМД элементарные данные и отношения между ними представляются в виде ориентированной сети (вершины — данные, дуги — отношения). База данных, описываемая сетевой моделью, состоит из нескольких областей. Область содержит записи. Одна запись состоит из нескольких полей. Набор, состоящий из запи-

сей, может размещаться в одной или нескольких областях (рис. 3.7). В СМД объекты предметной области объединяются в сеть. Графически сетевая модель описывается прямоугольниками и стрелками. Каждый тип записи может содержать множество атрибутов. На рис. 3.8 показан пример представления области в СМД. Здесь область — это часть БД, в которой располагаются записи, наборы и части наборов. Стрелками соединены несколько типов запи-

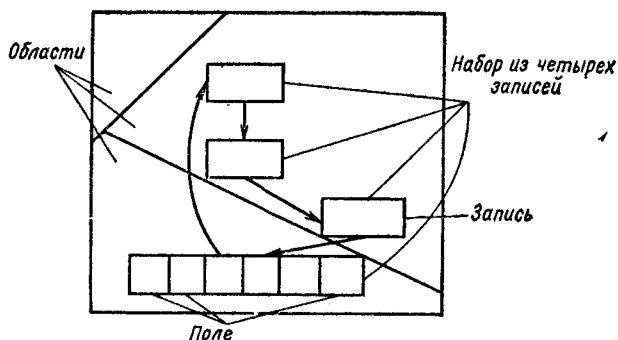


Рис. 3.7. База данных, описываемая сетевой моделью

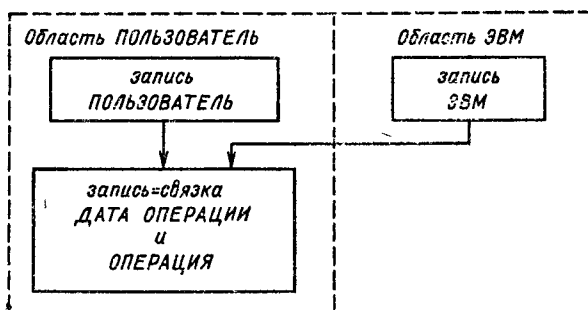


Рис. 3.8. Область в СМД

сей, изображающих типы набора. Тип набора представляет логическую взаимосвязь «один ко многим». На примере рис. 3.9 показан экземпляр набора в СМД: ПОЛЬЗОВАТЕЛЬ — ВЫПОЛНИЛ — ПРОЦЕДУРУ. Каждый экземпляр набора ПОЛЬЗОВАТЕЛЬ — ВЫПОЛНИЛ — ПРОЦЕДУРУ реализует иерархическую связь между поль-

зователем и операцией. Важное отличие СМД от ИМД состоит в том, что в СМД каждая запись может быть в любом числе наборов и может находиться как на верхнем, так и на нижнем иерархическом уровне. Следовательно, любая запись может быть задана как точка входа. Суще-

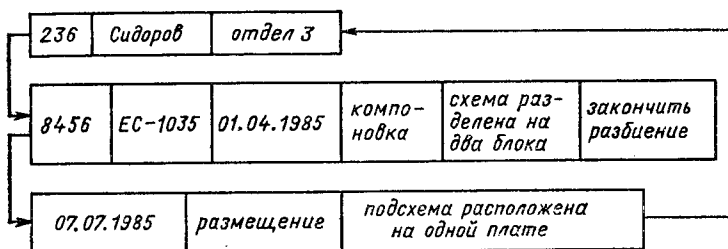


Рис. 3.9. Экземпляр набора в СМД

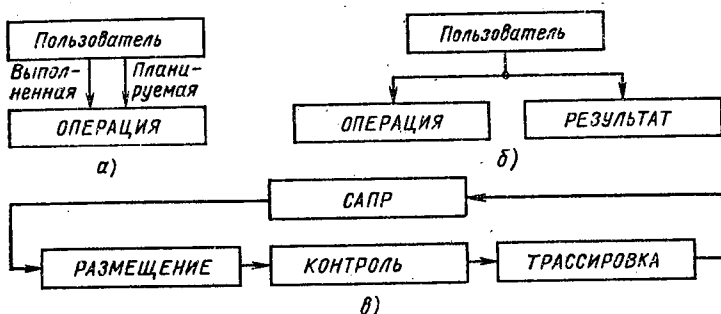


Рис. 3.10. Три типа наборов

ствует три типа наборов. Первый тип (рис. 3.10, а) образовывается типами записей ПОЛЬЗОВАТЕЛЬ и ОПЕРАЦИЯ. Второй тип (рис. 3.10, б) состоит из трех и более записей и называется многочисленным. Третий тип (рис. 3.10, в) представляется сингулярными наборами и объединяет записи, у которых нет корневой вершины, но которые могут ее иметь впоследствии. Основные достоинства СМД — наличие реализованных СУБД, обеспечивающих эту модель, простота реализации отношений «многие ко многим». Основной недостаток СМД — ее сложность. При реорганизации БД возможна потеря независимости данных.

При проектировании БД первым этапом, как отмечалось, является проектирование или построение КМ предметной области. Здесь выполняют структуризацию данных, определяют связи между ними, не учитывая особенностей реализации. Первым этапом построения КМ является анализ данных. При этом собирают информацию о данных, которые используются в имеющихся прикладных программах. В процессе сбора данных определяют имена объектов и элементов данных, описаний, атрибутов, источников, оценки, сложность, важность, отношения связности между элементами и объектами, продолжительность и способы хранения данных. Далее на основе анкетирования проводят анализ организации хранения данных и исследуют документооборот от источника к пользователю. После этого приступают к разработке КМ БД. Первоначально АБД собирает информацию о всех данных для прогнозирования и перспективных исследований. Концептуальная модель БД является основой для ЛМ, которая реализуется средствами реляционной, иерархической или сетевой СУБД. При разработке КМ используют нормализацию отношений, т. е. группируют элементы данных по свойствам модификации, включения и удаления данных. Концептуальная модель может быть также представлена в виде графов.

Рассмотрим проектирование ЛМ БД. Первоначально определяется модель данных, подходящих для отображения КМ. Отображение может выполняться РМД, ИМД, СМД. Пусть задана КМ (рис. 3.11, а), которую необходимо отобразить в логическую модель. Основные объекты — ПОЛЬЗОВАТЕЛЬ, ЭВМ, САПР. Первичные ключи объектов содержат один атрибут. Например, ключом объекта ПОЛЬЗОВАТЕЛЬ будет НОМ — ПОЛЬЗОВАТЕЛЯ (номер пользователя). Рассмотрим отображение КМ (рис. 3.11) на РМД. Для отображения КМ на ЛМ требуется определить отношения и их атрибуты. Построим, например, таблицу ПОЛЬЗОВАТЕЛЬ (табл. 3.3).

Т а б л и ц а 3.3

Номер	Фамилия	Статус	Приоритет	Отдел
125	Сидоров	инженер	2	4
799	Михайлов	аспирант	1	7
333	Иванов	студент	3	8

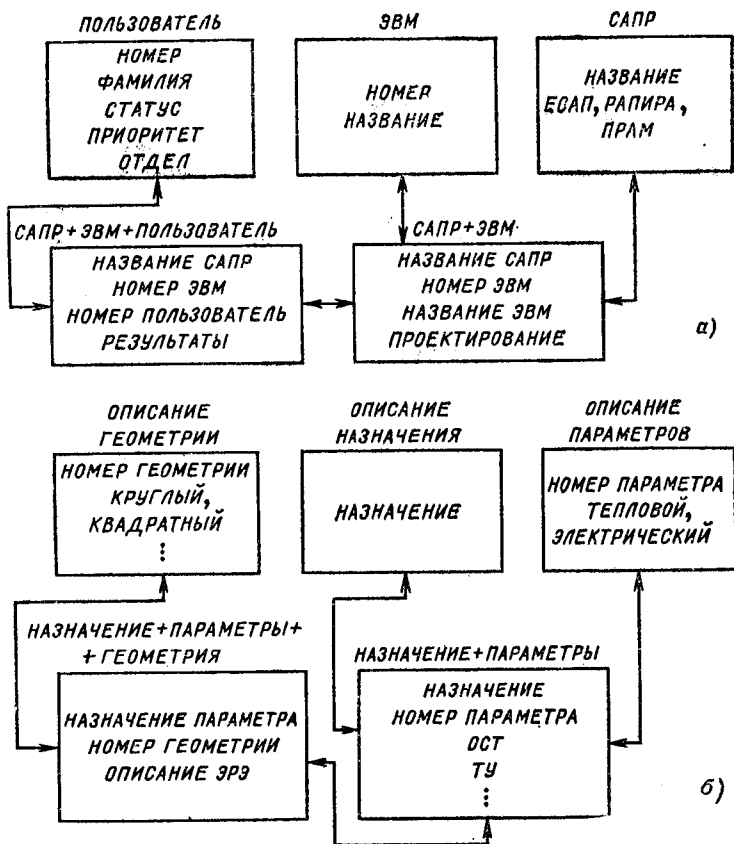


Рис. 3.11. Концептуальные модели

Ключевым атрибутом является НОМ — ПОЛЬЗОВАТЕЛЬ, неключевыми атрибутами являются фамилия, статус, приоритет, отдел. Аналогичным образом строятся отношения ЭВМ, САПР, ЭВМ+САПР, ПОЛЬЗОВАТЕЛЬ+ЭВМ+САПР. Из примера видно, что отображение КМ на РМД является простой процедурой. Отображение КМ на ИМД и СМД сложнее и выполняется в соответствии с описанными приемами построения ИМД и СМД.

Исходными данными для отображения КМ и СМД служат заданный набор отношений и требования оптимизации. Процесс проектирования состоит из следующих стадий:

разработка средства описания входных файлов и функциональных зависимостей и преобразование исходных сетей в иерархические файлы;

преобразование ненормализованных отношений в нормализованные;

построение сетевой структуры по заданному множеству нормализованных отношений;

оптимизация сетевой структуры для повышения эффективности работы.

Для решения задач первой стадии выполняют преобразование файлов с сетевой структурой в древовидную структуру. Для этого строят набор деревьев, покрывающих все связи данных в сети.

На второй стадии процесс нормализации выполняют с помощью группировки элементов данных по заданным правилам.

На третьей стадии строят базовую сетевую структуру, которая будет сохранять все взаимосвязи данных, которые представляются в отношениях. При этом проявляются зависимости между данными в различных отношениях.

На четвертой стадии анализируют эффект введения в сеть дополнительных записей и путей доступа. Для этого учитывают частоту запросов в систему и анализируют способы уменьшения времени доступа к данным: разбиение записей, создание нового пути доступа и т. д.

Рассмотрим пример отображения концептуальной схемы на иерархическую и сетевую модели данных. На рис. 3.11, б изображена концептуальная модель «Электрорадиоэлемент». Основные объекты ГЕОМЕТРИЯ, НАЗНАЧЕНИЕ, ПАРАМЕТР. Внутри объектов приведены атрибуты и их значения.

Основные этапы отображения на иерархическую модель данных: 1) вывод обобщенной модели, не учитывающей ограничение СУБД; 2) преобразование полученной модели с учетом СУБД и ее оптимизация; 3) упрощение имен ключей; 4) реализация неявно присутствующих взаимосвязей. Один из возможных результатов отображения изображен на рис. 3.12, а.

При отображении концептуальной схемы на сетевую модель данных основные этапы остаются такими же.

#### 3.4. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ

Производительность программ, взаимодействующих с БД, зависит от способа хранения и доступа к данным. Отметим, что увеличение числа операций ввода/вывода при



ИЕРАРХИЧЕСКОЕ ДЕРЕВО (ГЕОМЕТРИЯ, НАЗНАЧЕНИЕ, ОПИСАНИЕ ЭРЭ)

ОПИСАНИЕ ГЕОМЕТРИИ

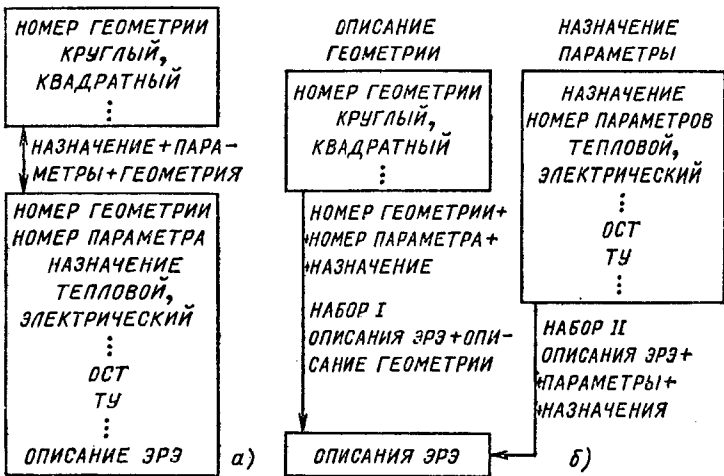
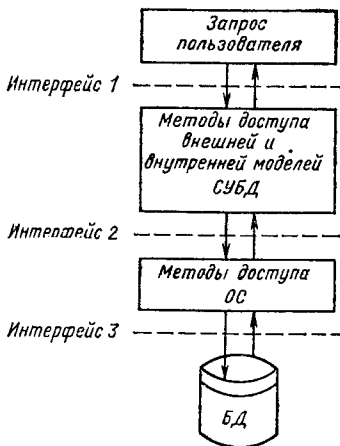


Рис. 3.12. Результаты отображения концептуальной схемы на иерархическую (а) и сетевую (б) модели данных

выборке данных приводит к уменьшению производительности. Выполнение запроса пользователя обеспечивается СУБД, причем в процессе выполнения запрос проходит несколько интерфейсов. Пример выполнения запроса пользователя показан на рис. 3.13.



**Интерфейс 1.** По запросу пользователя определяют, к какой физической БД осуществляется доступ.

**Интерфейс 2.** Система управления БД использует методы доступа внутренней модели, которые в разных системах выполняются различно.

**Интерфейс 3.** Методы доступа внутренней модели и методы доступа ОС осуществляют доступ к записям фи-

Рис. 3.13. Пример выполнения запроса пользователя

зической БД и позволяют найти необходимые записи.

Методы доступа ОС, внешней и внутренней моделей выбирают данные из физических БД и передают их СУБД. Последняя определяет, что передать пользователю, в каком виде и формате.

Производительность БД зависит от методов доступа внешней и внутренней моделей.

Рассмотрим сначала методы доступа внутренней модели: 1) физический, последовательный; 2) индексно-последовательный; 3) индексно-произвольный; 4) инвертированный; 5) прямой; 6) хеширования.

Для указанных методов известны два критерия эффективности: доступа — величина, обратная среднему числу физических обращений для выборки запроса конкретной записи базы данных; хранения — величина, обратная среднему числу байтов поля вторичной памяти, необходимого для хранения одного байта исходных данных.

Первый метод доступа хранит физические записи в логической последовательности. Эффективность доступа очень низкая, так как для выборки необходимой записи просматриваются в худшем случае все предшествующие БД. Число записей, к которым нужно осуществлять доступ, составляет в среднем  $1/2$  общего числа записей БД. Когда используют устройства прямого доступа, каждой физической записи система отводит следующий участок физической памяти. В основном этот метод применяют для восстановления, хранения и выборки данных. Эффективность использования памяти стремится к 100 %.

В основе второго метода доступа лежит способ создания отдельного файла. Такой файл называют индексным, и скорость поиска в нем высокая. Индексный файл упорядочивается по первичному ключу — основному атрибуту физической записи. По значению ключа идентифицируется физическая запись. На рис. 3.14, а показан пример индексно-последовательного метода доступа. Последовательная организация индексного файла допускает индексацию его содержимого. Записи индекса группируются в блоки, которые также можно индексировать. Для очень больших файлов строят несколько индексных файлов, причем индексный файл  $i$ -го уровня содержит указатели на индексный файл  $i+1$ -го уровня. Метод применяют для хранения и выборки данных. Эффективность доступа зависит от числа уровней индексации, распределения памяти для размещения индексов, числа записей БД и уровня переполнения.

Эффективность хранения зависит от размеров и применяемости БД.

При индексно-произвольном методе доступа записи хранятся в произвольном порядке. Создается отдельный файл статей, включающих значения действительного ключа и физического адреса хранимых записей. Весь файл назы-

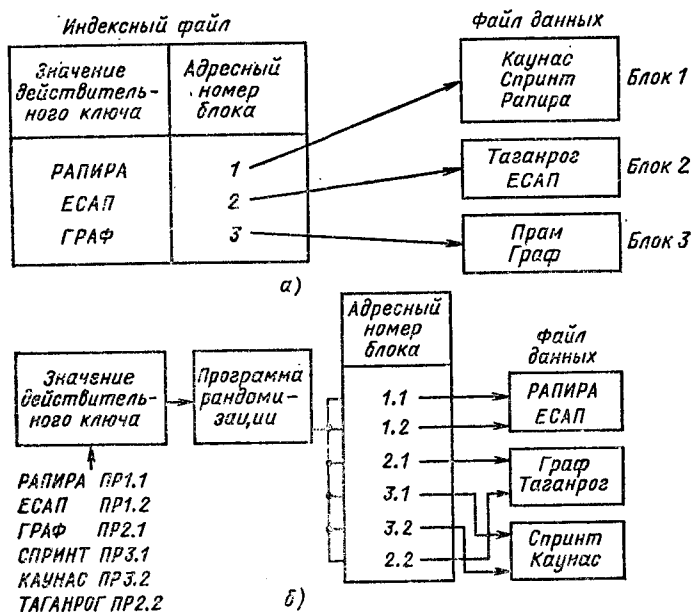


Рис. 3.14. Индексно-последовательный (а) и индексно-произвольный (б) методы доступа

вают индексом. Различают два основных способа создания индекса и доступа к нему. Первый заключается в применении для ведения схемы упорядочения, причем доступ к индексу последовательный. Во втором способе индекс упорядочивается, но доступ к нему произвольный. На рис. 3.14, б показан пример этого способа. Здесь значение ключа обрабатывается подпрограммой рандомизации (ПР), что позволяет определить адрес записи в файле данных. Между адресом и физической записью существует взаимно однозначное соответствие. Метод позволяет создавать индексный файл на любом поле записи. Его используют для

выборки и для заполнения данных. Если индекс не упорядочивается, то только с его помощью обработать последовательно записи в соответствии с порядком индекса невозможно. Так как каждой записи соответствует статья индексного файла, то объем может быть недопустимо большой.

Инвертированный метод доступа применяется только для поиска. Каждому инвертированному полю соответствует статья в таблице: имя поля, значение и адрес записи. После загрузки БД выполняется упорядочивание статей по имени поля и групп статей с общим полем — по значению поля. На рис. 3.15 показан пример инвертированного

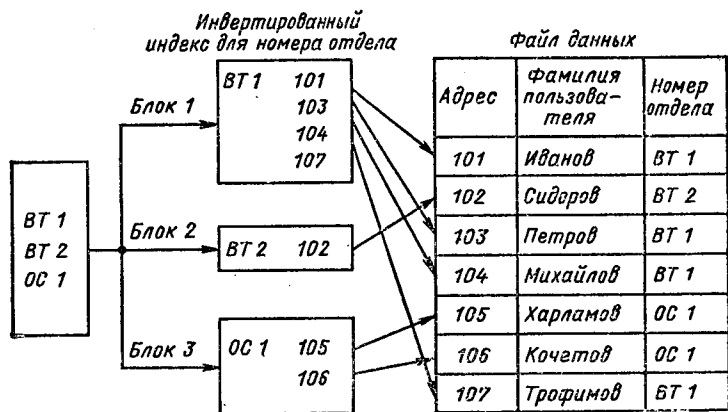


Рис. 3.15. Инвертированный метод доступа

метода доступа. Здесь файл данных инвертируется относительно поля «номер отдела». Каждому инвертированному полю соответствует статья в таблице. Для любого поля можно построить отдельный индекс. Эффективность доступа зависит от числа записей БД, уровней индексации и распределения памяти для размещения и в общем случае меньше 0,5. Эффективность может быть повышена, если инвертированный индекс постоянно находится в памяти.

Прямой метод доступа устанавливает взаимно однозначное соответствие между ключом записи и ее физическим адресом; не требует упорядочения значений ключей физических записей; применяется как для хранения, так и для поиска. Эффективность доступа всегда равна единице, а

эффективность хранения зависит от плотности ключей. При низкой плотности память расходуется впустую, так как резервирует адреса, соответствующие отсутствующим ключам.

Метод доступа с хешированием основан на алгоритмическом определении адресов физической записи по значениям ключей. Метод в отличие от прямого доступа допускает отображение многих ключей в один адрес. Алгоритм преобразования ключа в адрес называют алгоритмом хеширования или рандомизации. Одинаковые ключи преобразуются в одинаковые адреса. При использовании алгоритмов хеширования необходимо учитывать несоответствие порядка хранения физических записей порядку исходных ключей. Ниже показан пример метода доступа с помощью хеширования.

Исходный ключ	Преобразованный ключ	Адрес	Содержимое записи	Указатель цепочки	Адрес	Содержимое записи	Указатель цепочки
Иванов	101	101	Иванов	0			
Сидоров	102	102	Сидоров	0	107	Трофимов	108
Петров	103	103	Петров	106	108	Цветков	0
Михайлов	104	104	Михайлов				
Кочетов	103	106	Кочетов	107			
Трофимов	103						
Цветков	103						
				Основная область хранения		Область переполнения	

Здесь значение исходного ключа «Петров» преобразуется в адрес первой записи цепочки 103. Запись Петров содержит указатель на адрес 106, где размещается запись Кочетов, и т. д. Запись Цветков в цепочке имеет нулевой указатель. Эффективность доступа зависит от распределения исходных ключей, алгоритма их преобразования и распределения памяти. Эффективность хранения зависит от распределения ключей и алгоритма их преобразования. Метод не требует логической упорядоченности значений ключей физических записей.

Рассмотренные методы доступа внутренней модели позволяют войти в БД. Методы внешних моделей производят дальнейший поиск записей БД. По взаимосвязям между записями методы доступа внешней модели осуществляют

их хранение или поиск. Для определения записи можно использовать несколько методов доступа внешней модели, а для хранения только один.

Рассмотрим возможные взаимосвязи между записями.

1. Взаимосвязь полей упорядочения. Здесь поле упорядочения ключа записи  $Y$  следует за полем упорядочения ключа записи  $X$ .

2. Взаимосвязь по подчиненности. Здесь запись  $Y$  может быть подчиненной по отношению к записи  $X$ .

3. Взаимосвязь по порождению. Это запись, обратная 2.

4. Взаимосвязь «соседство». Определяет хранение записи  $Y$  рядом с записью  $X$ .

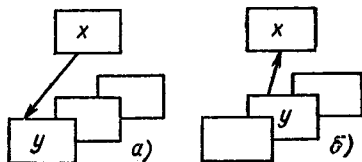
На основе взаимосвязей между записями существуют четыре типа доступа внешней модели.

1. Методы доступа с использованием полей упорядочения. Здесь подлежащая хранению запись  $Y$  — это запись, поле упорядочения которой идет за полем упорядочения записи  $X$ . На основе полей упорядочения можно реализовать методы доступа «к предшествующему» и «к следующему». Например, в иерархической модели данных можно включить либо найти экземпляр узла  $Y$ , предшествующей экземпляру узла  $X$ .

2. Метод доступа по подчиненности. Используют для хранения и поиска подчиненных вершин для иерархической модели данных. В этом методе способ включения или поиска вершин (записи) зависит от способа упорядочения, ограничений на доступ.

3. Метод доступа к исходному или владельцу. Противоположен методу доступа по подчиненности. На рис. 3.16

Рис. 3.16. Примеры методов доступа по подчиненности (а) и к исходному (б)



показаны примеры 2-го и 3-го методов доступа соответственно.

4. Метод доступа «соседство». Предназначен только для хранения данных. Например, в ИМД порожденная вершина должна храниться рядом с исходной с учетом заданной иерархией взаимосвязи.

Методы доступа внешней модели реализуются с по-

мощью указателей (рис. 3.17): прямого (а), относительного (б) и символического (в). Прямой указатель включает действительный адрес блока на диске, содержащего рассматриваемую запись Y, и хранится в записи X, указываю-

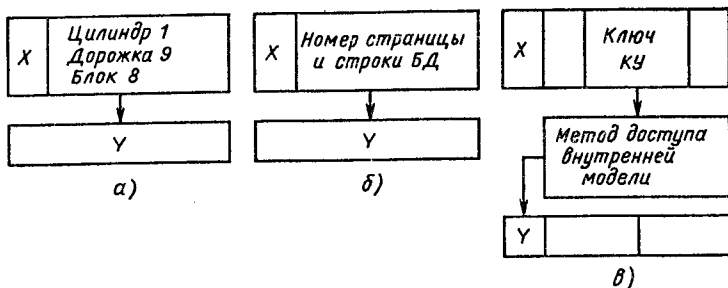


Рис. 3.17. Указатели прямой (а), относительный (б) и символический (в)

щей на запись Y. В прямом указателе — наилучшая эффективность доступа. Но он зависит от устройства, и при пересылке записей необходимо обновление всех связанных с записью прямых указателей.

Относительный указатель — это идентификатор, отображаемый далее в действительный адрес блока на диске. Преимущество этого указателя в том, что при пересылке файлов СУБД должно быть известно только размещение начального адреса. В качестве недостатка отметим, что он основан на заранее известной позиции.

В символическом указателе ключ рассматриваемой записи Y хранится в записи X. Поиск записи осуществляется методом доступа внутренней модели. Основное преимущество состоит в том, что символический указатель не зависит от устройства и при пересылке нет необходимости обновления указателей. В качестве недостатка отметим, что он длиннее других указателей и имеет в общем случае худшую эффективность доступа.

Хранение и выборка записей из БД в общем случае выполняются СУБД на основе сочетания методов доступа внутренней и внешних моделей. Современные СУБД используют сложные методы доступа в разнообразных сочетаниях. Например, СУБД IMS применяют для внутренней модели иерархические методы доступа: последовательный, индексно-последовательный, индексно-прямой, прямой.

Методы доступа внешней модели дают возможность доступа по физической иерархии, по модифицированной логическими взаимосвязями иерархии (к логически исходному, к физически подобному, к физически исходному и т. п.) и логический последовательный.

Рассмотрим в качестве примера иерархическую последовательный метод доступа (рис. 3.18). При этом методе вза-

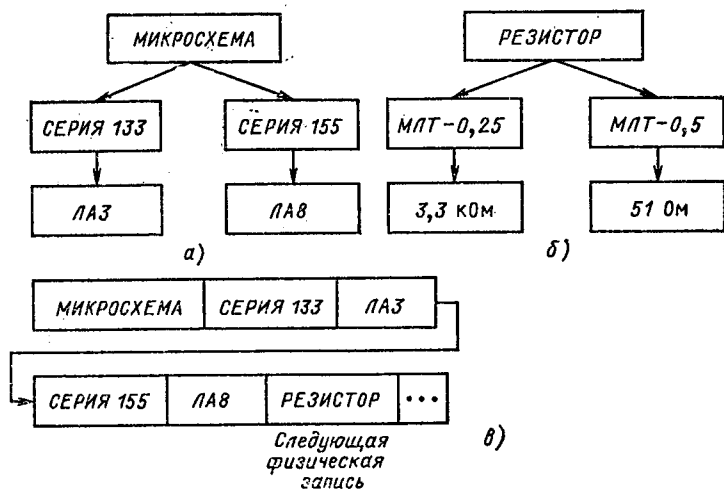


Рис. 3.18. Иерархический последовательный метод доступа:

а, б — структуры двух физических записей; в — порядок следования сегментов

имосвязи между сегментами иерархии отражаются в последовательной организации (сверху вниз, слева направо). При этом может использоваться только физический последовательный метод доступа.

### 3.5. ПОСТРОЕНИЕ ФИЗИЧЕСКОЙ МОДЕЛИ БАЗЫ ДАННЫХ

Проектирование физической БД — это процесс создания реализуемой структуры БД на основе логической структуры и требований пользователя. Опишем этапы создания физической модели БД для выбранной СУБД и оценки ее эксплуатационных характеристик.

Физическая модель — это структура БД, подлежащая хранению на различных физических устройствах. При проектировании БД оптимизируют ФМ по затратам времени и памяти. Очевидно, что введением избыточности данных



уменьшается число операций физического ввода/вывода, а отсутствие избыточности увеличивает временные затраты, но сохраняет память.

К понятию физической организации БД относят: тип внешней памяти, расположение данных на внешних носителях, способы адресации и поиска данных, типы используемого оборудования и т. п.

Физическая организация данных должна обеспечивать физическую независимость данных, т. е. возможность изменения способа хранения данных и соответствующих технических средств без изменения программного обеспечения. Факторы, влияющие на выбор физической организации БД:

1. Требование экономии памяти, удовлетворяется за счет использования блокирования записей, сжатия данных в более компактные записи.

2. Необходимость минимизации избыточности данных.

3. Согласование способа хранения отдельных данных и частоты их использования и стремление к записи наиболее часто применяемых данных на устройства с большим быстродействием.

4. Способность физической организации БД к расширению.

5. Обеспечение готовности системы к работе возможно большую часть времени.

6. Наличие средств восстановления информации при ее разрушениях.

7. Изолирование физической организации БД от логической с целью обеспечения возможности перераспределения данных на физических носителях для повышения эффективности работы.

Ниже приведены технические характеристики ряда устройств хранения информации:

Накопители на сменных магнитных дисках	ЕС-5061	ЕС-5066
Максимальный объем информации, байт . . . . .	$29 \cdot 10^6$	$100 \cdot 10^6$
Частота вращения пакета дисков, $s^{-1}$ . . . . .	2400	3600
Число магнитных головок . . . . .	20	20
Скорость передачи данных, кбайт/с . . . . .	312	806
Время доступа, мс:		
максимальное . . . . .	80	80
среднее . . . . .	62	55
Максимальная плотность информации, бит/мм . . . . .	90	160

<b>Накопители на гибких магнитных дисках</b>			
	<b>СМ-5601</b>	<b>СМ-5602</b>	<b>СМ-5608</b>
Полная емкость, Мбит . . .	3,2	12,8	8,2
Количество дисков, шт. . .	1	2	1
Наличие устройства управления	Нет	Есть	Есть
Габаритные размеры, мм . . .	134×217× ×375	310×483× ×790	354×483× ×766
Потребляемая мощность, кв·А . . . . .	65	400	500
<b>Накопители на магнитных лентах</b>			
	<b>ЕС-5010</b>	<b>ЕС-5017</b>	<b>ЕС-5022</b>
Емкость накопителя, Мбайт	20	20	40
Плотность записи, бит/мм .	32	32	32
Скорость считывания записи, кбайт/с . . . . .	64	64	126
Число дорожек записи . . .	9	9	9
Номинальная скорость движения, м/с . . . . .	2	2	2

Накопитель на гибких магнитных дисках: ЕС-5074 имеет емкость 3,2 Мбит, частоту вращения  $360 \text{ мин}^{-1}$ , скорость передачи данных 250 кбит/с, габаритные размеры  $135 \times 235 \times 380 \text{ мм}$ ; ЕС-5075 является комплексным устройством, которое состоит из двух накопителей на гибких дисках ЕС-5074, устройства управления и двух магазинов с механизмом подачи дисков.

При проектировании ФМ БД необходимо иметь сведения о функциях СУБД, характеристиках устройств прямого доступа; прикладных программах, на основе которых проектируют формат хранимой записи, метода доступа, целостность и безопасность данных.

Приведем рекомендации по созданию:

ФМ для СУБД на ИМД:

1. Упрощение иерархической структуры.
2. Изменение числа сегментов, в которых хранятся данные, за счет независимости данных.
3. Расположение сегментов, к которым чаще обращаются, ближе к вершине иерархии.

4. Разбиение большой физической БД на несколько БД; ФМ для СУБД на СМД:

1. Составление диаграммы структуры БД.
2. Учет принятых в ЛМ способов хранения и поиска записей, правила выбора, обработки и упорядочивания наборов.
3. Окончательное определение содержимого записей.

4. Структуризация переменных данных, исключение очень длинных наборов. Введение массивов указателей, равномерное распределение данных.

Очевидно, что приведенные рекомендации не догма, они могут изменяться и дополняться при работе с конкретными БД.

Отметим, что оценка ФМ до реализации БД необходима.

**Оценка памяти.** При этом обычно принимают допущение: все БД считаются неизменными; размеры сегментов малы по сравнению с блоками; сегменты имеют фиксированную длину.

Анализ проекта БД включает оценку памяти и анализа характеристик ввода/вывода. Имея характеристики ввода/вывода, образцы вызовов и длину пути к устройству, можно получить оценки языка данных (ЯД),

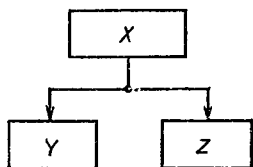


Рис. 3.19. Пример оценки фрагмента физической БД

состоящего из компонента управления БД и языка манипулирования данными. Проектировщик БД определяет средний размер физической записи БД в байтах и размер БД. Память, которую выделяют на устройстве прямого доступа, определяется числом «байтов в сегментах». Пусть имеется иерархическая древовидная структура (рис. 3.19): 3000 экземпляров сегмента  $X$  и для каждого  $X$  в среднем 10 экземпляров сегмента  $Y$  и по 60 экземпляров сегмента  $Z$ . Обозначим  $L$  длину сегмента, значит,  $L_X=3000$ ,  $L_Y=30$ ,  $L_Z=20$ . Пусть  $S$  — размер поддерева, определяемый как один экземпляр сегмента и все экземпляры порожденных им сегментов, тогда  $S_Y=30$ ,  $S_Z=20$ ,  $S_X=(10 \times S_Y) + (60 \times S_Z) + 100 = 1600$ .

Отметим, что в общем случае размер поддерева

$$S_X = L_X + (F_1 \times S_1) + (F_2 \times S_2) + \dots + (F_n \times S_n).$$

Размер поддерева равен среднему размеру записи физической БД.

В примере, изображенном на рис. 3.19, получим  $1600 \times 3000 = 4,8$  млн. байт. Для определения числа байтов в блоках умножают число байтов в сегментах на отношение размера блока к эффективному размеру блока.

**Оценка времени.** В современных СУБД при обращении

к БД основное время уходит на выполнение операций ввода/вывода. Если известно время выполнения операций ввода/вывода устройства, то нетрудно оценить время выполнения операций ввода/вывода для выполнения обращения к БД при обращении от сегмента А к сегменту В. В качестве исходных данных при определении количества операций ввода/вывода, требуемых для обращения к запрашиваемым сегментам, берут длину каждого типа сегмента, ожидаемую частоту появления экземпляра каждого типа сегмента, иерархию физической БД, размер блока и т. п. Основная часть расчетов — оценить среднее время работы процессора и время выполнения операций ввода/вывода.

После оценки параметров физической БД переходят к ее реализации. При создании сквозных интегрированных САПР, очевидно, нет смысла хранить данные для всего процесса проектирования в одной сверхсложной и большой БД, поэтому концептуально различные единицы САПР (например, этап логического и структурного синтеза) целесообразно описать в отдельных БД. Здесь не возникает проблемы установления связей и зависимостей между отдельными БД. Чисто фактическое размещение данных во вспомогательной памяти называют физической БД. Как правило, производительность БД определяется указанным размещением данных. При создании физической БД перед проектировщиком часто стоят противоречивые задачи. Приведем несколько из них. Каким образом разбивать БД на части? Необходимо ли резервировать память и в каком объеме? Каковы должны быть размеры блоков и размещаемых в них сегментов и записей? Какие будут выбраны методы доступа? Какой будет выбран метод уплотнения данных? Какая часть памяти должна располагаться на внешних носителях? и т. д. Как видно, создание физической БД, как и многие другие задачи САПР, относится к задачам многокритериальной оптимизации. Поэтому полная оптимизация физической БД в настоящее время невозможна.

Для повышения производительности БД в настоящее время проектировщики хранят используемые данные на быстродействующих устройствах, минимизируют число обращений к дискам и перемещений головок диска, распараллеливают операции проектирования.

Важным вопросом при реализации физической БД является конвертирование, т. е. преобразование в нее

существующих файлов. Иногда целесообразно или объединить существующие прикладные программы, или доработать их, или разработать совершенно новые. Все это определяет пользователь при реализации БД.

Важнейшими вопросами являются доступность БД пользователям и восстановление БД после разрушения.

Для восстановления БД после разрушения определяют ошибку, приведшую к разрушению, причины ее возникновения и выделяют разрушенную часть БД. Далее восстанавливают БД до состояния, контролируемого как свободное от ошибок. Для возвращения БД в состояние, свободное от ошибок, копируют дубликат неразрушенной БД. Снимая копии дубликатов БД и всех результатов проведенных изменений, БД приводят в рабочее состояние.

Современные СУБД записывают все действия с БД на специальный диск. Кроме того, на ленту может записываться информация о вводе/выводе в БД, состоянии БД до и после изменений.

Если при обнаружении ошибки необходимо вернуться к началу прикладной программы, то это приводит к потерям времени. Для повторного запуска процесса с некоторых точек вводят контрольные точки. Введение этих точек снижает общую производительность БД, поэтому необходим компромисс между минимальным числом таких точек и временным интервалом между двумя близлежащими контрольными точками.

Число операций, дублирование и восстановление зависят от АБД, проектировщика.

Очевидно, что в процессе эксплуатации БД нуждается в реорганизации. Современные СУБД имеют программы реорганизации, перераспределяющие БД, что позволяет освобождать память от исключенных записей и пересылать на освободившееся место записи из области переполнения БД. Под *реорганизацией* понимают копирование старой БД на другое устройство с исключением удаленных записей и повторную загрузку существующих записей с получением дубликата БД. При этом стараются часто используемые записи перевести на быстродействующие носители, а редко используемые — на медленнодействующие. По сути дела реорганизация БД — это объединение старой БД и файла изменений в новую БД. Вся реорганизация должна проводиться под контролем СУБД.

Реорганизация необходима, когда количество данных в

области переполнения достигает заданного значения, определяемого СУБД.

После некоторого времени эксплуатации БД могут меняться приоритеты прикладных программ, требования к эксплуатационным характеристикам, возникает необходимость расширения БД и т. д. Изменение БД на уровнях логической и внутренней моделей называют *реструктуризацией*. К реструктуризации также относят изменение процедур управления доступом и работой с БД. Реструктуризация БД в отличие от реорганизации БД захватывает процесс изменения вглубь и может влиять на прикладные программы и процедуры.

Классифицируют три типа реструктуризационных изменений.

1. Процедурные изменения, включающие получение дублей, восстановление и управление доступом.

2. Изменения во внутренней модели (на физическом уровне), включающие изменения указателей, конфигурации оборудования и размещения БД, размера блоков и областей, необходимых атрибутов, алгоритмов рандомизации.

3. Изменения на логическом уровне, включающие добавление или удаление элементов данных, записей, сегментов, изменение взаимосвязей между записями, изменение функций элементов данных.

В общем случае реструктуризация может вызвать изменение содержимого БД и перераспределение его на устройствах и носителях. По своим затратам реструктуризация аналогична первоначальной загрузке. Поэтому перед началом реструктуризации необходимо сопоставлять временные затраты на нее с получаемыми преимуществами.

При работе с БД необходимо иметь средства учета параметров функционирования БД и СУБД. Для этой цели в настоящее время используют мониторы съема параметров производительности, позволяющие получать общую картину работы с БД. Чаще всего производительность БД зависит от времени отклика, затрат на обновление и генерацию отчета, частоты реорганизации и затрат на реорганизацию, объема основной и внешней памяти. На производительность влияют структура БД, число различных записей и сегментов, взаимосвязи между различными физическими БД, описания и др.

Выделим основные факторы, влияющие на производительность БД. Это интенсивность обращений к БД, загруз-

ка и планирование программ обработки, число выводов БД, число операций ввода/вывода на один вызов, обработка в системе параллельно многих задач. За время отклика считают интервал времени между началом запроса и началом ответа на него:

$$t_0 = t_{\text{цп}} + t_{\text{ож}} + t_{\text{вв}} + t_{\text{обсл}} + t_{\text{з.б}} + t_{\text{з.п.д}},$$

где  $t_{\text{цп}}$  — время работы центрального процессора при запросе;  $t_{\text{ож}}$  — время ожидания в очереди на обслуживание процессора;  $t_{\text{вв}}$  — время ожидания ввода/вывода;  $t_{\text{обсл}}$  — время самого ввода/вывода;  $t_{\text{з.б}}$  — время задержки блокировки;  $t_{\text{з.п.д}}$  — время задержек передачи данных.

Существует зависимость между производительностью БД и параметрами ввода/вывода: время отклика возрастает по экспоненте при линейном увеличении интенсивности выполнения операций ввода/вывода.

Для определения производительности БД собирают статистические данные по работе процессора, каналов и устройств памяти, интенсивности потока обращений, по распределению содержимого БД, информацию по использованию модулей прикладных программ, процедур СУБД, применению БД пользователем, данные, собранные монитором БД и телеобработки. Очень важно прогнозировать производительность до построения БД. Важным вопросом при эксплуатации БД, особенно централизованного вида, является защита. Под *защитой* данных понимают предупреждение неразрешенного или случайного доступа к данным, их изменения или разрушения. В этой связи доступ к данным должен находиться под контролем. При возникновении потерь необходимо иметь программу полного восстановления данных. При надежном оборудовании и ПО нарушить защиту данных может пользователь, программист или эксплуатационник.

Пользователь при работе с БД обязан предоставлять информацию, позволяющую СУБД провести его однозначную идентификацию. В современных БД стремятся во избежание случайного разрушения данных больше функций переложить с пользователя на прикладные программы. Отметим, что разрушение данных может произойти и из-за ошибок в программах. В этой связи перед передачей программ в эксплуатацию проверяют правильность ее работы в граничной ситуации, достоверность входных данных, тестируют, проверяют простоту документирования.

Для обеспечения секретности данных проверяют пра-

тельность обработки информации, выделяют личные и решающие данные, исследуют восприимчивость системы к нарушениям. Определяют специальные средства, тестируют все процедуры проверки, выясняют права доступа пользователей и организуют полный контроль доступа к информации.

### 3.6. КРАТКАЯ ТЕХНИЧЕСКАЯ ХАРАКТЕРИСТИКА СУБД РАПИРА — 5.3—82

В базе данных САПР РАПИРА—5.3—82 (см. рис. 3.20) используется древовидная структура организации данных. Данные определяются по именам, под которыми они записываются, хранятся и извлекаются из БД. Имеющие одинаковый физический смысл данные объеди-

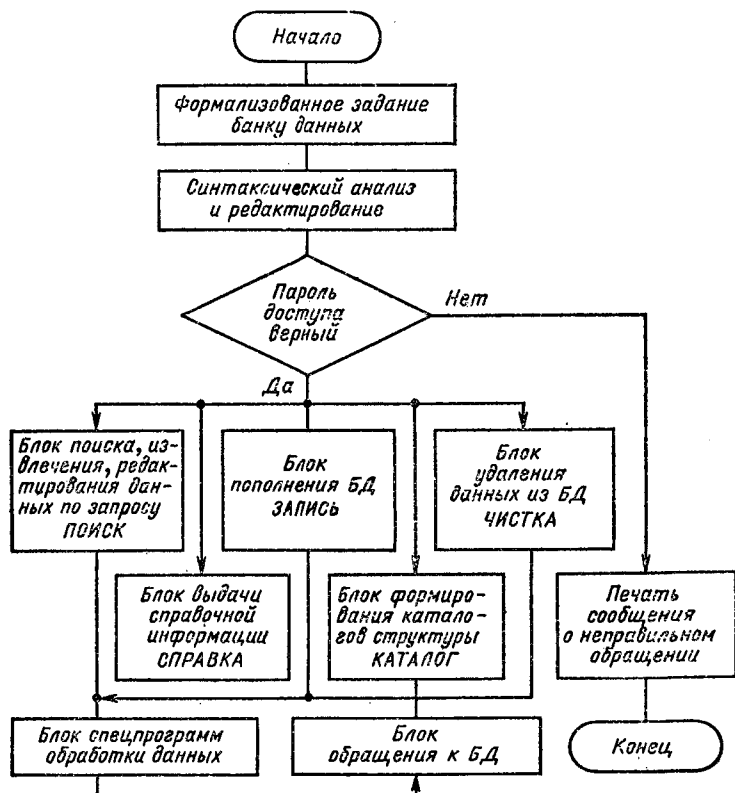


Рис. 3.20. Структура СУБД ППП РАПИРА—5.3—82



няются в группы, называемые шинами данных. Имя состоит из совокупности иерархически упорядоченных индексов, например:

ЭЛ	МИКРОСХЕМА	K155	ГГ	Б
1	2	3	4	5

1 — множество объектов ЭЛЕМЕНТЫ;

2 — подмножество МИКРОСХЕМЫ;

3 — подмножество K155;

4 — группа геометрии;

5 — модификация Б.

Система управления БД использует набор данных на магнитных дисках с прямой организацией. Обращения к СУБД могут быть из входного потока заданий или из активной программы САПР. Минимальный объем оперативной памяти, требуемый для работы СУБД: при обращении из входного потока заданий — 160 кбайт; при обращении из активных программ САПР — 60 кбайт.

Среднее время ответа при поиске одного объекта для ЭВМ ЕС-1033  $15 \text{ с} \leq T \leq 45 \text{ с}$ .


Опишем язык манипулирования данными СУБД САПР РАПИРА—5.3—82. Средством обращения к данным является формализованное задание базе данных (ФЗБД).

**Описание ФЗБД.**

ФЗБД ::= <СПИСОК ЭЛЕМЕНТОВ ФЗ>; <КОММЕНТАРИЙ>  
СПИСОК ЭЛЕМЕНТОВ ФЗ ::= <ЭЛЕМЕНТ ФЗ> | <СПИСОК ЭЛЕМЕНТОВ ФЗ> <ЭЛЕМЕНТ ФЗ>

КОММЕНТАРИЙ ::= (<НАБОР РАЗРЕШЕННЫХ СИМВОЛОВ>) |  
| <ПУСТО> <ЭЛЕМЕНТ ФЗ> ::= <ШИФР ПОЛЬЗОВАТЕЛЯ>

<ИМЯ ФУНКЦИИ> <ПРЕДПИСАНИЕ>  <КОММЕНТА-

РИЙ> |  <КОММЕНТАРИЙ>

<ШИФР ПОЛЬЗОВАТЕЛЯ> ::= ШП = <НАБОР ИЗ 10 ИЛИ МЕНЕЕ СИМВОЛОВ>, | <ПУСТО>

<ФУНКЦИЯ> может быть:

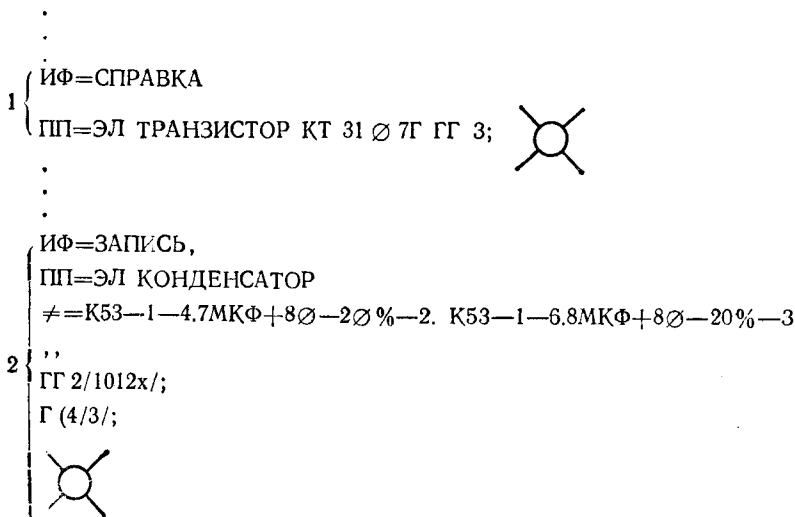
ЗАПИСЬ (для записи информации в базу данных);

СПРАВКА (для печати справочной информации о содержимом и организации базы данных);

ПОИСК (для поиска информации из базы данных);

КАТАЛОГП, КАТАЛОГС, КАТАЛОГЗ (для формирования каталогов структур функции ПОИСК, СПРАВКА, ЗАПИСЬ соответственно).

Пример.



Элемент формализованного задания 1 содержит задания на выдачу справочной информации о первой группе геометрии элемента ТРАНЗИСТОР КТ3107Г.

Элемент формализованного задания 2 содержит информацию для записи в БД второй группы геометрии элементов КОНДЕНСАТОР К53—1—4.7МКФ+80—20%—2 и КОНДЕНСАТОР К53—1—6.8МКФ+80—20%—3 и четвертой группы эквивалентности для этих элементов.

Программы СУБД РАПИРА—5.3—82 обеспечивает обработку следующих типов данных: групп геометрии и эквивалентных клемм; описаний групповых и одиночных базовых элементов; групп эквивалентных клемм, технических параметров элементов, параметров тепловой модели, системных таблиц программного комплекса «документ»; данных, не требующих обработки при записи или извлечении из базы данных. Язык описания данных СУБД САПР РАПИРА—5.3—82 предназначен для описания:

электрорадиоэлементов (ЭРЭ); компонентов коммутационной платы; параметров тепловой модели платы; постоянных частей формализованного задания на проектирование.

Электрорадиоэлементы и компоненты коммутационной платы описываются: одиночными (ОБЭ) и групповыми (ГБЭ) базовыми элементами, группами геометрии, группами эквивалентности элементов, техническими параметрами базовых элементов.

Одиночными базовыми элементами описывают элементарную составляющую геометрии базового элемента.

Групповыми базовыми элементами описывают совокупность и взаимное расположение ОБЭ, входящих в описание геометрии базовых элементов.

Группа геометрии описывает установочное место базового элемента.

Группа эквивалентности описывает группы инвариантных в электрическом отношении контактов базового элемента.

Технические параметры базового элемента описывают: максимально допустимую мощность и температуру базового элемента; тепловое сопротивление корпуса базового элемента.

#### Описание ОБЭ

Формальное описание ОБЭ имеет вид:

$\langle \text{ОПИСАНИЕ ОБЭ} \rangle ::= \langle \text{ФРАГМЕНТ ОБЭ} \rangle | \langle \text{ОПИСАНИЕ ОБЭ} \rangle \langle \text{ФРАГМЕНТ ОБЭ} \rangle$

$\langle \text{ФРАГМЕНТ ОБЭ} \rangle ::= \langle \text{КОД ФРАГМЕНТА} \rangle, \langle \text{ОПИСАНИЕ ФРАГМЕНТА} \rangle$

$\langle \text{КОД ФРАГМЕНТА} \rangle$  определяет конструктивно-технологические характеристики фрагмента и определяется по специальному кодификатору.

$\langle \text{ОПИСАНИЕ ФРАГМЕНТА} \rangle$  содержит информацию о размерах фрагмента, координатах левого нижнего угла фрагмента относительно базовой точки ОБЭ, высоте навесного элемента, маркировочной информации.

#### Пример.

Запись

.  
.  
0150 и /((01230,2.0)/2.0—1.0/—1.0)/;  
.  
.  
.

Определяют прямоугольную клемму навесного элемента, сквозную с металлизацией на 1- и 2-м слоях платы, предназначенную для использования в технологической документации и проектировании, размером 2×2 мм с координатами левого нижнего угла — 1,—1 мм относительно базовой точки.

Запись

.  
.  
0210/(00041, 10.0/5.0.—5.0/2.5)/  
.  
.  
.

Определяет прямоугольную, планарную клемму навесного элемента на 1-м слое платы, предназначенную для использования в конструкторской документации с размерами  $10 \times 5$  мм и координатами левого нижнего угла—5,0,—2,5 мм относительно базовой точки.

#### Описание ГБЭ.

Формальное описание ГБЭ имеет вид:

$\langle \text{ОПИСАНИЕ ГБЭ} \rangle ::= \langle \text{ОПИСАНИЕ УСТАНОВОК ОБЭ} \rangle,$   
 $\langle \text{ОПИСАНИЕ АППРОКСИМИРУЮЩЕГО ПРЯМОУГОЛЬНИКА} \rangle,$   
 $\langle \text{ОПИСАНИЕ КОРПУСА БАЗОВОГО ЭЛЕМЕНТА} \rangle$

$\langle \text{ОПИСАНИЕ УСТАНОВОК ОБЭ} \rangle ::= \langle \text{ОПИСАНИЕ УСТАНОВКИ ОБЭ} \rangle | \langle \text{ОПИСАНИЕ УСТАНОВОК ОБЭ} \rangle,$   $\langle \text{ОПИСАНИЕ УСТАНОВКИ ОБЭ} \rangle$

$\langle \text{ОПИСАНИЕ УСТАНОВКИ ОБЭ} \rangle ::= \langle X_{бт} \rangle | \langle Y_{бт} \rangle$

НОМЕР ОБЭ

$\langle \text{ОПИСАНИЕ АППРОКСИМИРУЮЩЕГО ПРЯМОУГОЛЬНИКА} \rangle ::= \langle X_{лну} \rangle | \langle Y_{лну} \rangle,$   $\langle D_{хап} \rangle | \langle D_{уап} \rangle$

$\langle \text{ОПИСАНИЕ КОРПУСА БАЗОВОГО ЭЛЕМЕНТА} \rangle ::= \langle X_{ц} \rangle |$   
 $| \langle Y_{ц} \rangle,$   $\langle L_x \rangle | \langle L_y \rangle$

$\langle X_{бт} \rangle,$   $\langle Y_{бт} \rangle$  — координаты базовой точки ОБЭ относительно базовой точки ГБЭ;

$\langle X_{лну} \rangle,$   $\langle Y_{лну} \rangle$  — координаты левого нижнего угла аппроксимирующего прямоугольника относительно базовой точки ГБЭ;

$\langle D_{хап} \rangle,$   $\langle D_{уап} \rangle$  — габариты аппроксимирующего прямоугольника;

$\langle X_{ц} \rangle,$   $\langle Y_{ц} \rangle$  — координаты центра корпуса базового элемента относительно базовой точки ГБЭ;

$\langle L_x \rangle,$   $\langle L_y \rangle$  — габариты корпуса базового элемента.

#### Пример.

·  
·  
·

ИФ — ЗАПИСЬ, ПП-ГБЭ 0215/:/2.5:0430, 10/2.5:0430,15/2.5:0110.—  
10/2—2.5,20/5,0/0,19/9/:

·  
·  
·

В примере приведено задание на запись в базу данных ГБЭ с номером 0215, состоящего из ОБЭ номер 0430, находящихся в точках с координатами (5.2.5), (10.2.5), и ОБЭ номер 0110 в точке с координатами (15.2.5) с размером корпуса  $18 \times 9$  и размером аппроксимирующего прямоугольника  $20 \times 5$  мм.

Рассмотрим пример проектирования БД для проектной организации, выполняющей проектирование специализированных СБИС.

## Проектирование концептуальной модели БД.

1.1. Исследование предметной области и формализация. Информационные потребности проектной организации представлены на рис. 3.21. Требуется следующие документы:

перечень проектных операций для данной СБИС. Содержит перечень проектных операций, которые должны быть выполнены для получения проекта СБИС, который удовлетворяет требованиям заказчика;

список СБИС, требующих данную проектную операцию. Содержит перечень СБИС, процесс проектирования которых включает данную проектную операцию (размещение, электрическое моделирование и т. п.);

распределение машинного времени. Содержит информацию о СБИС, проектных операциях, проектировщиках и т. п. и распределении машинного времени для их проектирования;

перечень возможных проектных операций. Содержит список проектных операций, которые могут быть рекомендованы имеющимися в проектной организации средствами и специалистами;

информация о ранее спроектированных СБИС. Сведения о ранее проектированных СБИС, заказчиках и т. п.;

сведения о СБИС. Информация о проектируемой СБИС;

сведения о проектировщиках. Информация о проектировщиках (квалификация, выполняемые проектные операции и т. п.);

сведения об ЭВМ. Информация о типе, производительности и про-



Рис. 3.21. Предметная область «Проектирование СБИС»

чих характеристиках ЭВМ, имеющих в распоряжении проектировщиков. Форматы перечисленных документов приведены ниже:

Формат документа РАСПРЕДЕЛЕНИЕ МАШИННОГО ВРЕМЕНИ

НОМЕР ПРОЕКТНОЙ ОПЕРАЦИИ	НАИМЕНОВАНИЕ ПРОЕКТНОЙ ОПЕРАЦИИ	ИМЯ ПРОЕКТИРОВЩИКА	ЭВМ	ТРЕБУЕМОЕ ВРЕМЯ	НЕОБХОДИМАЯ ДОКУМЕНТАЦИЯ
04	ТРАССИРОВКА КАНАЛЬНАЯ		ЕС-1045	40 мин	РАЗМЕЩЕНИЕ
		...			

Формат документа ПЕРЕЧЕНЬ ВОЗМОЖНЫХ ПРОЕКТНЫХ ОПЕРАЦИЙ

ОПЕРАЦИЯ: . . .  
 ТИП ОПЕРАЦИИ: . . .  
 НОМЕР ОПЕРАЦИИ. . .  
 ОПИСАНИЕ ОПЕРАЦИИ. . .  
 НЕОБХОДИМАЯ ДОКУМЕНТАЦИЯ . . .

Формат документа ИНФОРМАЦИЯ О РАННЕ СПРОЕКТИРОВАННЫХ СБИС

СБИС	НАИМЕНОВАНИЕ СБИС	ЗАКАЗЧИК	АДРЕС ЗАКАЗЧИКА
A127	МУЛЬТИПЛЕКСОР		
		...	

Формат документа ПЕРЕЧЕНЬ ПРОЕКТНЫХ ОПЕРАЦИЙ ДАННОЙ СБИС

СБИС: . . . . . (регистрационный номер)  
 НАИМЕНОВАНИЕ: . . . . .

ОТВЕТСТВЕННЫЙ ПРОЕКТИРОВЩИК:...(ФИО)

СТАТУС: (срочный, обычный)...

НОМЕР ПРОЕКТНОЙ ОПЕРАЦИИ	НАЗВАНИЕ ПРОЕКТНОЙ ОПЕРАЦИИ	ИМЯ ПРОЕКТИРОВЩИКА	ЭВМ	ТРЕБУЕМОЕ ВРЕМЯ	НЕОБХОДИМАЯ ДОКУМЕНТАЦИЯ
ТР04	ТРАССИРОВКА КАНАЛЬНАЯ	. . .	ЕС-1045	40 мин	РАЗМЕЩЕНИЕ

Формат документа СПИСОК СБИС, ТРЕБУЮЩИХ ДАННУЮ ПРОЕКТНУЮ ОПЕРАЦИЮ

ОПЕРАЦИЯ:... (наименование операции)...

ТИП ОПЕРАЦИИ... (моделирование, конструирование)

НОМЕР ПРОЕКТНОЙ ОПЕРАЦИИ:...

ИМЯ ПРОЕКТИРОВЩИКА...

ЭВМ...

ТРЕБУЕМОЕ ВРЕМЯ...

СБИС	НАИМЕНОВАНИЕ СБИС	РЕЗУЛЬТАТ ПРОЕКТИРОВАНИЯ
A127	МУЛЬТИПЛЕКСОР ...	ПОЛНАЯ ТРАССИРОВКА ПОСЛЕ ОПЕРАЦИИ ТР04

Формат документа РАСПРЕДЕЛЕНИЕ МАШИННОГО ВРЕМЕНИ

НОМЕР ПРОЕКТНОЙ ОПЕРАЦИИ	НАИМЕНОВАНИЕ ПРОЕКТНОЙ ОПЕРАЦИИ	ИМЯ ПРОЕКТИРОВЩИКА	ЭВМ	ТРЕБУЕМОЕ ВРЕМЯ	НЕОБХОДИМАЯ ДОКУМЕНТАЦИЯ
ТР04	ТРАССИРОВКА КАНАЛЬНАЯ		ЕС-1045	40 мин	РАЗМЕЩЕНИЕ

Формат документа СВЕДЕНИЯ О СБИС

ОПЕРАЦИЯ:...

ТИП ОПЕРАЦИИ:...

НОМЕР ОПЕРАЦИИ...  
 ОПИСАНИЕ ОПЕРАЦИИ...  
 НЕОБХОДИМАЯ ДОКУМЕНТАЦИЯ...

Формат документа СВЕДЕНИЯ О ПРОЕКТИРОВЩИКАХ

Ф.И.О. ПРОЕКТИРОВЩИКА...

НОМЕР ПРОЕКТНОЙ

ОПЕРАЦИИ

КВАЛИФИКАЦИЯ

ТР04

КОРРЕКТИРОВКА РЕЗУЛЬТАТА

Формат документа СВЕДЕНИЯ ОБ ЭВМ

НАИМЕНОВАНИЕ ЭВМ...

ПРОИЗВОДИТЕЛЬНОСТЬ...

ТЕЛЕФОН ЭВМ...

КОЛИЧЕСТВО НАКОПИТЕЛЕЙ НА МАГНИТНЫХ ДИСКАХ...

1.2. Определение элементов данных, требуемых в каждом из документов.

Таблица соответствия данных и документов показана ниже:

ЭЛЕМЕНТЫ ДАННЫХ	ДОКУМЕНТЫ							
	Перечень проектных операций данной СБИС	Список СБИС, требующих данную проектную операцию	Распределение машинного времени	Перечень возможных проектных операций	Информация о ранее спроектированных СБИС	Сведения о СБИС	Сведения о проектировщиках	Сведения об ЭВМ
СБИС	×	×			×	×		
НАИМЕНОВАНИЕ	×	×			×	×		
ОТВЕТСТВЕННЫЙ ПРОЕКТИРОВЩИК	×					×		
СТАТУС	×							
НОМЕР ПРОЕКТНОЙ ОПЕРАЦИИ	×	×	×	×			×	
НАЗВАНИЕ ПРОЕКТНОЙ ОПЕРАЦИИ	×	×	×	×				
ИМЯ ПРОЕКТИРОВЩИКА	×	×	×	×				
ЭВМ	×	×	×	×				×
ТРЕБУЕМОЕ ВРЕМЯ	×	×	×	×				
НЕОБХОДИМАЯ ДОКУМЕНТАЦИЯ	×	×	×	×				
ТИП ОПЕРАЦИИ		×		×		×		



ЭЛЕМЕНТЫ ДАННЫХ	ДОКУМЕНТЫ							
	Перечень проектных операций данной СБИС	Список СБИС, требующих данную проектную операцию	Распределение машинного времени	Перечень возможных проектных операций	Информация о ранее спроектированных СБИС	Сведения о СБИС	Сведения о проектировщиках	Сведения об ЭВМ
РЕЗУЛЬТАТ ПРОЕКТИРОВАНИЯ		×				×		
ОПИСАНИЕ ОПЕРАЦИИ				×				
ЗАКАЗЧИК					×	×		
АДРЕС ЗАКАЗЧИКА					×	×		
НОМЕР ЗАДАНИЯ НА ПРОЕКТИРОВАНИЕ						×		
НОМЕР ДОКУМЕНТАЦИИ НА ПРОЕКТИРОВАНИЕ					×	×		
КВАЛИФИКАЦИЯ							×	
ПРОИЗВОДИТЕЛЬНОСТЬ								×
ТЕЛЕФОН ЭВМ								×
КОЛИЧЕСТВО НАКОПИТЕЛЕЙ НА МАГНИТНЫХ ДИСКАХ								×

1.3. Установление взаимосвязей между элементами данных. Приведение схемы отношений элементов данных к нормальной форме. Построение концептуальной модели.

1. НОМЕР СБИС НАИМЕНОВАНИЕ СБИС, ЗАКАЗЧИК.....НОМЕР ДОКУМЕНТАЦИИ НА ПРОЕКТИРОВАНИЕ.
  2. НОМЕР СБИС И НОМЕР ПРОЕКТНОЙ ОПЕРАЦИИ РЕЗУЛЬТАТ ПРОЕКТИРОВАНИЯ.
  3. ИМЯ ПРОЕКТИРОВЩИКА И НОМЕР ПРОЕКТНОЙ ОПЕРАЦИИ КВАЛИФИКАЦИЯ.
  4. ЭВМ ПРОИЗВОДИТЕЛЬНОСТЬ....КОЛИЧЕСТВО НАКОПИТЕЛЕЙ НА МАГНИТНЫХ ДИСКАХ....
- и т. д.

Опишем теперь кратко основные функции промышленных СУБД, используемых в САПР различного назначения. Для построения информационно-справочных и управляю-

щих систем, обрабатывающих большие объемы информации и устанавливающих сложные отношения между элементами данных, используют СУБД ОКА. Она поддерживает ИМД и реализуется в среде ОС ЕС ЭВМ.

СУБД ОКА выполняет следующие основные функции: многоаспектный доступ к данным; защиту данных от несанкционированного доступа; расширение базы данных без изменения используемых проблемных программ, реорганизацию и реструктуризацию БД; работу в пакетном и диалоговом режимах, а также в режиме телеобработки; протоколирование всех изменений в БД и восстановление при отказах и сбоях в ТС.

СУБД БАНК ОС используют для создания БД различной структуры на основании конкретных требований пользователя. БАНК ОС позволяет выполнять: многоаспектный доступ к данным; работу в мультизадачном режиме; сбор статистики; протоколирование всех изменений и восстановление БД; реорганизацию и реструктуризацию БД.

СУБД ИНЕС поддерживает ИМД, реализована в среде ОС ЕС ЭВМ и выполняет следующие функции: ввод и контроль исходных данных, загрузку и корректировку БД; вывод формализованных данных на печать; управление неструктурированными данными; обеспечение процедур отладки проблемных программ.

СУБД АИСТ реализована в среде ОС ЕС ЭВМ, поддерживает СМД и выполняет функции: загрузку БД; установление связи между файлами БД; многоаспектный доступ к данным; мультизадачный режим работы; проектирование, сбор статистики, выдачу информации о состоянии БД; копирование и восстановление БД; реорганизацию и реструктуризацию БД.

СУБД СЕТОР-2 реализована в среде ОС ЕС ЭВМ, поддерживает СМД и выполняет следующие функции: многоаспектный доступ к данным; многозадачный режим работы; сбор статистики, протоколирование и восстановление БД, анализ и восстановление БД, анализ ее состояния; реорганизацию и реструктуризацию БД.

Отметим, что описанные СУБД комплексированы с пакетами прикладных программ.

В заключение раздела отметим следующее. Тенденции развития ИО САПР показывают перспективность построения распределенных баз данных. Это позволит обеспечить управляемый прямой доступ и независимое обращение к данным, распределенным в сетях ЭВМ или много-

процессорных вычислительных системах. Предполагается за счет развития технологии получить в ближайшем будущем устройства массовой памяти с прямым доступом емкостью триллионы бит и выше.

Перспективным направлением совершенствования организации САПР и повышения качества получаемых проектных решений является использование в составе информационного обеспечения баз знаний (БЗ).

Применительно к САПР основными функциями БЗ являются: 1) описание предметной области САПР; 2) поддержка интеллектуальных решателей задач (ИРЗ), входящих в состав САПР; 3) реализация возможности экспертного анализа проектных заданий.

Знания в БЗ описываются на некотором внутреннем языке в виде таких объектов: понятия, индивидуальные объекты, отношения, факты и закономерности.

Понятия представляют собой общие описания предметов и событий, характеризующихся одинаковыми наборами свойств, и могут находиться в подчиненности друг у друга. Самое общее понятие является базовым. Каждое более общее понятие по отношению к соседнему, менее общему понятию играет роль надпонятия. Например, для К155ТМ2 примером надпонятия является МИКРОСХЕМА.

Индивидуальные объекты представляются описаниями самого нижнего уровня иерархической структуры понятий. Они характеризуют терминальные значения свойств. Отношения выражают связи между предметами и действиями. Конкретизация отношения является фактом.

Закономерности описывают процедуры, выражающие различные актуальные знания.

Для описания вышеперечисленных объектов используются специальные языки БЗ, которые должны отвечать следующим основным требованиям: агрегированности для обеспечения легкости описания предметной области БЗ; возможности определения в языке состава знаний; упорядоченности знаний; характера представления знаний; представления знаний в едином формализме, пригодном для решения интеллектуальных задач автоматизированного проектирования широкого класса.

Использование БЗ совместно с интеллектуальными решателями задач САПР позволяет:

1. Имитировать мыслительную деятельность человека при решении широкого класса задач автоматизированного проектирования.

2. Осуществлять эвристический поиск в пространстве возможных решений.

3. Использовать для представления знаний языки высокого уровня.

4. Обучать САПР и пополнять БЗ в процессе работы.

Базы знаний предполагают наличие в составе ТС САПР высокопроизводительных ЭВМ с быстродействием порядка  $10^7$  оп/с и объемом памяти 10—50 Мбайт. Перспективным подходом является создание специализированных машин БЗ, широко использующих аппаратную реализацию различных функций обработки знаний.

### Контрольные вопросы и задания

1. Что такое система данных?

2. Определите предметную область, объект, атрибут (элемент данных), значения данных и постройте таблицы связи между ними.

3. Что такое идентификаторы объекта и ключевые атрибуты?

4. Что такое запись данных? Приведите примеры.

5. Что такое файл данных (набор данных)?

6. Приведите пример взаимно однозначного соответствия между прикладными программами логического проектирования и файлами данных.

7. Какие проблемы возникают при обработке данных с несколькими файлами?

8. Приведите известные определения базы данных (БД).

9. В чем сходство и различие между БД и файлом?

10. Приведите основные определения системы управления базами данных (СУБД).

11. Опишите основные функции СУБД и требования к ним.

12. В чем заключается информационная согласованность в САПР?

13. Что такое функция администрирования БД и кто такой администратор БД (АБД)? Какие функции выполняет АБД?

14. Что такое независимость данных?

15. Какие языки используются в БД?

16. Что такое концептуальная модель (КМ)?

17. Приведите определение логической, внешней, внутренней (физической) моделей.

18. Что такое независимость данных?

19. Что такое словарь данных (СД) и каковы его функции?

20. Каковы функции информационного фонда САПР?

21. Постройте структурную схему проектирования БД САПР ЭВМ пятого поколения.

22. Приведите основные этапы работы системы с БД.

23. Что такое модель данных?
24. Опишите иерархическую модель данных (ИМД). Постройте пример.
25. Опишите сетевую модель данных (СМД) и построьте пример.
26. Опишите реляционную модель данных (РМД) и постройте пример.
27. Приведите виды взаимосвязей между объектами и атрибутами.
28. Что такое распределенные БД?
29. В чем заключается проектирование концептуальной модели?
30. Опишите последовательность проектирования логической модели. Постройте пример.
31. Приведите классификацию методов доступа внутренних моделей.
32. Опишите физический последовательный метод доступа. Постройте пример.
33. Опишите индексно-произвольный и индексно-последовательный методы доступа. Постройте пример.
34. Опишите метод хеширования и инвертированный метод. Постройте примеры.
35. Приведите методы доступа внешней модели.
36. Опишите этап создания физической модели БД.
37. Приведите технические характеристики устройств хранения информации.
38. Приведите пример оценки памяти ФМ БД.
39. В чем заключается конвертирование при реализации физической БД?
40. Опишите процессы реорганизации и реструктуризации при эксплуатации БД.
41. Чем определяется производительность БД?
42. В чем заключается защита данных?
43. Постройте структурные схемы современных СУБД.
44. Что такое базы знаний, какова их роль в САПР?

## ГЛАВА 4

### МАТЕМАТИЧЕСКИЕ МОДЕЛИ ПРОЕКТИРУЕМЫХ ОБЪЕКТОВ

#### 4.1. КЛАССИФИКАЦИЯ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ

*Математическая модель* (ММ) технического объекта есть совокупность математических объектов (чисел, переменных, матриц, множеств и т. п.) и отношений между ними.

ми, которая адекватно отображает свойства технического объекта, интересующие инженера, разрабатывающего этот объект. Выполнение проектных операций и процедур в САПР основано на оперировании ММ. С их помощью прогнозируются характеристики и оцениваются возможности предложенных вариантов схем и конструкций, проверяется их соответствие предъявляемым требованиям, проводится оптимизация параметров, разрабатывается техническая документация и т. п.

В САПР для каждого иерархического уровня сформулированы основные положения математического моделирования, выбран и развит соответствующий математический аппарат, получены типовые ММ элементов проектируемых объектов, формализованы методы получения и анализа математических моделей систем. Сложность задач проектирования и противоречивость требований высокой точности, полноты и малой трудоемкости анализа обуславливают целесообразность компромиссного удовлетворения этих требований с помощью соответствующего выбора моделей. Это обстоятельство приводит к расширению множества используемых моделей и развитию алгоритмов адаптивного моделирования.

**Функциональные и структурные модели.** В проектных процедурах, связанных с функциональным аспектом проектирования, как правило, используются ММ, отражающие закономерности процессов функционирования объектов. Такие модели называют *функциональными*. Типичная функциональная модель представляет собой систему уравнений, описывающих либо электрические, тепловые, механические процессы, либо процессы преобразования информации.

В то же время в процедурах, относящихся к конструкторскому аспекту проектирования, преобладает использование математических моделей, отражающих только структурные свойства объекта, например его геометрическую форму, размеры, взаимное расположение элементов в пространстве. Такие модели называют *структурными*. Структурные модели чаще всего представляются в виде графов, матриц инцидентий и смежности, списков и т. п.

Как правило, функциональные модели более сложные, поскольку в них отражаются также сведения о структуре объектов. Однако при решении многих задач конструирования использование сложных функциональных моделей неоправдано, так как нужные результаты могут быть по-

лучены на основе более простых структурных моделей. Функциональные модели применяют преимущественно на завершающих этапах верификации описаний объектов, предварительно синтезированных с помощью структурных моделей.

**Иерархия математических моделей в САПР.** Блочнo-иерархический подход к проектированию радиоэлектронной аппаратуры (РЭА) включает в качестве своей основы иерархию математических моделей. Деление моделей по иерархическим уровням (уровням абстрагирования) происходит по степени детализации описываемых свойств и процессов, протекающих в объекте. При этом на каждом иерархическом уровне используют свои понятия «система» и «элементы». Так, система  $k$ -го уровня рассматривается как элемент на соседнем более высоком ( $k-1$ )-м уровне абстрагирования.

Представим структуру некоторого объекта в виде множества элементов (рис. 4.1) и связей между ними. Выде-

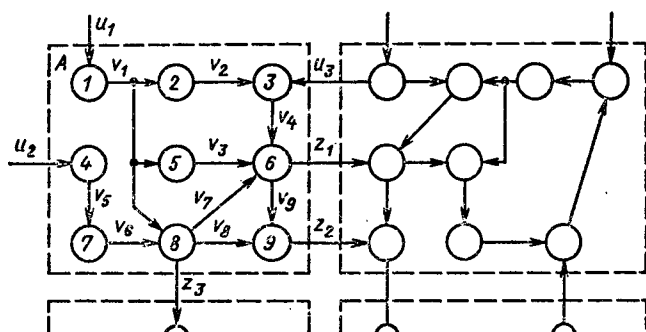


Рис. 4.1. Представление структуры объекта

лим в соответствии с блочно-иерархическим подходом в структуре объекта некоторые подмножества элементов и назовем их блоками (на рисунке показаны штриховыми линиями). Пусть состояние каждой связи характеризуется одной фазовой переменной  $v_i$ ,  $z_j$  или  $u_k$ . Здесь  $v_i$  относится к внутренним связям между элементами данного блока,  $z_j$ ,  $u_k$  относятся к выходам и входам блока соответственно. Теперь поясним важные для функциональных моделей понятия полной модели и макромоделли.

*Полная модель* блока есть модель, составленная из мо-

делей элементов с учетом межэлементных связей, т. е. модель, описывающая как состояние выходов, так и состояние каждого из элементов блока. Моделями элементов блока  $A$  являются уравнения, связывающие входные и выходные переменные:

$$\left. \begin{aligned} f_1(v_1, u_1) &= 0; \\ f_2(v_1, v_2) &= 0; \\ f_3(v_2, u_3, v_4) &= 0; \\ \dots &\dots \dots \dots \\ f_n(v_n, v_n, z_2) &= 0. \end{aligned} \right\} \quad (4.1)$$

Полная модель блока есть система уравнений

$$F(V, U) = 0, Z = \Psi(V, U), \quad (4.2)$$

где  $V$ ,  $Z$  и  $U$  — векторы внутренних, выходных и входных фазовых переменных блока.

При большом количестве элементов размерность вектора  $V$  и порядок системы уравнений (4.2) становятся чрезмерно большими и требуются упрощения.

При переходе к более высокому иерархическому уровню упрощения основаны на исключении из модели вектора внутренних переменных  $V$ . Полученная модель представляет собой систему уравнений

$$\varphi(Z, U) = 0 \quad (4.3)$$

существенно меньшей размерности, чем полная модель (4.2), и называется *макромоделью*. Следовательно, макромодель уже не описывает процессы внутри блока, а характеризует только процессы взаимодействия данного блока с другими в составе системы блоков.

Модели (4.2) и (4.3) относятся друг к другу как полная модель и макромодель на  $n$ -м уровне иерархии. На более высоком ( $n-1$ )-м уровне блок  $A$  рассматривается как элемент и макромодель (4.3) становится моделью элемента  $A$ . Следовательно, модели (4.1) и (4.3) относятся друг к другу как модели элементов соседних иерархических уровней. Из моделей типа (4.3) может быть составлена полная модель системы на ( $n-1$ )-м уровне.

**Микро-, макро- и метауровни.** В зависимости от сложности объекта при его проектировании используют большее или меньшее число уровней абстракции. Объединение уровней, родственных по характеру используемого математического аппарата, приводит к образованию трех укруп-



ненных уровней — микро-, макро- и метауровня — в иерархии функциональных моделей для большинства проектируемых сложных объектов.

На *микроуровне* используют математические модели, описывающие физическое состояние и процессы в сплошных средах. Для моделирования применяют аппарат уравнений математической физики. Примерами таких уравнений служат дифференциальные уравнения в частных производных—уравнения электродинамики, теплопроводности, упругости, газовой динамики. Эти уравнения описывают поля электрического потенциала и температуры в полупроводниковых кристаллах интегральных схем, напряженно-деформированное состояние деталей механических конструкций и т. п. К типичным фазовым переменным на микроуровне относятся электрические потенциалы, давления, температуры, концентрации частиц, плотности токов, механические напряжения и деформации. Независимыми переменными являются время и пространственные координаты. В качестве операторов  $\mathbf{F}$  и  $\Psi$  в уравнениях (4.2) фигурируют дифференциальные и интегральные операторы. Уравнения (4.2), дополненные краевыми условиями, составляют ММ объектов на микроуровне. Анализ таких моделей сводится к решению краевых задач математической физики.

На *макроуровне* производится дискретизация пространств с выделением в качестве элементов отдельных деталей, дискретных электрорадиоэлементов, участков полупроводниковых кристаллов. При этом из числа независимых переменных исключают пространственные координаты. Функциональные модели на макроуровне представляют собой системы алгебраических или обыкновенных дифференциальных уравнений, для их получения и решения используют соответствующие численные методы. В качестве фазовых переменных фигурируют электрические напряжения, токи, силы, скорости, температуры, расходы и т. д. Они характеризуют проявления внешних свойств элементов при их взаимодействии между собой и внешней средой в электронных схемах или механических конструкциях.

На *метауровне* с помощью дальнейшего абстрагирования от характера физических процессов удается получить приемлемое по сложности описание информационных процессов, протекающих в проектируемых объектах. На метауровне для моделирования аналоговой РЭА широко применяют аппарат анализа систем автоматического управ-

ления, а для моделирования цифровой РЭА — математическую логику, теорию конечных автоматов, теорию массового обслуживания. Математические модели на метауровне — системы обыкновенных дифференциальных уравнений, системы логических уравнений, имитационные модели систем массового обслуживания.

**Формы представления моделей.** Для представления моделей используют следующие основные формы:

*Инвариантная* форма — запись соотношений модели с помощью традиционного математического языка безотносительно к методу решения уравнений модели.

*Алгоритмическая* форма — запись соотношений модели и выбранного численного метода решения в форме алгоритма.

*Аналитическая* форма — запись модели в виде результата аналитического решения исходных уравнений модели; обычно модели в аналитической форме представляют собой явные выражения выходных параметров как функций внутренних и внешних параметров.

*Схемная* форма, называемая также *графической* формой, — представление модели на некотором графическом языке, например на языке графов, эквивалентных схем, диаграмм и т. п. Графические формы удобны для восприятия человеком. Использование таких форм возможно при наличии правил однозначного истолкования элементов чертежей и их перевода на язык инвариантных или алгоритмических форм.

Модели в алгоритмической и аналитической формах называют соответственно *алгоритмическими* и *аналитическими*. Среди алгоритмических моделей важный класс составляют *имитационные модели*, предназначенные для имитации физических или информационных процессов в объекте при задании различных зависимостей входных воздействий от времени. Собственно имитацию названных процессов называют *имитационным моделированием*. Результат имитационного моделирования — зависимости фазовых переменных в избранных элементах системы от времени. Примерами имитационных моделей являются модели электронных схем в виде систем обыкновенных дифференциальных уравнений или модели систем массового обслуживания, предназначенные для имитации процессов прохождения заявок через систему.

## 4.2. ТРЕБОВАНИЯ К МАТЕМАТИЧЕСКИМ МОДЕЛЯМ

Основными требованиями, предъявляемыми к математическим моделям, являются требования адекватности, универсальности и экономичности.

**Адекватность.** Модель считается адекватной, если отражает заданные свойства объекта с приемлемой точностью. Точность определяется как степень совпадения значений выходных параметров модели и объекта. Пусть  $\varepsilon_j$  — относительная погрешность модели по  $j$ -му выходному параметру:

$$\varepsilon_j = (\tilde{y}_j - y_j) / y_j,$$

где  $\tilde{y}_j$  —  $j$ -й выходной параметр, рассчитанный с помощью модели;  $y_j$  — тот же выходной параметр, имеющий место в моделируемом объекте.

Погрешность модели  $\varepsilon_m$  по совокупности учитываемых выходных параметров оценивается одной из норм вектора  $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m)$ , например

$$\varepsilon_m = \max_{j=1, m} |\varepsilon_j| \text{ или } \varepsilon_m = \sqrt{\sum_{j=1}^m \varepsilon_j^2}.$$

Точность модели различна в разных условиях функционирования объекта. Эти условия характеризуются внешними параметрами. Если задаться предельной допустимой погрешностью  $\varepsilon_{\text{пред}}$ , то можно в пространстве внешних параметров выделить область, в которой выполняется условие

$$\varepsilon_m < \varepsilon_{\text{пред}}.$$

Эту область называют *областью адекватности* (ОА) модели. Возможно введение индивидуальных предельных значений  $\varepsilon_{\text{пред } j}$  для каждого выходного параметра и определение ОА как области, в которой одновременно выполняются все  $m$  условий вида  $|\varepsilon_j| \leq \varepsilon_{\text{пред } j}$ .

Пример ОА (заштрихована) в двумерном пространстве дан на рис. 4.2. Здесь  $q_k$  —  $k$ -й внешний параметр.

Определение областей адекватности для конкретных моделей — сложная процедура, требующая больших вычислительных затрат. Эти затраты и трудности представления ОА быстро растут с увеличением размерности пространства внешних параметров. Определение ОА — более трудная задача, чем, например, задача параметрической

оптимизации, поэтому для моделей вновь проектируемых объектов ОА не рассчитывают.

Однако для моделей унифицированных элементов расчет областей адекватности становится оправданным в связи с однократностью определения ОА и многократностью их использования при проектировании различных систем. Знание ОА позволяет правильно выбирать модели элементов из числа имеющихся и тем самым повышать достоверность результатов машинных расчетов.

В общем случае ОА может иметь произвольную форму, сведения о которой выражаются громоздко, и неудобна в использовании, поэтому на практике вместо истинных ОА применяют те или иные их аппроксимации. Наиболее просто представляются и используются сведения об областях, имеющих форму гиперпараллелепипеда, который задается  $p$  двусторонними неравенствами:

$$q_k' \leq q_k \leq q_k'', k = \overline{1, p},$$

где  $p$  — размерность пространства внешних параметров.

Так, для одной из возможных макромоделей логического элемента транзисторно-транзисторной логики, реализующего функцию И—НЕ, рассчитанная область адекватности выражается следующими неравенствами:

$$4,47\text{В} \leq E \leq 5,82\text{В}; 3,89 \text{ нс} \leq t_{\text{ф.вх}} \leq 11,2 \text{ нс}; \\ 0 \leq N \leq 3; 3 \text{ нс} \leq \tau_{\text{вх}} \leq 6 \text{ нс},$$

где  $E$  — напряжение питания;  $t_{\text{ф.вх}}$  — длительность фронта входного сигнала;  $N$  — коэффициент нагружения;  $\tau_{\text{вх}}$  — длительность входного сигнала.

В библиотеку моделей элементов наряду с алгоритмом, реализующим модель, и номинальными значениями параметров должны включаться граничные значения внешних параметров  $q_k'$  и  $q_k''$ , задающие область адекватности.

На рис. 4.2 дано графическое представление области адекватности и аппроксимирующего ее гиперпараллелепипеда. Такое представление удобно для двумерных случаев.

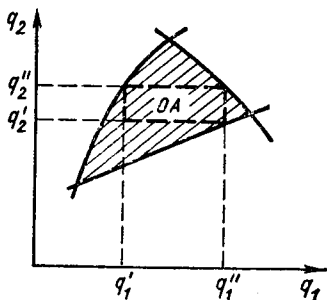


Рис. 4.2. Пример области адекватности

Возможно использование и других аппроксимаций ОА, например областей с линеаризованными границами в виде участков гиперплоскостей, областей в форме гиперсфер и т. п.

**Универсальность.** При определении ОА необходимо выбрать совокупность внешних параметров и совокупность выходных параметров  $y_j$ , отражающих учитываемые в модели свойства. Типичными внешними параметрами при этом являются параметры нагрузки и внешних воздействий (электрических, механических, тепловых, радиационных и т. п.). Увеличение числа учитываемых внешних факторов расширяет применимость модели, но существенно удорожает работу по определению ОА. Выбор совокупности выходных параметров также неоднозначен, однако для большинства объектов число и перечень учитываемых свойств и соответствующих им выходных параметров сравнительно невелики, достаточно стабильны и составляют типовой набор выходных параметров. Например, для макромоделей логических элементов БИС такими выходными параметрами являются уровни выходного напряжения в состояниях логических «0» и «1», запасы помехоустойчивости, задержка распространения сигнала, рассеиваемая мощность.

Если адекватность характеризуется положением и размерами ОА, то универсальность модели определяется числом и составом учитываемых в модели внешних и выходных параметров.

**Экономичность.** Экономичность модели характеризуется затратами вычислительных ресурсов для ее реализации, а именно затратами машинного времени  $T_m$  и памяти  $P_m$ . Общие затраты  $T_m$  и  $P_m$  на выполнение в САПР какой-либо проектной процедуры зависят как от особенностей выбранных моделей, так и от методов решения.

В большинстве случаев при реализации численного метода происходят многократные обращения к модели элемента, входящего в состав моделируемого объекта. Тогда удобно экономичность модели элемента характеризовать затратами машинного времени, получающимися при обращении к модели, а число обращений к модели должно учитываться при оценке экономичности метода решения.

Экономичность модели по затратам памяти оценивается объемом оперативной памяти, необходимой для реализации модели.

Требования широких областей адекватности, высокой степени универсальности, с одной стороны, и высокой эко-

номичности, с другой, являются противоречивыми. Наилучшее компромиссное удовлетворение этих требований оказывается неодинаковым в различных применениях. Это обстоятельство обуславливает использование в САПР многих моделей для объектов одного и того же типа — различного рода макромоделей, многоуровневых, смешанных моделей и т. п.

#### 4.3. МЕТОДЫ ПОЛУЧЕНИЯ МОДЕЛЕЙ ЭЛЕМЕНТОВ

Получение моделей элементов (моделирование элементов) в общем случае — процедура неформализованная. Основные решения, касающиеся выбора вида математических соотношений, характера используемых переменных и параметров, принимает проектировщик. В то же время такие операции, как расчет численных значений параметров модели, определение областей адекватности и другие, алгоритмизированы и решаются на ЭВМ. Поэтому моделирование элементов обычно выполняется специалистами конкретных технических областей с помощью традиционных средств экспериментальных исследований и средств САПР.

Методы получения функциональных моделей элементов делят на теоретические и экспериментальные. *Теоретические методы* основаны на изучении физических закономерностей протекающих в объекте процессов, определении соответствующего этим закономерностям математического описания, обосновании и принятии упрощающих предположений, выполнении необходимых выкладок и приведении результата к принятой форме представления модели. *Экспериментальные методы* основаны на использовании внешних проявлений свойств объекта, фиксируемых во время эксплуатации однотипных объектов или при проведении целенаправленных экспериментов.

Несмотря на эвристический характер многих операций моделирования имеется ряд положений и приемов, общих для получения моделей различных объектов. Достаточно общий характер имеют методика макромоделирования, математические методы планирования экспериментов, а также алгоритмы формализуемых операций расчета численных значений параметров и определения областей адекватности.

**Методика макромоделирования.** Применение методики состоит из следующих этапов:

1. Определение тех свойств объекта, которые должны отражаться моделью (устанавливаются требования к степени универсальности будущей модели).

2. Сбор априорной информации о свойствах моделируемого объекта. Примерами собираемых сведений могут служить справочные данные, математические модели и результаты эксплуатации существующих аналогичных объектов и т. п.

3. Получение общего вида уравнений модели (структуры модели). Этот этап в случае теоретических методов включает выполнение всех присущих этим методам операций, перечисленных выше. Часто проектировщику модели удобнее оперировать не уравнениями, а эквивалентными схемами, с помощью которых инженеру проще устанавливать физический смысл различных элементов математической модели.

4. Определение численных значений параметров модели. Возможны следующие приемы выполнения этого этапа: а) использование специфических расчетных соотношений с учетом собранных на этапе 2 сведений; б) решение экстремальной задачи, в которой в качестве целевой функции выбирается степень совпадения известных значений выходных параметров объекта с результатами использования модели, а управляемыми параметрами являются параметры модели; в) проведение экспериментов и обработка полученных результатов.

5. Оценка точности полученной модели и определение области ее адекватности. При неудовлетворительных точностных оценках выполняют итерационное приближение к желаемому результату повторением этапов 3—5.

6. Представление полученной модели в форме, принятой в используемой библиотеке моделей.

**Методы планирования экспериментов.** Для целей моделирования используют пассивные и активные эксперименты. В *пассивных экспериментах* нет возможности выбирать условия опыта по своему усмотрению и устанавливать значения факторов на желаемом уровне. В *активных экспериментах* опыты проводятся по заранее разработанному плану, выражающему количество опытов и значения факторов в каждом опыте.

Выбор вида зависимости выходного параметра макро-модели  $y$  (в общем случае рассматривается вектор выходных параметров  $Y$ ) от внешних параметров  $q_k$ , объединенных в вектор факторов  $Q$ , осуществляется проектировщи-

ком. Чаще всего в методах планирования эксперимента используются модели линейные

$$y = \mathbf{A}\mathbf{Q} \quad (4.4)$$

или квадратичные

$$y = \mathbf{A}\mathbf{B}, \quad (4.5)$$

где  $\mathbf{A}$  — вектор-строка коэффициентов (параметров) модели;  $\mathbf{B}$  — вектор, включающий факторы  $q_k$ , те или иные произведения из двух, трех или более факторов и возможно также квадраты факторов  $q_k^2$ ;  $k = \overline{1, p}$ ;  $p$  — число факторов.

Число опытов  $N$ , как правило, должно превышать число определяемых параметров вектора  $\mathbf{A}$ . Параметры рассчитывают по методу наименьших квадратов, т. е. из условия минимизации суммы квадратов отклонений значений  $\tilde{y}_l$ , определенных по уравнению модели (4.4), и измеренных значений  $y_l$ :

$$\min_{\mathbf{A}} \sum_{l=1}^N (\tilde{y}_l(\mathbf{A}) - y_l)^2,$$

где  $l$  — номер опыта.

В зависимости от способов планирования преимущества активных экспериментов перед пассивными могут выражаться в получении оптимального положения области адекватности, в ее увеличенном объеме, в упрощении оценок точности и т. п.

**Регрессионный анализ.** Связь между  $y$  и  $\mathbf{Q}$  может быть не функциональной, а статистической, что особенно характерно при пассивных экспериментах. Для получения моделей в такой ситуации часто применяют регрессионный анализ. Модель ищется в форме уравнения регрессии (4.4), в котором роль коэффициентов  $a_k$  в векторе  $\mathbf{A}$  выполняют коэффициенты относительной регрессии.

Рассмотрим алгоритм вычисления коэффициентов  $a_k$ . По результатам пассивных экспериментов получают оценки математических ожиданий  $M_y$ ,  $M_k$ , среднеквадратичных отклонений  $\sigma_y$ ,  $\sigma_k$  соответственно для выходного  $y$  и внешних  $q_k$  параметров, а также коэффициенты корреляции  $r_k$  между  $y$  и  $q_k$ , образующие вектор  $\mathbf{R}$ , и коэффициенты корреляции  $d_{kj}$  между факторами  $q_k$  и  $q_j$ , образующие матрицу  $\mathbf{D}$ . Далее решается система линейных алгебраических уравнений

$$\mathbf{D}\boldsymbol{\eta} = \mathbf{R} \quad (4.6)$$



и полученный вектор  $\eta = (\eta_1, \eta_2 \dots \eta_p)$  используется при расчете относительных коэффициентов регрессии по формуле

$$a_k = \eta_k \sigma_y / \sigma_k.$$

Если факторы  $q_k$  некоррелированы, то  $D$  — единичная матрица и можно обойтись без решения системы (4.6), так как  $\eta_k = r_k$ .

**Диалоговое моделирование.** Наличие в методике макро-моделирования эвристических и формальных операций обусловливает целесообразность разработки моделей элементов в диалоговом режиме работы с ЭВМ. Язык взаимодействия человека с ЭВМ должен позволять оперативный ввод исходной информации о структуре модели, об известных характеристиках и параметрах объекта, о плане экспериментов. Диалоговое моделирование должно иметь программное обеспечение, в котором реализованы алгоритмы статистической обработки результатов экспериментов, расчета выходных параметров эталонных моделей и создаваемых макромоделей, в том числе расчета параметров по методам планирования экспериментов и регрессионного анализа, алгоритмы методов поиска экстремума, расчета областей адекватности и др. Пользователь, разрабатывающий модель, может менять уравнения модели, задавать их в аналитической, схемной или табличной форме, обращаться к нужным подпрограммам и тем самым оценивать результаты предпринимаемых действий, приближаясь к получению модели с требуемыми свойствами.

#### 4.4. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ, ИСПОЛЬЗУЕМЫЕ НА МИКРОУРОВНЕ

Математические модели деталей и процессов на микроуровне отражают физические процессы, протекающие в сплошных средах и непрерывном времени. Независимыми переменными в этих моделях являются пространственные координаты и время. В качестве зависимых переменных выступают фазовые переменные, такие как потенциалы, напряженности полей, концентрации частиц, деформации и т.п. Взаимосвязи переменных выражаются с помощью уравнений математической физики — интегральных, интегродифференциальных или дифференциальных уравнений в частных производных. Эти уравнения составляют основу ММ на микроуровне.

Для получения законченной математической модели, используемой в задачах проектирования, необходимо дополнительно выполнить ряд процедур:

выбрать краевые условия. Краевые условия представляют собой сведения о значениях фазовых переменных и (или) их производных на границах рассматриваемых пространственных и временных областей;

дискретизировать задачу. Дискретизация подразумевает разделение рассматриваемых пространственных и временных областей на конечное число элементарных участков с представлением фазовых переменных конечным числом значений в избранных узловых точках, принадлежащих элементарным участкам;

алгебраизировать задачу — аппроксимировать дифференциальные и интегральные уравнения алгебраическими.

Используют два основных подхода к дискретизации и алгебраизации краевых задач, составляющие сущность *методов конечных разностей (МКР)* и *конечных элементов (МКЭ)*. С помощью любого из этих методов формируется окончательная модель, исследуемая при выполнении различных процедур анализа проектируемого объекта.

Пользователь САПР средствами входного языка задает исходную информацию о конфигурации проектируемого объекта, о способе дискретизации — разделения среды на элементы, о физических свойствах участков среды. Формирование модели объекта, т. е. разделение среды на элементы, выбор математических моделей элементов из заранее составленных библиотек, объединение моделей элементов в общую систему уравнений, так же как и решение получающихся уравнений, осуществляется автоматически на ЭВМ.

**Основные уравнения математической физики, используемые в моделях проектируемых объектов.** Процессы, протекающие в техническом объекте при его функционировании, по своей физической природе могут быть разделены на электрические, тепловые, магнитные, оптические, механические, гидравлические и т. п. Каждому типу процессов в математической модели соответствует своя подсистема, основанная на определенных уравнениях математической физики. Рассмотрим примеры уравнений, составляющих основу математических моделей технических объектов на микроуровне.

Электрические процессы в современных полупроводниковых приборах с достаточной точностью удается описать с помощью уравнений непрерывности и Пуассона. Уравнения непрерывности выражают ско-

рости изменения концентраций свободных носителей заряда и записываются отдельно для дырок и электронов:

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \operatorname{div} \mathbf{J}_p + g_p; \quad (4.7)$$

$$\frac{\partial n}{\partial t} = \frac{1}{q} \operatorname{div} \mathbf{J}_n + g_n, \quad (4.8)$$

где  $p$  и  $n$  — концентрации дырок и электронов соответственно;  $q$  — заряд электрона;  $g_p$  и  $g_n$  — скорости процесса генерации-рекомбинации соответственно дырок и электронов;

$$\mathbf{J}_p = q(-\mu_p p \operatorname{grad} \varphi - D_p \operatorname{grad} p); \quad (4.9)$$

$$\mathbf{J}_n = q(-\mu_n n \operatorname{grad} \varphi + D_n \operatorname{grad} n) \quad (4.10)$$

— плотности дырочного и электронного токов;  $\mu_p$ ,  $\mu_n$  — подвижности;  $D_p$ ,  $D_n$  — коэффициенты диффузии дырок и электронов;  $\varphi$  — электрический потенциал.

Уравнения (4.7)–(4.8) показывают, что причинами изменения концентрации носителей могут быть неодинаковость числа носителей, вытекающих (и вытекающих) в элементарный объем полупроводника (тогда  $\operatorname{div} \mathbf{J} \neq 0$ ), и нарушение равновесия между процессами генерации и рекомбинации носителей. Уравнения (4.9) и (4.10), называемые уравнениями плотности тока, характеризуют причины протекания электрического тока в полупроводнике: электрический дрейф под воздействием электрического поля ( $\operatorname{grad} \varphi \neq 0$ ) и диффузию носителей при наличии градиента концентрации. Уравнение Пуассона характеризует зависимость изменений в пространстве напряженности электрического поля  $\mathbf{E} = -\operatorname{grad} \varphi$  от распределения плотности электрических зарядов  $\rho$ :

$$\operatorname{div} \mathbf{E} = \rho / (\epsilon \epsilon_0),$$

где  $\epsilon$  — относительная диэлектрическая проницаемость среды;  $\epsilon_0$  — диэлектрическая постоянная.

В качестве краевых условий в моделях полупроводниковых приборов используют зависимости потенциалов на контактах от времени, принимают значения концентраций носителей на границе между внешним выводом и полупроводником равными равновесным концентрациям  $p_0$  и  $n_0$ , для границ раздела полупроводника и окисла задаются скоростью поверхностной рекомбинации  $g_s$ , что определяет величины нормальных к поверхности раздела составляющих плотностей тока  $\mathbf{J}_p$  и  $\mathbf{J}_n$ , и т. д.

Результат решения уравнений непрерывности и Пуассона при известных краевых условиях — это поля потенциала и концентраций подвижных носителей в различных областях полупроводниковой структуры. Знание этих полей позволяет оценить электрические параметры прибора.

В основе моделей диффузионных процессов, используемых, в частности, для описания технологических операций диффузии примесей при изготовлении интегральных схем и полупроводниковых приборов, лежит уравнение диффузии

$$\partial N / \partial t = \operatorname{div} (D \operatorname{grad} N),$$

где  $N$  — концентрация примеси;  $D$  — коэффициент диффузии.

Краевые условия представлены зависимостью распределения примеси  $N$  в объеме полупроводника в начальный момент времени и зависимостью поверхностной концентрации от времени.

На использовании закономерностей протекания тепловых процессов основано действие многих теплофизических установок. В РЭА полезные свойства обусловлены закономерностями электрических процессов, однако рассеяние мощности и изменения температуры оказывают заметное влияние на характер функционирования аппаратуры. Поэтому в моделях РЭА, как и в моделях многих устройств иной природы, приходится учитывать тепловые процессы. Теплоперенос в твердых телах описывается уравнением теплопроводности

$$C_p \frac{\partial T}{\partial t} = \operatorname{div} (\lambda \operatorname{grad} T) + g_Q,$$

где  $T$  — температура;  $C$  — удельная теплоемкость;  $\rho$  — плотность;  $\lambda$  — коэффициент теплопроводности;  $g_Q$  — количество теплоты, выделяемой в единицу времени в единице объема.

При проектировании различных изделий машиностроения важное значение для определения выходных параметров имеет возможность исследования напряженно-деформированных состояний конструкций. Подобные исследования важны и при проектировании несущих конструкций, механических и электромеханических устройств в составе РЭА. Знание напряжений и деформаций позволяет оценить прочность, долговечность, виброустойчивость аппаратуры.

Модели для анализа напряжений и упругих деформаций твердых тел формируют с помощью основного уравнения теории упругости — уравнения Ламе. Это уравнение получается из условия равновесия сил, действующих на элемент твердого тела в направлении оси  $x_i$ :

$$\sum_{j=1}^3 \frac{\partial \sigma_{ij}}{\partial x_j} + g_{mi} = \rho \frac{\partial^2 w_i}{\partial t^2}, \quad i = \overline{1, 3}. \quad (4.11)$$

где  $\rho$  — плотность;  $g_{mi}$  — проекция вектора  $G_m$  массовых сил, приходящихся на единицу объема, на ось  $x_i$ ;  $w_i$  — перемещение элемента вдоль оси  $x_i$ ;  $\sigma_{ij}$  — напряжение, действующее вдоль оси  $x_i$  в грани элемента, перпендикулярной оси  $x_j$ . Напряжения  $\sigma_{ij}$  связаны с деформациями  $\epsilon_{ij}$ , а последние — с перемещениями. В случае линейности связи

$\sigma_{ij}$  и  $\varepsilon_{ij}$  имеем

$$\sigma_{ii} = \lambda \sum_{j=1}^3 \varepsilon_{jj} + 2\mu \varepsilon_{ii}, \quad (4.12a)$$

а при  $i \neq j$  имеем

$$\sigma_{ij} = 2\mu \varepsilon_{ij}, \quad (4.12b)$$

где  $\lambda = E_0 \nu / [(1+\nu)(1-2\nu)]$  и  $\mu = E_0 / [2(1+\nu)]$  — постоянные Ламе, характеризующие упругие свойства среды;  $E_0$  — модуль упругости;  $\nu$  — коэффициент Пуассона. Учитывая, что

$$\varepsilon_{ij} = \frac{1}{2} \left( \frac{\partial w_i}{\partial x_j} + \frac{\partial w_j}{\partial x_i} \right), \quad (4.13)$$

подстановкой (4.12)–(4.13) в (4.11) получаем уравнение Ламе

$$(\lambda + \mu) \operatorname{grad} \operatorname{div} \mathbf{W} + \mu \Delta \mathbf{W} + \mathbf{G}_M = \rho \frac{\partial^2 \mathbf{W}}{\partial t^2}, \quad (4.14)$$

где  $\mathbf{W}$  — вектор перемещений;  $\Delta$  — оператор Лапласа.

Модели для анализа напряжений и деформаций часто оказываются более удобными, если представлены в интегральной форме, вытекающей из вариационных принципов механики. Вариационный принцип Лагранжа (принцип потенциальной энергии) гласит, что потенциальная энергия системы получает стационарное значение на тех кинематически возможных перемещениях, отвечающих заданным граничным условиям, которые удовлетворяют условиям равновесия. Поэтому модель представляют в виде выражения потенциальной энергии  $\Pi$  системы как разности энергии деформации  $\mathcal{E}$  и работы массовых и приложенных поверхностных сил  $A$ :

$$\Pi = \mathcal{E} - A,$$

где  $\mathcal{E} = 0,5 \int_{\mathbf{R}} \mathbf{e}^t \boldsymbol{\sigma} d\mathbf{R}$ ;  $\mathbf{e}^t = (\varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23})^t$  — вектор-строка деформаций;  $\boldsymbol{\sigma} = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})$  — вектор-столбец напряжений. Вводя матрицу

$$\mathbf{D} = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & 2\mu & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\mu & 0 \\ 0 & 0 & 0 & 0 & 0 & 2\mu \end{bmatrix},$$

можно (4.12) заменить более лаконичной записью:

$$\boldsymbol{\sigma} = \mathbf{D} \mathbf{e}.$$

Таким образом, при использовании принципа Лагранжа вместо ре-

шения уравнения (4.14) требуется минимизировать функционал

$$\Pi = 0,5 \int_{\mathbf{R}} \varepsilon' \mathbf{D} \varepsilon \, d\mathbf{R} - A. \quad (4.15)$$

В заключение рассмотрим основные уравнения газодинамики, лежащие в основе моделей разнообразных пневматических и гидравлических устройств. Уравнение закона сохранения массы называют *уравнением неразрывности*:

$$\partial \rho / \partial t = - \operatorname{div} (\rho \mathbf{U}),$$

где  $\rho$  — плотность;  $\mathbf{U}$  — вектор скорости.

Уравнение закона сохранения количества движения в случае идеальной жидкости называют *уравнением Эйлера*:

$$d(\rho \mathbf{U}) / dt = - \rho \mathbf{U} \operatorname{div} \mathbf{U} - \operatorname{grad} P. \quad (4.16)$$

где  $P$  — давление.

Учет массовых сил и сил трения приводит к появлению дополнительных членов в правой части уравнения (4.16), которое называют в этом случае *уравнением Навье — Стокса*.

Таким образом, основу большинства моделей объектов на микроуровне составляют дифференциальные уравнения. Далее будем использовать обобщенную форму записи уравнений

$$Lv(\mathbf{X}) = f(\mathbf{X}), \quad (4.17)$$

где  $L$  — дифференциальный оператор;  $v(\mathbf{X})$  — зависимая фазовая переменная;  $\mathbf{X} = (x_1, x_2, x_3)$  — вектор независимых (пространственных) координат;  $f(\mathbf{X})$  — заданная функция.

Нестационарные уравнения в операторной форме можно записывать как

$$\partial v / \partial t + Lv = f(\mathbf{X}, t) \quad (4.18)$$

или

$$\partial^2 v / \partial t^2 + Lv = f(\mathbf{X}, t), \quad (4.19)$$

где  $v$  — функция  $\mathbf{X}$  и времени  $t$ . Например, для двумерного уравнения диффузии

$$\partial N / \partial t - D (\partial^2 N / \partial x_1^2 + \partial^2 N / \partial x_2^2) = 0$$

используем запись  $\partial N / \partial t = LN$ , где

$$L = D (\partial^2 / \partial x_1^2 + \partial^2 / \partial x_2^2).$$

При необходимости применим и более лаконичную запись нестационарных уравнений в виде

$$\Lambda v(\mathbf{X}, t) = f(\mathbf{X}, t),$$

где  $\Lambda$  — дифференциальный оператор, включающий дифференцирование по всем независимым переменным.

Математическая модель должна содержать не только уравнения типа (4.18) или (4.19), но и краевые условия, представленные уравнениями

$$\alpha_1 v + \alpha_2 \Lambda v = \varphi(\mathbf{X}, t), \quad (4.20)$$

где  $\alpha_1, \alpha_2$  — коэффициенты;  $\varphi(\mathbf{X}, t)$  — заданная функция.

Краевые условия состоят из граничных и начальных. Для граничных условий в (4.20) задается  $\mathbf{X} \in \Gamma$ , где  $\Gamma$  — граница рассматриваемой пространственной области. Для начальных условий в (4.20) задается  $t = t_{\text{нач}}$ , где  $t_{\text{нач}}$  — начальный момент времени.

**Основные положения метода конечных разностей.** Рассмотрим уравнение (4.17). Пусть  $\mathbf{X} \in \mathbf{R}$  ( $\mathbf{R}$  — ограниченная область изменения независимых переменных), заданы граничные условия.

Дискретизация задачи заключается в покрытии  $\mathbf{R}$  сеткой и замене множества  $\mathbf{R}$  конечным множеством точек  $\mathbf{X}_k$ , являющихся узлами сетки. Сетка может быть прямоугольной, косоугольной, с постоянными или переменными межузловыми расстояниями вдоль координатных осей (величинами шагов). Наиболее часто используют прямоугольную сетку с постоянными величинами шагов. На рис. 4.3 представлен фрагмент такой сетки для двумерной задачи с величинами шагов  $h_1$  и  $h_2$  вдоль координатных осей  $x_1$  и  $x_2$ .

Алгебраизация задачи заключается в замене дифференциального оператора  $\Lambda v$  разностным. Это означает, что непрерывная переменная  $v(\mathbf{X})$  заменяется конечным множеством значений  $v_k = v(\mathbf{X}_k)$  в узлах сетки, а производные  $\partial v / \partial \mathbf{X}$  аппроксимируются конечноразностными выражениями.

Применяется несколько способов выражения производных через значения  $v_k$ . Вид разностных операторов удобно представлять графически в форме шаблонов. На рис. 4.4, а—г даны примеры шаблонов для одномерных, а на рис. 4.4, д, ж — для двумерных стационарных задач. Шаблон представляет собой часть сетки, включающую множество узлов  $\mathbf{X}_k$ , значения переменных в которых используются при аппроксимации производных в заданном узле  $\mathbf{X}^*$ . Узлы  $\mathbf{X}_k$  на рис. 4.4 показаны темными кружками, а узел  $\mathbf{X}^*$  обведен дополнительной окружностью. В левой части рисунков указан аппроксимируемый дифференциальный оператор, а рядом с узлами сетки записаны значения коэф-

Рис. 4.3. Дискретизация пространства с помощью сетки

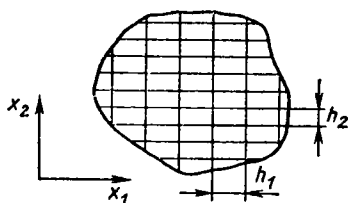


Рис. 4.4. Примеры шаблонов для разностных операторов

a)  $h \frac{d}{dx}$

б)  $h \frac{d}{dx}$

в)  $2h \frac{d}{dx}$

г)  $h^2 \frac{d^2}{dx^2}$

д)  $h^2 \nabla^2$

е)  $2h^2 \nabla^2$

ж)  $4h^2 \frac{\partial^2}{\partial x \partial y}$

коэффициентов, с которыми соответствующие величины  $v_k$  входят в конечноразностное аппроксимирующее выражение. При этом принято  $h_1 = h_2 = h$ . Так, для рис. 4.4, а получаем

$$h \frac{\partial v}{\partial x} \Big|_{x^*} = v_{i+1} - v_i,$$

для рис. 4.4, г

$$h^2 \frac{d^2 v}{dx^2} \Big|_{x^*} = v_{i+1} - 2v_i + v_{i-1}, \quad (4.21)$$



для рис. 4.4, б

$$h^2 \nabla^2 v|_{(x_{1i}, x_{2j})} = v_{i-1}^j + v_{i+1}^j + v_i^{j-1} + v_i^{j+1} - 4v_i^j \quad (4.22)$$

и т. д. В последнем выражении использовано обозначение  $v_i^j = v(x_{1i}, x_{2j})$ .

Подстановка выбранных аппроксимаций производных в исходное уравнение (4.17) преобразует его в систему разностных уравнений

$$F(V) = 0, \quad (4.23)$$

где  $V$  — вектор, элементами которого являются значения фазовой переменной во внутренних узлах сетки. Аналогичная подстановка в исходное нестационарное уравнение (4.18) преобразует его в систему обыкновенных дифференциальных уравнений (ОДУ)

$$dV/dt = F(V), \quad (4.24)$$

которая вместе с заданными начальными условиями представляет собой задачу Коши.

Следовательно, дискретизация и алгебраизация уравнений в МКР сводит задачу анализа моделей на микроуровне к численному решению систем конечных (4.23) или обыкновенных дифференциальных (4.24) уравнений. Следует отметить, что точность аппроксимации растет с уменьшением величин шагов, однако при этом увеличивается порядок систем уравнений (4.23) или (4.24). Так, если окажется, что для достижения приемлемой точности рассматриваемую область  $R$  нужно делить вдоль каждой из координатных осей на  $10^2$  участков, то порядки систем уравнений (4.23) или (4.24) в одно-, дву- и трехмерных задачах составляют соответственно около  $10^2$ ,  $10^4$  и  $10^6$ . Очевидно, что решение двумерных и особенно трехмерных задач требует значительных вычислительных ресурсов и тщательного отбора соответствующего математического обеспечения. Методы решения таких уравнений, применяемые в САПР, рассматриваются в следующей главе.

**Основные положения метода конечных элементов.** Рассмотрим применение МКЭ к решению задачи

$$Lv(X) = f(X).$$

Дискретизация исследуемой пространственной области  $R$  в МКЭ осуществляется ее разделением на непересекаю-

щиеся подобласти — конечные элементы (КЭ). В одномерных задачах КЭ представляют собой отрезки линий, в двумерных имеют форму треугольников, прямоугольников, в трехмерных — тетраэдров, параллелепипедов и т. п. В пределах каждого КЭ выбирают конечное число узловых точек  $X_k$ . Непрерывную фазовую переменную  $v$ , фигурирующую в модели (4.17), заменяют конечным числом значений  $v_k$  этой фазовой переменной в точках  $X_k$ .

Возможности использования КЭ различной формы, размеров и пространственной ориентации обуславливают легкость дискретизации граничных условий при произвольной форме области  $R$ . Это обстоятельство — одно из основных преимуществ МКЭ перед МКР, объясняющее широкое применение конечноэлементных представлений при моделировании процессов в деталях сложной конфигурации.

Второе отличие МКЭ от МКР заключается в способе алгебраизации дифференциальных уравнений  $Lv(X) = f(X)$ . Если в МКР аппроксимируются производные  $dv/dX$ , то в МКЭ аппроксимируется решение  $v(X)$  некоторой функцией  $u(X)$  с неопределенными коэффициентами. Решение исходной задачи получается путем вычисления этих коэффициентов. В свою очередь задача вычисления коэффициентов формулируется как задача минимизации функционала, характеризующего качество аппроксимации решения  $v(X)$  функцией  $u(X)$ , а эта задача сводится к решению системы алгебраических уравнений.

Очевидно, что чем сложнее применяемые аппроксимирующие функции и чем шире класс этих функций, тем сложнее задача формализации метода и его реализации в САПР. Особенностью МКЭ является выбор аппроксимирующих функций для каждого КЭ в отдельности. Малые размеры КЭ позволяют использовать простые аппроксимирующие функции, причем одного и того же типа для всех КЭ определенной формы. Обычно в качестве  $u(X)$  для отдельного КЭ применяют полиномы степени не выше третьей, например в одномерном случае

$$u(x) = \sum_{i=0}^r a_i x^i.$$

В МКЭ  $u(X)$  представляют в форме

$$u(\mathbf{X}) = \sum_{i=0}^r q_i \varphi_i(\mathbf{X}), \quad (4.25)$$

где коэффициенты  $q_i$  имеют вполне определенный физический смысл — это значения аппроксимирующей функции в узловых точках;  $\varphi_i(\mathbf{X})$  — функции, называемые координатными (базисными, пробными или функциями формы);  $r$  — число узловых точек в конечном элементе.

В общем случае аппроксимации вектора  $\mathbf{V}(\mathbf{X})$  в  $m$ -мерном пространстве выражение (4.25) принимает вид

$$\mathbf{U}(\mathbf{X}) = \mathbf{N}\mathbf{Q}, \quad (4.26)$$

где  $\mathbf{U}$  — вектор размера  $m \times 1$ ;  $\mathbf{Q}$  — вектор размера  $(mr) \times 1$ ;  $\mathbf{N}$  — интерполяционная матрица размера  $m \times (mr)$ , ее элементами являются координатные функции.

Важной составной частью МКЭ является выбор функционала, характеризующего качество используемой аппроксимации.

Примером такого функционала может служить следующий:

$$I = \int_{\mathbf{R}} (L\mathbf{U}(\mathbf{X}) - f(\mathbf{X}))^2 d\mathbf{R}, \quad (4.27)$$

где  $\mathbf{X} \in \mathbf{R}$ ,  $\mathbf{R}$  — исследуемая область. Очевидно, что интеграл  $I$ , относящийся ко всей области  $\mathbf{R}$ , получается суммированием интегралов, подобных (4.27), но относящихся к отдельным КЭ.

Иногда математические модели объектов на микроуровне уже в своем исходном виде могут быть представлены в вариационной формулировке, т. е. в виде задачи минимизации функционала. Типичным примером таких моделей служат модели, описывающие статические напряженно-деформированные состояния деталей. В этих моделях в качестве минимизируемого функционала используется выражение полной потенциальной энергии (4.15)

$$\Pi = \frac{1}{2} \int_{\mathbf{R}} \mathbf{e}^t \mathbf{D} \mathbf{e} d\mathbf{R} - A, \quad (4.28)$$

а в качестве искомой экстремали функционала фигурирует вектор перемещений  $\mathbf{W}(\mathbf{X})$ ,  $\mathbf{X} \in \mathbf{R}$ . Деформации  $e_{ij}$  связаны с перемещениями  $w_i$  соотношениями (4.13), что можно выразить в матричной форме записью

$$\begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{23} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & 2 \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & 2 \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}, \quad (4.29)$$

или более лаконично

$$\varepsilon = SW, \quad (4.30)$$

где  $S$  — фигурирующая в (4.29) матрица-оператор дифференцирования. Подставляя (4.30) в (4.28) и заменяя вектор  $W$  его аппроксимацией  $U = NQ$  в соответствии с (4.26), получаем

$$P = 0,5 \int_R Q^t B^t DBQ dR - A,$$

где  $B = SN$ . Обозначим

$$K = \int_R B^t DB dR \quad (4.31)$$

и назовем *матрицей жесткости*. Тогда  $P = 0,5 Q^t K Q - A$ . В соответствии с принципом Лагранжа дифференцируем  $P$  по вектору  $Q$  и приравниваем результат нулю, получаем систему алгебраических уравнений

$$KQ = P, \quad (4.32)$$

где  $P = \partial A / \partial Q$  — вектор правых частей, называемый *вектором нагрузок*.

Матрица жесткости  $K$  всей исследуемой детали составляется из матриц жесткости  $K_{ij}$  отдельных КЭ. Матрицы  $K_{ij}$  несут информацию о конфигурации и упругих свойствах материала конечных элементов и подсчитываются по формуле (4.31), в которой при этом под  $R$  понимается подобласть, относящаяся к рассматриваемому КЭ.

Система уравнений типа (4.32) получается при применении МКЭ к решению и других стационарных уравнений.

При этом требуется минимизировать некоторый функционал, например (4.27), что также приводит к решению системы алгебраических уравнений

$$\partial I / \partial \mathbf{Q} = 0.$$

В случае нестационарных уравнений основные положения МКЭ — деление на КЭ, подбор аппроксимирующей функции и минимизируемого функционала — по-прежнему применяют по отношению к пространственной области. Тогда получаем систему обыкновенных дифференциальных уравнений (ОДУ)

$$d\mathbf{Q}/dt = \mathbf{F}(\mathbf{Q}),$$

решением которой являются зависимости фазовых переменных  $\mathbf{Q}(t)$  в узловых точках от времени.

#### 4.5. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ, ИСПОЛЬЗУЕМЫЕ НА МАКРОУРОВНЕ

**Компонентные и топологические уравнения.** Для одного и того же объекта (детали) на микро- и макроуровнях используют разные математические модели. На микроуровне ММ должна отражать внутренние по отношению к объекту процессы, протекающие в сплошных средах. На макроуровне ММ того же объекта служит для отражения только тех его свойств, которые характеризуют взаимодействие этого объекта с другими элементами в составе исследуемой системы.

Математические модели элементов на макроуровне получают одним из способов, рассмотренных в § 4.3. Математи-

Т а б л и ц а 4.1

Подсистема	Фазовая переменная	
	типа потенциала $U$	типа потока $I$
Механическая	Скорость	Сила
Механическая вращательная	Угловая скорость	Вращательный момент
Электрическая	Электрическое напряжение	Электрический ток
Тепловая	Температура	Тепловой поток
Гидравлическая и пневматическая	Давление	Расход

ческие модели систем (ММС) формируют из математических моделей элементов (ММЭ) с помощью методов, излагаемых ниже.

Уравнения, входящие в ММЭ, называют *компонентными*. Наряду с компонентными уравнениями в ММС обязательно входят уравнения, отражающие способ связи элементов между собой в составе системы и называемые *топологическими*. Топологические уравнения могут выражать законы сохранения, условия неразрывности, равновесия и т. п.

В используемых в САПР методах формирования ММС принято моделируемую систему представлять в виде совокупности физически однородных подсистем. Каждая подсистема описывает процессы определенной физической природы, например механические, электрические, тепловые, гидравлические. Как правило, для описания состояния одной подсистемы достаточно применять фазовые переменные двух типов — потенциала и потока. В первых столбцах табл. 4.1 конкретизированы типы фазовых переменных применительно к ряду встречающихся подсистем.

Особенностью топологических уравнений является то, что каждое из них связывает однотипные фазовые переменные, относящиеся к разным элементам системы. Примером могут служить уравнения законов Кирхгофа, записываемые относительно либо токов, либо напряжений ветвей. Для компонентных уравнений характерно то, что они связывают разнотипные фазовые переменные, относящиеся к одному элементу. Так, уравнение закона Ома связывает ток и напряжение резистора.

**Формы представления моделей.** Элементы подсистем могут быть простыми и сложными. Элемент называют *про-*

Параметры простых элементов

C		L		R
Масса		Гибкость (обратная величина — жесткость)		Механическое сопротивление
Момент инерции		Вращательная гибкость		Вращательное сопротивление
Электрическая емкость		Электрическая индуктивность		Электрическое сопротивление
Теплоемкость		—		Тепловое сопротивление
Гидравлическая емкость	ем-	Гидравлическая индуктивность	ин-	Гидравлическое сопротивление

стем, если соответствующая ему ММЭ может быть представлена в виде одного линейного уравнения, связывающего переменную типа потенциала  $U$  и переменную типа потока  $I$ , характеризующие состояние данного элемента.

В физически однородных подсистемах различают три типа простых элементов. Это элементы емкостного, индуктивного и резистивного типов. Соответствующие им ММЭ имеют вид

$$CdU/dt = I, LdI/dt = U, U = RI, \quad (4.33)$$

где  $C$ ,  $L$ ,  $R$  — параметры элементов, физический смысл которых поясняется в табл. 4.1.

Элементы подсистем в зависимости от числа однотипных фазовых переменных, входящих в ММЭ, делят на двухполюсники и многополюсники. *Двухполюсник* характеризуется парой переменных типа  $U$  и  $I$ , определяется так же, как простой элемент, если снять условие линейности уравнения. *Многополюсник* можно представить как совокупность взаимосвязанных двухполюсников.

Для представления математических моделей на макроуровне применяют несколько форм.

*Инвариантная форма* — представление модели в виде системы уравнений, записанной на общепринятом математическом языке, безотносительно к методу численного решения. Применительно к системам обыкновенных дифференциальных уравнений различают две инвариантные формы — нормальную и общую, определяемые тем, в каком виде — явном или неявном относительно вектора производных — представлена система.

Ряд форм модели получается при преобразовании ее уравнений на основе формул и требований выбранного численного метода решения. Так, численное решение дифференциальных уравнений как в частных производных, так и обыкновенных требует их предварительного преобразования — дискретизации и алгебраизации. Дискретизация заключается в замене непрерывных независимых переменных (времени и пространственных координат) дискретным множеством их значений.

*Алгебраизованная форма* — результат представления дифференциальных уравнений в полученных после дискретизации точках в алгебраизованном виде с помощью формул численного интегрирования. Ряд численных методов решения основан на линеаризации исходных уравнений.

*Линеаризованная форма* модели — представление ее

уравнений в линейном виде. Алгебраизация и линеаризация могут осуществляться по отношению ко всем или только избранным переменным, уравнениям или их частям, что увеличивает разнообразие возможных форм представления моделей.

Формы представления моделей определяются также используемыми языковыми средствами. Наряду с традиционным математическим языком применяют алгоритмические языки, а также те или иные графические изображения, облегчающие пользователю восприятие модели и приводящие к представлению модели в той или иной *схемной* форме, например представление моделей в виде эквивалентных схем, графов, к таким формам относится также представление разностных уравнений с помощью шаблонов (см. § 4.4).

Рассмотрим особенности представления моделей в виде эквивалентных схем.

В разных областях техники применяют специфические системы обозначений элементов на эквивалентных схемах. Будем использовать в дальнейшем единую систему обозначений для элементов всех подсистем, обычно применяемую при изображении электрических эквивалентных схем. При этом элементы представляют собой двухполюсники, которые могут быть пяти различных видов, их условные обозначения приведены на рис. 4.5, а.

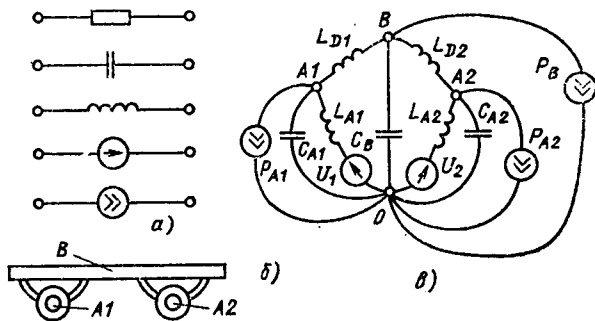


Рис. 4.5. Условные обозначения двухполюсных элементов

Получение эквивалентных схем — обычная для инженеров-схемотехников операция, выполняемая при анализе функционирования радиоэлектронных устройств. Переход от принципиальной электрической схемы к эквивалентной



заключается в замене обозначений электронных приборов обозначениями двухполюсников (рис. 4.5, а) и добавлении ветвей, отображающих учитываемые паразитные параметры. Не вызывает затруднений и составление на основе электрогидравлических и электротепловых аналогий эквивалентных схем, отражающих гидравлические, пневматические и тепловые процессы в проектируемых устройствах.

Составление эквивалентных схем для механических систем начинается с выбора системы координат, начало  $O$  которой должно быть связано с инерциальной системой отсчета. Далее формируются  $n$  эквивалентных схем, где  $n$  — число степеней свободы. В общем случае возможны три эквивалентные схемы, соответствующие поступательным движениям вдоль координатных осей, и три эквивалентные схемы, соответствующие вращательным движениям вокруг осей, параллельных координатным осям. Рассмотрим правила составления эквивалентных схем на примере одной из эквивалентных схем для поступательного движения: 1) для каждого тела  $A_i$  с учитываемой массой  $C_i$  в эквивалентной схеме выделяется узел  $i$  и между узлом  $i$  и узлом  $O$  включается двухполюсник массы  $C_i$ ; 2) трение между контактируемыми телами  $A_p$  и  $A_q$  отражается двухполюсником механического сопротивления, включаемым между узлами  $p$  и  $q$ ; 3) пружина, соединяющая тела  $A_p$  и  $A_q$ , а также другие упругие взаимодействия контактируемых тел  $A_p$  и  $A_q$  отражаются двухполюсником гибкости (жесткости), включаемым между узлами  $p$  и  $q$ .

В качестве примера на рис. 4.5, в приведена эквивалентная схема, моделирующая вертикальные скорости и усилия, возникающие в элементах движущегося транспортного устройства, условно изображенного на рис. 4.5, б в виде платформы  $B$  и колес  $A_1$  и  $A_2$ . Здесь учитываются массы платформы  $C_B$  и колес  $C_A$ , жесткости колес  $L_A$  и рессор  $L_D$ , а также веса  $P_B$ ,  $P_{A1}$ ,  $P_{A2}$  платформы и колес. Внешние воздействия отражены источниками скорости  $U$ .

Часто на эквивалентных схемах рядом с обозначением нелинейного элемента указан его тип или записано его компонентное уравнение.

Для отражения взаимосвязей подсистем различной физической природы, из которых состоит моделируемая техническая система, в эквивалентные схемы подсистем вводят специальные преобразовательные элементы. Различают три вида связей подсистем. Трансформаторная и гираторная связи выражают соотношения между фазовыми перемен-

ными двух подсистем, этим типам связей соответствуют преобразовательные элементы, представляемые парами источников тока или напряжения. Третий вид связи выражает влияние фазовых переменных одной подсистемы на параметры элементов другой и задается в виде зависимостей  $C$ ,  $L$  или  $R$  от фазовых переменных. Варианты эквивалентных схем трансформаторной (а) и гираторной (б, в) связей даны на рис. 4.6, где запись вида  $A(B)$  означает, что фазовая переменная  $A$  является функцией фазовой переменной  $B$ .

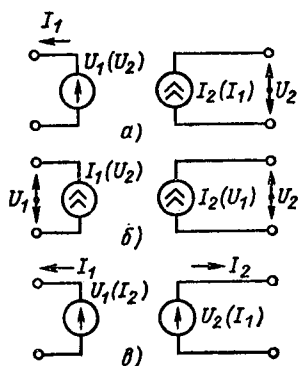
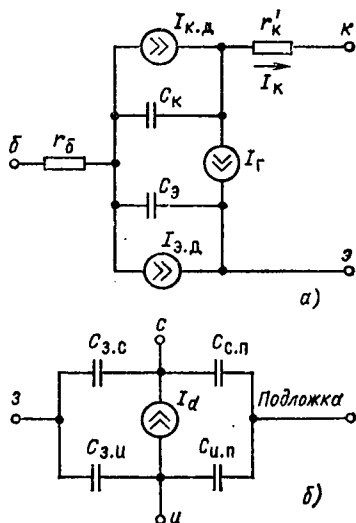


Рис. 4.6. Эквивалентные схемы трансформаторной (а) и гираторных (б, в) связей

Рис. 4.7. Эквивалентные схемы биполярного (а) и МДП (б) транзисторов



Так, в гидравлическом приводе связь механической и гидравлической подсистем является гираторной и соответствует рис. 4.6, б, если для источника объемного расхода в гидравлической подсистеме использовать выражение  $g = SV$ , а для источника силы в механической подсистеме — выражение  $F = SP$ , где  $V$  — скорость перемещения поршня;  $S$  — площадь поршня;  $P$  — давление жидкости в цилиндре.

**Примеры математических моделей элементов электронных схем.** Для конденсаторов, катушек индуктивности и резисторов чаще всего применяют простые модели (4.83). Примерами сложных элементов являются транзисторы, диоды, трансформаторы.

На рис. 4.7, а представлена эквивалентная схема биполярного транзистора, используемая во многих программах анализа электронных

схем. Этой схеме соответствуют следующие компонентные уравнения:

$$dU_{C_3}/dt = (I_3 - I_r - I_{3,д})/C_3; \quad dU_{C_k}/dt = (I_k + I_r - I_{к,д})/C_k,$$

где  $C_3, C_k$  — емкости и  $I_r, I_{3,д}, I_{к,д}$  — токи переходов, являющиеся заданными функциями напряжений  $U_{c_3}$  и  $U_{c_k}$  на емкостях, а для  $r_6$  и  $r'_k$  используются уравнения закона Ома.

На рис. 4.7, б представлена эквивалентная схема МДП-транзистора, в которой  $I_d$  является функцией потенциалов на электродах прибора, а емкости между затвором и стоком  $C_{з.с}$ , затвором и истоком  $C_{з.и}$ , истоком и подложкой  $C_{и.п}$ , стоком и подложкой  $C_{с.п}$  считаются либо постоянными, либо зависящими от потенциалов электродов.

**Примеры математических моделей элементов систем неэлектрической природы.** Простыми элементами механических поступательных систем являются элементы массы и гибкости (жесткости). Математическая модель массы выражает закон Ньютона

$$F = mdU/dt,$$

где  $F$  — сила;  $m$  — масса;  $U$  — скорость.

Математическая модель упругого стержня получается из закона Гука

$$\sigma = E\Delta l/l, \quad (4.34)$$

где  $\sigma$  и  $\Delta l$  — напряжение и изменение длины стержня в продольном направлении;  $E$  — модуль упругости;  $l$  — длина стержня.

Так как  $\sigma = F/S$ , где  $S$  — площадь поперечного сечения стержня, то после дифференцирования (4.34) по времени имеем

$$dF/dt = U/L_M,$$

где  $L_M = l/(SE)$  — гибкость стержня (напомним, что обратная величина гибкости — жесткость).

Сочетания элементов массы, гибкости и механического трения образуют разнообразные сложные элементы многих механических систем, например манипуляторов.

В качестве примера на рис. 4.8 приведена эквивалентная схема плоского сложного элемента «шарнирная связь двух твердых тел», где  $C1, C2$  — массы, а  $C3, C4$  — моменты инерции соединенных тел. Математическая модель представляет собой систему уравнений, отражающих геометрические соотношения, действующие в системе шарнирно связанных тел:

$$U_{x1} - a_1\omega_1 - U_{x2} + a_2\omega_2 = 0; \quad U_{y1} + b_1\omega_1 - U_{y2} - b_2\omega_2 = 0, \quad (4.35)$$

где  $U_x$  и  $U_y$  — проекции вектора скорости центра масс на оси координат  $x$  и  $y$  (индексы 1 и 2 относят соответствующую величину к первому или второму телу);  $\omega$  — угловая скорость относительно центра масс тела;  $a_i$  и  $b_i$  — коэффициенты, зависящие от угла поворота  $i$ -го тела в выбранной системе координат. Реакции в шарнире  $I_x$  и  $I_y$  определяются в результате решения системы уравнений, состоящей из компонентных уравнений (4.35), компонентных уравнений других элементов и топологических уравнений. Реакции  $I_x$  и  $I_y$  фигурируют в топологических уравнениях, составляемых для внешних узлов эквивалентной схемы. В случае связи тел с помощью упругой тяги в эквивалентной схеме появляется элемент гибкости, при учете трения — элемент механического сопротивления.

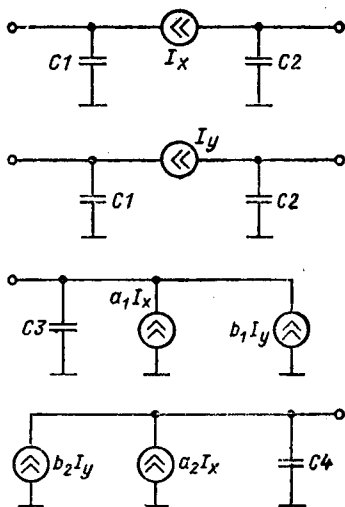


Рис. 4.8. Эквивалентная схема «шарнирная связь двух твердых тел»

Процессы теплопереноса в твердых телах отображаются элементами теплопроводности и теплоемкости. Математическая модель теплового сопротивления вытекает из уравнения Фурье

$$\mathbf{J} = -\lambda \text{grad } T,$$

где  $\mathbf{J}$  — плотность теплового потока;  $\lambda$  — коэффициент теплопроводности;  $T$  — температура.

Рассматривая распределение теплоты вдоль одной координатной оси по телу с площадью поперечного сечения  $S$  и длиной  $l$ , имеем тепловой поток

$$I = U/R_T, \quad (4.36)$$

где  $U$  — перепад температур на участке длиной  $l$ ;  $R_T = l/(S\lambda)$  — тепловое сопротивление.

Теплоперенос с помощью конвекции и лучеиспускания на макроуровне описывается также уравнениями вида (4.36) с той разницей, что величины  $R_T$  определяются иначе, чем в случае теплопроводности.

Из определения удельной теплоемкости

$$C_{уд} = \frac{1}{\rho} \frac{dQ}{dT}$$

следует уравнение теплоемкости

$$I = dQ/dt = C_T dU/dt. \quad (4.37)$$

Здесь  $Q$  — количество теплоты;  $\rho$  — плотность;  $C_T = \rho C_{уд}$  — теплоемкость тела.

Из элементов с компонентными уравнениями (4.36) и (4.37) составляются эквивалентные схемы для анализа тепловых процессов во многих объектах, например в конструкциях РЭА.

В гидравлических системах наличие вязкого трения обуславливает появление в эквивалентных схемах гидравлического сопротивления. Математическая модель гидравлического сопротивления для участка трубопровода круглого сечения при ламинарном течении жидкости имеет

$$U = R_r I,$$

где  $U$  — перепад давлений на рассматриваемом участке длиной  $l$  и радиусом  $r$ ;  $I$  — массовый поток жидкости;  $R_r = 8\nu l / (\pi r^4)$ ;  $\nu$  — кинематическая вязкость.

При турбулентном характере течения жидкости

$$U = R_r I |I|; \quad R_r = \lambda l / (4\pi^2 r^5),$$

где  $\lambda = 0,37 \sqrt[4]{\pi r \nu / |I|}$ .

Кроме сопротивления участок трубопровода характеризуется гидравлической индуктивностью  $L_r = l/S$ , а любая полость в гидравлической системе может рассматриваться как гидравлическая емкость:

$$C_r = Sl\rho/E.$$

Поэтому трубопровод в гидравлической и пневматической системах можно представлять состоящим из нескольких

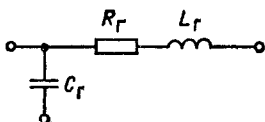


Рис. 4.9. Эквивалентная схема одной секции трубопровода

секций, заменяя каждую секцию эквивалентной схемой рис. 4.9. Пользователь, учитывая конкретные условия, может пренебрегать какими-либо из элементов  $R_r$ ,  $C_r$  или  $L_r$  в отдельных секциях и выбирать нужное число секций.

#### 4.6. ПОЛУЧЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ СИСТЕМ НА МАКРОУРОВНЕ

Математическая модель системы (ММС) на макроуровне представляет собой систему обыкновенных дифференциальных уравнений (ОДУ)

$$F\left(\frac{dV}{dt}, V, t\right) = 0, \quad (4.38)$$

где  $V$  — вектор зависимых переменных, называемых также базисными или определяющими координатами;  $t$  — время.

При получении системы (4.38) исходными являются компонентные и топологические уравнения. Поскольку выбор как формы исходных топологических уравнений, так и формы итоговой модели неоднозначен, для получения ММС возможно применение ряда методов. В настоящее время используются три основные формы представления ММС (4.38) на макроуровне:

нормальная форма

$$dV/dt = \varphi(V, t),$$

ее применение удобно при ориентации на последующее численное решение явными методами;

линеаризованная форма

$$D \frac{dV}{dt} + HV = B, \quad (4.39)$$

где  $D$  и  $H$  — матрицы;  $B$  — вектор. Форма (4.39) используется в программах, открытых по отношению к методам интегрирования;

алгебраизованная и линеаризованная форма

$$Y_n V_n = B_n, \quad (4.40)$$

где  $Y$  — матрица;  $B$  — вектор. Их вид определяется выбранными методами алгебраизации и линеаризации. Обозначения  $Y_n$ ,  $V_n$ ,  $B_n$  используются для значений переменных  $Y$ ,  $V$ ,  $B$ , соответствующих моменту времени  $t_n$ .

В подавляющем большинстве современных программ анализа применяют форму (4.40). Для получения ММС в такой форме применяют методы узловых потенциалов (МУП) и табличные методы. В этих методах для алгебраизации реализуют одну из неявных разностных формул численного интегрирования

$$\left. \frac{dV}{dt} \right|_n = \eta_n V_n + \mu_n, \quad (4.41)$$

где  $(dV/dt)_n$  и  $V_n$  — значения векторов  $dV/dt$  и  $V$  в момент времени  $t_n$ ;  $n$  — номер шага интегрирования;  $\eta_n$  — коэффициент, зависящий от выбранного метода интегрирования и значения шага  $h_n$ ;  $\mu_n$  — вектор, зависящий также от значений векторов  $V$  на  $p$  предыдущих шагах интегрирования,  $p$  — порядок метода.

Линеаризацию выполняют с помощью разложения нелинейных элементов вектора  $F(dV/dt, V, t)$  в ряд Тейлора с сохранением в разложении только линейных членов.

Различия между МУП и табличными методами заключаются в выборе исходных топологических уравнений и вектора базисных координат.

Для получения ММС в нормальной форме наиболее приемлем метод переменных, характеризующих состояние системы, называемой обычно более коротко — метод переменных состояния (МПС).

**Метод узловых потенциалов.** Исходные топологические уравнения в МУП — уравнения закона токов Кирхгофа (ЗТК) или аналогичные им уравнения, выражающие равновесие переменных типа потока во всех узлах эквивалентной схемы, за исключением лишь одного узла, принимаемого за базовый,

$$A\mathbf{I} = 0, \quad (4.42)$$

где  $A$  — матрица инцидентий;  $\mathbf{I}$  — вектор токов.

Матрица инцидентий характеризует связи узлов и ветвей эквивалентной схемы. В матрице инцидентий  $i$ -я строка соответствует  $i$ -му узлу, а  $j$ -й столбец —  $j$ -й ветви дерева. Всего в матрице  $\alpha$  столбцов и  $\beta$  строк, где  $\alpha$  и  $\beta$  — число ветвей и узлов в эквивалентной схеме. Элемент матрицы  $a_{ij} = +1$ , если  $i$ -й узел инцидентен  $j$ -й ветви и положительное направление тока в этой ветви выбрано от  $i$ -го узла;  $a_{ij} = -1$  при тех же условиях инцидентности, но при противоположном направлении тока, иначе  $a_{ij} = 0$ .

Исходные компонентные уравнения в классическом варианте МУП должны иметь вид

$$\mathbf{I}_n = \mathbf{G} \left( \frac{dU}{dt} \Big|_n, \mathbf{U}_n, t_n \right), \quad (4.43)$$

где  $\mathbf{I}_n$  и  $\mathbf{U}_n$  — векторы токов и напряжений ветвей на  $n$ -м шаге.

Алгебраизация с помощью (4.41) приводит к системе уравнений

$$\mathbf{I}_n = \mathbf{G} [(\eta_n \mathbf{U}_n + \mu_n), \mathbf{U}_n, t_n].$$

После линеаризации относительно вектора  $U_n$  компонентные уравнения имеют вид

$$I_n = Y_n U_n + Q_n, \quad (4.44)$$

где  $Y_n = \partial I_n / \partial U_n$  — матрица проводимостей;  $Q_n$  — вектор, зависящий от значений вектора  $U$ , полученных на предыдущих шагах интегрирования.

В качестве основных базисных координат в МУП используют узловые потенциалы, вектор которых на  $n$ -м шаге обозначим  $\varphi_n$ . Отметим, что связь между векторами  $U_n$  и  $\varphi_n$  выражается с помощью матрицы инциденций

$$U_n = -A^t \varphi_n. \quad (4.45)$$

Подставив (4.45) в (4.44) и (4.44) в (4.42), имеем

$$-AY_n A^t \varphi_n + AQ_n = 0,$$

т. е. получена алгебраизованная и линеаризованная форма ММС в виде

$$Y_n \varphi_n = B_n, \quad (4.46)$$

где  $Y_n = AY_n A^t$  — матрица Якоби;  $B_n = AQ_n$  — вектор правых частей.

В классическом варианте МУП имеются ограничения на вид компонентных уравнений. Применительно к схемной форме представления моделей эти ограничения выражаются в недопустимости таких ветвей, как идеальные источники напряжения и любые ветви, параметры которых зависят от каких-либо токов. В модифицированном варианте МУП эти ограничения снимаются благодаря расширению вектора базисных координат — дополнительно к узловым потенциалам к базисным координатам относят также токи особых ветвей. Особыми ветвями при этом называют: 1) ветви источников напряжения; 2) ветви, токи которых являются управляющими (аргументами в выражениях для параметров зависимых ветвей); 3) индуктивные ветви.

Обозначим через  $I_1$  и  $I_2$  векторы токов, а через  $A_1$  и  $A_2$  — подматрицы инциденций узлов с неособыми и особыми ветвями соответственно. Тогда уравнение (4.42) перепишем в виде

$$A_1 I_1 + A_2 I_2 = 0. \quad (4.47)$$

Компонентные уравнения неособых ветвей имеют вид

$$I_1 = Y_1 U + Y_2 I_2 + Q,$$

где в дополнение к уравнению (4.44) в правой части фигу-



рирует слагаемое, определяемое токами особых ветвей. Здесь  $Y_1 = \partial I_1 / \partial U$ ,  $Y_2 = \partial I_1 / \partial U_2$ ;  $n$  — индекс, указывающий номер шага, опущен, но подразумевается, что все переменные величины относятся к  $n$ -му шагу интегрирования. Подстановка  $I_1$  в (4.47) дает подсистему из  $(\beta - 1)$ -го уравнения

$$-A_1 Y_1 A_1^t \varphi + (A_1 Y_2 + A_2) I_2 + A_1 Q = 0,$$

в которую входят  $\beta + \gamma - 1$  неизвестных ( $\gamma$  — число особых ветвей). Эта подсистема доопределяется с помощью  $\gamma$  компонентных уравнений особых ветвей.

**Алгоритм вычисления матрицы Якоби в методе узловых потенциалов.** Вычисление матрицы Якоби как матричного произведения  $AYA^t$  без учета разреженности матриц  $A$  и  $Y$  нерационально, так как приводит к излишне большим затратам машинного времени и памяти. Например, в схеме средней сложности, включающей  $\beta = 51$  и  $\alpha = 80$ , матрица  $A$  имеет размер  $50 \times 80$ , а матрица  $Y$  — размер  $80 \times 80$ , т. е. только эти две матрицы для хранения всех их элементов требуют около 10 000 ячеек памяти. В то же время статистические исследования показывают, что ненулевыми в этих матрицах оказываются лишь около 240 элементов. Поэтому на практике используют алгоритмы формирования матрицы Якоби, учитывающие сильную разреженность матриц  $A$  и  $Y$ .

Рассмотрим алгоритм вычисления матрицы Якоби в классическом варианте МУП. Этот же алгоритм используют в модифицированном варианте МУП для вычисления подматрицы  $A_1 Y A_1^t$ , входящей в матрицу Якоби. Исходная информация о матрицах  $A$  и  $Y$  в этом алгоритме задается в виде списка, где  $k$ -я строка списка соответствует  $k$ -й ветви эквивалентной схемы. В этой строке указываются номера  $i$  и  $j$  узлов, инцидентных  $k$ -й ветви, местонахождение (адрес) проводимости  $\partial I_k / \partial U_k$  (где  $I_k$  и  $U_k$  — ток и напряжение  $k$ -й ветви) и, если  $I_k$  зависит также от потенциала  $\varphi_p$  некоторого узла  $p$ , отличного от  $i$  и  $j$ , — местонахождение производной  $I_k$  по потенциалу этого узла. Каждая  $k$ -я строка приводит к добавлению проводимостей в некоторые клетки матрицы  $Y$ . Строки списка просматриваются поочередно. Просмотр  $k$ -й строки приводит к добавлению проводимости  $\partial I_k / \partial U_k$  в диагональные клетки  $y_{ii}$  и  $y_{jj}$  и к вычитанию этой проводимости из клеток  $y_{ij}$  и  $y_{ji}$ . Если имеется  $\partial I_k / \partial \varphi_p \neq 0$ , то эта производная добавляется в клетки  $y_{ip}$  и  $y_{jp}$ , причем если ток  $I_k$  направлен от узла  $j$  к узлу  $i$ ,

то добавление в  $y_{ip}$  происходит со знаком «плюс», а в  $y_{jp}$  — со знаком «минус». Пример формирования матрицы Якоби приведен ниже.

**Табличный метод.** В качестве базисных координат используют токи и напряжения всех ветвей схемы, а в качестве исходных топологических уравнений — уравнения Кирхгофа. Эти уравнения записывают для системы контуров и сечений, выбранной в схеме так, чтобы получить  $\alpha$  топологических линейно независимых уравнений, где  $\alpha$  — число ветвей в схеме. В этих уравнениях фигурируют 2  $\alpha$  неизвестных токов и напряжений, поэтому система уравнений доопределяется с помощью  $\alpha$  компонентных уравнений.

Применяемый способ выбора системы независимых контуров и сечений основан на построении фундаментального дерева в графе схемы. Используется полюсный граф, повторяющий структуру эквивалентной схемы. Фундаментальное дерево связного графа есть связный подграф, включающий  $\beta - 1$  ребро и не имеющий циклов. Ребра, вошедшие в дерево, образуют множество ветвей дерева (ВД), а остальные ребра — множество ветвей, называемых хордами (ВХ). Контуром  $k$ -й хорды называют подмножество ребер графа (ветвей схемы), входящих в замкнутый контур, образуемый при подключении  $k$ -й хорды к дереву. Сечения образуются следующим образом: отделим часть вершин графа от остальных с помощью замкнутой линии сечения, проведя ее так, чтобы ни одно ребро не пересекалось более одного раза и при этом пересекалась одна и только одна ветвь дерева. Следовательно, каждому сечению соответствует определенная ветвь дерева. На рис. 4.10, *а* для примера приведена некоторая схема, а на рис. 4.10, *б* — ее граф с выделенным жирными линиями фундаментальным деревом. Штрихом показаны линии сечения. Уравнения токов Кирхгофа для сечений ветвей дерева и напряжений Кирхгофа для контуров хорд образуют систему независимых топологических уравнений

$$\left. \begin{aligned} I_{в.д} - M^t I_x &= 0, \\ U_x + M U_{в.д} &= 0, \end{aligned} \right\} \quad (4.48)$$

где  $I_{в.д}$ ,  $I_x$ ,  $U_{в.д}$ ,  $U_x$  — векторы токов и напряжений ветвей и хорд соответственно;  $M$  — матрица контуров и сечений.

Деление ветвей на хорды и ветви дерева может выполняться по различным правилам. В табличном методе рекомендуется выбирать фундаментальное дерево так, что-

бы минимизировалось количество ненулевых элементов в матрице  $\mathbf{M}$ . Элемент  $m_{ij}$  матрицы  $\mathbf{M}$  равен  $\pm 1$ , если в контур  $i$ -й хорды входит  $j$ -я ветвь дерева, иначе  $m_{ij}=0$ . Знаки у  $m_{ij} \neq 0$  выбирают по специальному правилу:  $m_{ij}=+1$ , если положительные направления токов в  $i$ -й хорде и  $j$ -й ветви дерева при обходе контура совпадают;  $m_{ij}=-1$ ,

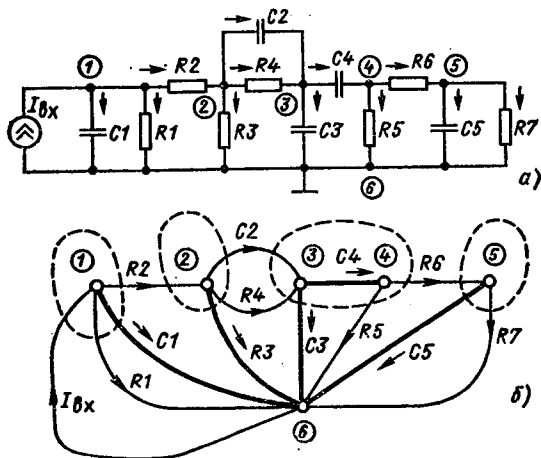


Рис. 4.10. Эквивалентная схема (а) и ее граф (б)

если эти направления не совпадают. В столбцах источников напряжения записываются  $+1$ , если ток подключенной хорды втекает в выбранный за положительный полюс источника.

На рис. 4.10 матрица  $\mathbf{M}$  имеет вид табл. 4.2.

Компонентные уравнения  $\mathbf{F}(\dot{\mathbf{V}}, \mathbf{V}, t) = 0$ , где  $\mathbf{V} = (\mathbf{U}_x, \mathbf{I}_{\text{в.д.}}, \mathbf{U}_{\text{в.д.}}, \mathbf{I}_x)$ ,  $\dot{\mathbf{V}} = d\mathbf{V}/dt$ , предварительно алгебраизуются с помощью (4.41) и линеаризуются относительно вектора  $\mathbf{V}_n$ :

$$\mathbf{W}\mathbf{V}_n = \mathbf{Q}, \quad (4.49)$$

где  $\mathbf{W} = \eta \partial \mathbf{F} / \partial \dot{\mathbf{V}} + \partial \mathbf{F} / \partial \mathbf{V}$ ;  $\mathbf{Q}$  — вектор правых частей.

Окончательно ММС представляет собой систему из  $2\alpha$  уравнений (4.48) и (4.49). Сравнительно высокий порядок получающейся системы уравнений — недостаток табличного метода.

**Метод переменных состояния.** Метод ориентирован на получение ММС в виде системы обыкновенных дифференциальных уравнений в нормальной форме с последующим

Таблица 4.2

	$C1$	$C3$	$C4$	$C5$	$R3$
$C2$	0	+1	0	0	-1
$R1$	-1	0	0	0	0
$R2$	-1	0	0	0	+1
$R4$	0	+1	0	0	-1
$R5$	0	-1	+1	0	0
$R6$	0	-1	+1	+1	0
$R7$	0	0	0	-1	0
$I_{вх}$	+1	0	0	0	0

ее интегрированием явными методами. В качестве базисных координат (переменных состояния) выбирают независимые емкостные напряжения и индуктивные токи, а в качестве исходных топологических уравнений — уравнения типа (4.48). Подмножество ветвей дерева ВД образуется путем включения в него ветвей схемы в следующем порядке — ветви источников напряжения, емкостные, резистивные, индуктивные. Именно такой приоритет ветвей обуславливает последующее получение ММС в форме либо нормальной, либо легко преобразуемой в нормальную. После выбора дерева и построения матрицы  $M$  систему (4.48) преобразуют — из нее исключают все токи и напряжения, не относящиеся к переменным состояниям. Исключения производят с помощью компонентных уравнений.

Наиболее просто такие исключения выполняют в том случае, если в схеме отсутствуют топологические вырождения (рис. 4.11), под которыми понимают емкостные контуры (а) и индуктивные звезды (б). К топологическим вырождениям относят также такие ситуации, при которых

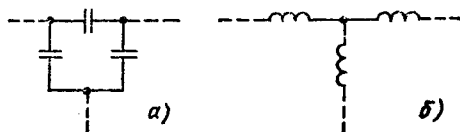


Рис. 4.11. Топологические вырождения

в контур резистивной хорды оказывается включенной хотя бы одна резистивная ветвь дерева. Наличие топологических вырождений усложняет процедуру получения ММС в нормальной форме и ее использование — требуется либо решение систем линейных алгебраических уравнений на каждом шаге численного интегрирования системы обыкновенных дифференциальных уравнений, либо предварительное устранение топологических вырождений с помощью изменений схемы. Очевидно, что последний прием может привести к дополнительным погрешностям решения или к ухудшению обусловленности системы уравнений.

Пример получения ММС различными методами. Для примера рассмотрим схему, показанную на рис. 4.10. Предполагается, что численное интегрирование уравнений будет выполняться с помощью метода первого порядка точности. Неявная формула такого метода

$$\dot{V}_n = (V_n - V_{n-1})/h_n, \quad (4.50)$$

где  $h_n$  — величина  $n$ -го шага интегрирования.

В методе узловых потенциалов компонентные уравнения

$$I_n = y_i U_n; \quad I = C_j dU_j/dt.$$

Последнее уравнение, алгебраизованное на  $n$ -м шаге интегрирования с помощью (4.50), принимает вид

$$I_n - y_j U_n = Q_n. \quad (4.51)$$

Здесь  $y_i = 1/R_i$ ;  $y_j = C_j/h_n$ ;  $Q_n = -C_j U_{n-1}/h_n$ .

Использование вышеописанного алгоритма формирования матрицы Якоби непосредственно приводит к ММС вида (4.52). Здесь  $y_p$  — проводимость ветви  $p$ ;  $\varphi_k$  — потенциал  $k$ -го узла на данном шаге интегрирования;  $\varphi'_k$  — то же на предыдущем шаге интегрирования.

В табличном методе полученная матрица  $M$  дана в табл. 4.3.

Итоговая ММС есть следующая система линейных алгебраических уравнений:

1	0	M	0
0	1	0	-M'
0	1	-Y <sub>в.д</sub>	0
-Y <sub>х</sub>	0	0	1

U <sub>х</sub>
I <sub>в.д</sub>
U <sub>в.д</sub>
I <sub>х</sub>

0
0
Q1
Q2

где  $I$  — единичная подматрица;  $Y_{в.д}$  и  $Y_x$  — диагональные подматрицы с проводимостями ветвей дерева и хорд соответственно на диагонали;  $Q1$  и  $Q2$  — подвекторы правых частей компонентных уравнений ветвей

$y_{R1} + y_{R2} + y_{C1}$	$-y_{R2}$	0	0	0
$-y_{R2}$	$y_{R2} + y_{R3} + y_{R4} + y_{C2}$	$-y_{R4} - y_{C2}$	0	0
0	$-y_{R4} - y_{C2}$	$y_{R4} + y_{C2} + y_{C3} + y_{C4}$	$-y_{C4}$	0
0	0	$-y_{C4}$	$y_{R5} + y_{R6} + y_{C4}$	$-y_{R6}$
0	0	0	$-y_{R6}$	$y_{R6} + y_{R7} + y_{C5}$

×

$\varphi_1$	$I_{BX} + y_{C1} \varphi_1'$
$\varphi_2$	$y_{C2} (\varphi_2' - \varphi_3')$
$\varphi_3$	$y_{C2} (\varphi_3' - \varphi_2') + y_{C3} \varphi_3' + y_{C4} (\varphi_3' - \varphi_4')$
$\varphi_4$	$y_{C4} (\varphi_4' - \varphi_3')$
$\varphi_5$	$y_{C5} \varphi_5'$

=

$\varphi_1$	$I_{BX} + y_{C1} \varphi_1'$
$\varphi_2$	$y_{C2} (\varphi_2' - \varphi_3')$
$\varphi_3$	$y_{C2} (\varphi_3' - \varphi_2') + y_{C3} \varphi_3' + y_{C4} (\varphi_3' - \varphi_4')$
$\varphi_4$	$y_{C4} (\varphi_4' - \varphi_3')$
$\varphi_5$	$y_{C5} \varphi_5'$

×

(4.52)

Таблица 4.3

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>
<i>R1</i>	-1	0	0	0	0
<i>R2</i>	-1	+1	+1	0	0
<i>R3</i>	0	-1	-1	0	0
<i>R4</i>	0	-1	0	0	0
<i>R5</i>	0	0	-1	+1	0
<i>R6</i>	0	0	-1	+1	+1
<i>R7</i>	0	0	0	0	-1
<i>I<sub>вх</sub></i>	-1	0	0	0	0

(4.51). Здесь первые две строки суть топологические уравнения (4.48), а две последующие — компонентные уравнения.

В методе переменных состояния граф и дерево, выбранное в соответствии с приоритетами ветвей, показаны на рис. 4.12. Матрица *M*

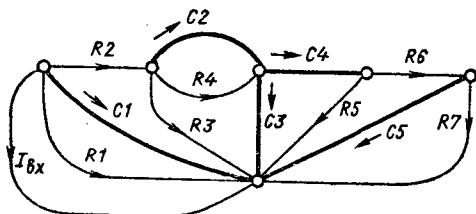


Рис. 4.12. Граф и выбранное дерево в методе переменных состояния

получается в виде табл. 4.3. Вектор правых частей  $\varphi(V, t)$  в системе уравнений  $\dot{V} = \varphi(V, t)$  с помощью полученной матрицы *M* находят по следующему алгоритму:

1. Вычисление вектора резистивных напряжений с помощью второго из уравнений (4.48).

$$U_x = -MU_{в.д.}$$

Это матричное уравнение удобно развернуть в следующую совокупность уравнений путем сканирования матрицы *M* по строкам:

$$U_{R1} = U_{C1};$$

$$U_{R2} = U_{C1} - U_{C2} - U_{C3};$$

$$U_{R3} = U_{C2} + U_{C3};$$

$$U_{R4} = U_{C2};$$

$$U_{R5} = U_{C3} - U_{C4};$$

$$U_{R6} = U_{C3} - U_{C4} - U_{C5};$$

$$U_{R7} = U_{C5}.$$

Отметим, что значения фигурирующих в правой части переменных состояния известны — на первом шаге это начальные условия, на каждом из последующих шагов — это значения, полученные на предыдущем шаге.

2. Вычисление вектора резистивных токов с помощью компонентных уравнений

$$I_{Ri} = U_{Ri} / R_i.$$

3. Вычисление вектора емкостных токов с помощью первого из топологических уравнений (4.48)

$$\mathbf{I}_{в.д} = \mathbf{M}' \mathbf{I}_{\mathbf{x}}.$$

Это матричное уравнение при вычислениях разворачивается в системе уравнений путем сканирования матрицы  $\mathbf{M}$  по столбцам:

$$I_{C1} = -I_{R1} - I_{R2} - I_{вх};$$

$$I_{C2} = I_{R2} - I_{R3} - I_{R4};$$

$$I_{C3} = I_{R2} - I_{R3} - I_{R5} - I_{R6};$$

$$I_{C4} = I_{R5} + I_{R6};$$

$$I_{C5} = I_{R6} - I_{R7}.$$

4. Вычисление вектора производных переменных состояния с помощью компонентных уравнений

$$dU_{Cj} / dt = I_{Cj} / C_j.$$

Далее применяют ту или иную формулу численного интегрирования, преобразующую вектор производных в вектор переменных состояния для очередного момента моделируемого времени, после чего переходят к новому шагу интегрирования.

#### 4.7. МАТЕМАТИЧЕСКИЕ МОДЕЛИ НА МЕТАУРОВНЕ

**Математические модели аналоговой РЭА.** Использование рассмотренных положений схмотехнического моделирования для проектирования сложной аналоговой РЭА на метауровне оказывается затруднительным из-за чрезмерно



больших размерностей задач. Необходимы упрощения. Основой снижения размерности задач является макро-моделирование. Часто используют ряд дополнительных упрощений и допущений. Главные из них формулируются следующим образом: 1. Однонаправленность в передаче сигналов, т. е. использование макромоделей, в которых отсутствует влияние выходных переменных на состояние входных цепей. 2. Отсутствие влияния нагрузки на параметры и состояние моделируемых систем. 3. Использование вместо фазовых переменных двух типов (напряжение и ток) переменных одного типа, называемых сигналами. При этом компонентные уравнения элемента представляют собой уравнения связи сигналов на входах и выходах этого элемента. 4. Линейность моделей инерционных элементов.

Перечисленные допущения характерны для функционального моделирования, широко используемого для анализа систем автоматического управления. Элементы (звенья) систем при функциональном моделировании делят на три группы: 1) линейные безынерционные звенья для отображения таких функций, как повторение, инвертирование, чистое запаздывание, идеальное усиление, суммирование сигналов; 2) нелинейные безынерционные звенья для отображения различных нелинейных преобразований сигналов (ограничение, детектирование, модуляция и т. п.); 3) линейные инерционные звенья для выполнения дифференцирования, интегрирования, фильтрации сигналов. Инерционные элементы представлены отношениями преобразованных по Лапласу или Фурье выходных и входных фазовых переменных. При анализе во временной области применяют преобразование Лапласа, модель инерционного элемента с одним входом и одним выходом есть передаточная функция, а при анализе в частотной области — преобразование Фурье, модель элемента есть выражения амплитудно-частотной и частотно-фазовой характеристик. При наличии нескольких входов и выходов ММ элемента представляется матрицей передаточных функций или частотных характеристик.

Рассмотрим примеры моделей элементов аналоговой РЭА. Анализ однокаскадного RC-усилителя, проводимый в курсах основ электроники, позволяет получить следующее приближенное выражение для передаточной функции каскада:

$$h(p) = \frac{K_0}{(1 + p\tau_B)(1 + 1/(p\tau_H))}$$

где  $K_0$  — коэффициент усиления на средних частотах;  $\tau_b$  и  $\tau_n$  — постоянные времени каскада на высоких и низких частотах соответственно.

Выражение частотной характеристики того же усилителя

$$h(j\omega) = \frac{K_0}{(1 + j\omega\tau_b)(1 + 1/(j\omega\tau_n))}.$$

Для однокаскадного резонансного LC-усилителя передаточная функция имеет вид

$$h(p) = S \frac{pL + r}{LCp^2 + Crp + 1}, \quad (4.53)$$

где  $S$  — крутизна усиления на резонансной частоте;  $L$ ,  $C$ ,  $r$  — индуктивность, емкость и последовательное сопротивление потерь параллельного колебательного контура.

Примером нелинейного элемента является амплитудный модулятор, для которого при функциональном моделировании используют модель в виде

$$U_{\text{вых}}(t) = U_m (1 + mU_{\text{вх}}(t)) \sin(\omega t + \varphi),$$

где  $U_{\text{вых}}$  и  $U_{\text{вх}}$  — напряжения модулированных и модулирующих колебаний;  $m$  — коэффициент модуляции;  $\omega$  и  $\varphi$  — частота и фаза несущей;  $U_m$  — амплитуда несущей на выходе модулятора.

Для автогенераторов в качестве модели используется описание формы генерируемых колебаний, так, в случае генератора гармонических колебаний имеем

$$U(t) = U_m \sin(\omega t + \varphi),$$

где  $U(t)$ ,  $U_m$ ,  $\omega$ ,  $\varphi$  — напряжение, амплитуда, частота и фаза колебаний.

Допущения, принимаемые при функциональном моделировании, существенно упрощают алгоритмы получения математических моделей систем (ММС) из математических моделей элементов (ММЭ). Математическая модель системы представляет собой совокупность ММЭ, входящих в систему, при отождествлении переменных, относящихся к соединяемым входам и выходам.

**Пример 4.1.** Электронный усилитель работает в малосигнальном режиме и состоит из двух каскадов и цепи обратной связи, имеющих передаточные функции  $K_1(p)$ ,  $K_2(p)$  и  $K_3(p)$  соответственно. Математическая модель может быть получена непосредственно по схеме усилителя, представленной на рис. 4.13:

$$U_{\text{вых}}(p) = K^* (U_{\text{вх}}(p) + K_3(p) U_{\text{вых}}(p)),$$

т. е. передаточная функция усилителя

$$h(p) = U_{\text{вых}}(p)/U_{\text{вх}}(p) = K^*/[1 - K^*K_3(p)], \quad (4.54)$$

где  $K^* = K_1(p)K_2(p)$ .

Для исследования линейных систем во временной области на основе модели типа (4.54) можно использовать два подхода. Первый подход связан с применением правил операционного исчисления и требует выполнения прямого преобразования Лапласа над входными сигналами и об-

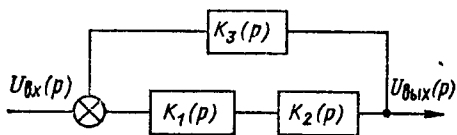


Рис. 4.13. Схема усилителя

ратного преобразования над выходными. Второй подход связан с представлением модели (4.54) в виде системы обыкновенных дифференциальных уравнений с ее последующим численным интегрированием.

**Пример 4.2.** Модель резонансного усилителя в виде системы обыкновенных дифференциальных уравнений получается из (4.53) с учетом того, что  $U_{\text{вых}}(p) = h(p)U_{\text{вх}}(p)$  и  $p = d/dt$ :

$$LC \frac{d^2 U_{\text{ВЫХ}}}{dt^2} + Cr \frac{dU_{\text{ВЫХ}}}{dt} + U_{\text{ВЫХ}} = S \left( L \frac{dU_{\text{ВХ}}}{dt} + rU_{\text{ВХ}} \right)$$

или, вводя обозначение  $I = CdU_{\text{ВЫХ}}/dt$ , образуем систему уравнений в нормальной форме Коши

$$I = CdU_{\text{ВЫХ}}/dt; LdI/dt + rI + U_{\text{ВЫХ}} = S(LdU_{\text{ВХ}}/dt + rU_{\text{ВХ}}),$$

где  $U_{\text{ВХ}}$ ,  $U_{\text{ВЫХ}}$  — заданная и рассчитанная функции времени соответственно.

При наличии в схеме нелинейных звеньев применяют второй из вышеназванных подходов.

При проектировании систем, в которых информация представлена в виде огибающей высокочастотных колебаний, возможны два способа введения переменных в модели. При первом *способе несущей* переменные изображают высокочастотные модулированные колебания. При анализе приходится имитировать поведение объекта в течение большого числа периодов несущей, что зачастую делает неприемлемо крупными затраты машинного времени. При втором *способе огибающей* переменные отображают огибающие высокочастотных колебаний. Отражение только низкочастотной огибающей существенно ускоряет вычисления, однако построение моделей может оказаться затруднительным.

## Математические модели логических схем цифровой РЭА.

На функционально-логическом уровне необходим ряд положений, упрощающих модели устройств и тем самым позволяющих анализировать более сложные объекты по сравнению с объектами, анализируемыми на схемотехническом уровне. Часть используемых положений аналогична положениям, принимаемым для моделирования аналоговой РЭА. Во-первых, это положение о представлении состояний объектов с помощью одноптипных фазовых переменных (обычно напряжений), называемых сигналами. Во-вторых, не учитывается влияние нагрузки на функционирование элементов-источников. В-третьих, принимается допущение об однонаправленности, т. е. о возможности передачи сигналов через элемент только в одном направлении — от входов к выходам. Дополнительно к этим положениям при моделировании цифровой РЭА принимается положение о дискретизации переменных, их значения могут принадлежать только заданному конечному множеству — алфавиту, например двоичному алфавиту  $\{0,1\}$ .

Моделирование цифровой РЭА возможно с различной степенью детализации. На логическом (*вентильном*) подуровне функционально-логического проектирования в качестве элементов аппаратуры рассматривают простые схемы типа вентиляей, на регистровом подуровне элементами могут быть как отдельные вентили, так и любые более сложные сочетания простых схем, например регистры, счетчики, дешифраторы, сумматоры, арифметико-логические устройства и т. п.

Рассмотрим математические модели элементов на логическом подуровне. Для одновыходных комбинационных элементов ММ представляет собой выражение (в общем случае алгоритм), позволяющее по значениям входных переменных (значениям входов) в заданный момент времени  $t$  вычислить значение выходной переменной (значение выхода) в момент времени  $t+t_3$ , где  $t_3$  — задержка сигнала в элементе. Такую модель элемента называют *асинхронной*. При  $t_3=0$  модель элемента называют *синхронной*. Модель многовыходного элемента должна включать в себя алгоритм вычисления задержек и значений всех выходных сигналов.

**Пример 4.3.** Модель элемента ЗИ—НЕ с фиксированной задержкой  $t_3$ , входами  $u_1, u_2, u_3$  и выходом

$$y(t+t_3) = \overline{u_1(t) \& u_2(t) \& u_3(t)}, \quad (4.55)$$

где  $\&$  — символ операции конъюнкции.

Для элементов последовательных схем (элементов с памятью) используют модели, в которых аргументами выходных переменных  $y_j$  могут быть как входные  $u_i$ , так и внутренние  $u_k$  переменные. Вектор внутренних переменных  $V$  отражает состояние элемента (состояние его памяти).

**Пример 4.4.** Модель  $J$ - $K$  триггера с входами  $J$  и  $K$  и состояниями  $V_0$  в предыдущем такте и  $V$  в последующем такте

$$V = V_0 \& \bar{J} \& \bar{K} + \bar{V}_0 \& J \& K + J \& \bar{K},$$

где « $\&$ » — символ операции дизъюнкции.

Объединение моделей элементов в общую математическую модель системы выполняется на основе вышперечисленных допущений отождествлением переменных на соединяемых входах и выходах элементов.

**Пример 4.5.** На рис. 4.14 представлен фрагмент схемы цифрового устройства. Соответствующая ему модель есть следующая система логических уравнений:

$$a = \overline{u_4 \& (u_1 + u_2)};$$

$$b = \overline{u_4 \& u_3};$$

$$y_1 = \overline{a \& y_2};$$

$$y_2 = \overline{b \& y_1}.$$

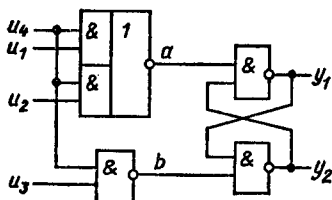


Рис. 4.14. Фрагмент схемы цифрового устройства

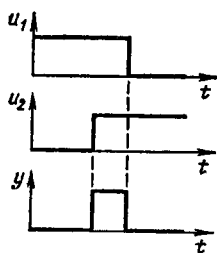


Рис. 4.15. Диаграммы статического риска сбоя

**Двузначные и многозначные модели.** Применение двузначного алфавита приводит к наиболее экономичным алгоритмам моделирования, однако двузначный алфавит ограничивает возможности анализа работоспособности схем. Поэтому чаще используют многозначное моделирование.

В трехзначном алфавите сигналы могут принимать зна-

чения из множества  $\{0, 1, X\}$ , где 0 и 1 соответствуют низкому и высокому уровням сигнала, а значение  $X$  интерпретируется как неопределенное (неизвестное). В моделях элементов с трехзначным (иначе — троичным) алфавитом, называемых трехзначными (троичными) моделями, сигналы могут принимать значение  $X$  во время переходных процессов, например во время своего переключения из единичного в нулевое состояние и обратно, а также в установившихся режимах при неопределенных значениях входных и внутренних переменных. Трехзначный алфавит используют для выявления ситуаций, в которых возможны сбои аппаратуры из-за рассогласования времен прохождения сигналов по различным цепям (из-за состязаний сигналов).

Дополнительные возможности обнаружения сбойных ситуаций предоставляет использование пятизначного алфавита  $\{0, 1, X, Д, Е\}$ , в котором с помощью значений  $Д$  и  $Е$  различают переходы сигналов из состояния 1 в состояние 0 и из состояния 0 в состояние 1 соответственно. Иногда алфавит расширяют в еще большей степени, добавляя в него символы, соответствующие таким ситуациям, как импульсные помехи, воздействие некоторых внешних факторов и т. п.

Результаты выполнения основных логических операций над аргументами  $a$  и  $b$  в дву-, трех- и пятизначном алфавитах представлены в табл. 4.4 (здесь аргументом для опе-

Таблица 4.4

$a$		0	1	$X$	$E$	$Д$
$b$		0 1 X E Д	0 1 X E Д	0 1 X E Д	0 1 X E Д	0 1 X E Д
Результат операции И	операция И	0 0 0 0 0	0 1 X E Д	0 X X X X	0 E X E X	0 Д X X Д
Результат операции ИЛИ	операция ИЛИ	0 1 X E Д	1 1 1 1 1	X 1 X X X	E 1 X E X	Д 1 X X Д
Результат операции НЕ	операция НЕ	1 0 X Д E	-----	-----	-----	-----

рации НЕ служит переменная  $b$ ). При этом к дву- и трехзначному моделированию относятся только те столбцы, в которых оба аргумента имеют значения соответственно из множества  $\{0, 1\}$  и  $\{0, 1, X\}$ .

**Синхронные и асинхронные модели.** Синхронные модели применяют для анализа установившихся состояний логических схем. Они представляют собой системы логических уравнений вида

$$\mathbf{V} = \mathbf{F}(\mathbf{V}, \mathbf{U}), \quad (4.56)$$

где  $\mathbf{V}$  — вектор, состоящий из выходных  $y_j$  и внутренних переменных;  $\mathbf{U}$  — вектор входных переменных.

Примером синхронной модели может служить вышеприведенная система уравнений для схемы, показанной на рис. 4.14.

Система уравнений (4.56) для последовательных схем имеет столько решений, сколько устойчивых состояний при заданном  $\mathbf{U}$  имеет моделируемая схема. Как правило, для анализа синхронных моделей используют методы, позволяющие получить то решение, которое соответствует исходному значению вектора  $\mathbf{V}$  и заданному входному набору  $\mathbf{U}$ . Получение такого решения называют *синхронным моделированием*.

Синхронное моделирование на основе двузначного алфавита позволяет проверить схему на отсутствие грубых ошибок типа неправильных соединений элементов. Дополнительную информацию о наличии в схеме рисков сбоя получают при применении трех- и пятизначного алфавита.

*Трехзначное синхронное моделирование* позволяет обнаружить статические риски сбоя. *Статический риск* сбоя выражается в появлении ложных сигналов на выходе схемы при неблагоприятном рассогласовании времен переключения входных сигналов.

**Пример 4.6.** На рис. 4.15 приведены возможные временные диаграммы для входов  $u_1$ ,  $u_2$  и выхода  $y$  схемы И. Из рисунка видно появление ложного сигнала на выходе схемы, что и является проявлением статического риска сбоя.

Для обнаружения статического риска сбоя требуется на каждом такте синхросигналов двукратное решение уравнений синхронной модели. Первое решение проводится при промежуточных значениях входных переменных: все изменяющиеся из состояний 1 или 0 входные переменные получают значение  $X$ , не изменяющиеся сохраняют свои исходные значения. Второе решение проводится при итоговых значениях входных переменных. Если  $y$  какой-либо переменной в схеме исходное, промежуточное и итоговое значения имеют последовательности 0— $X$ —0 или 1— $X$ —1, то данная переменная изображает ложный сигнал, т.е. указывает на наличие статического риска сбоя.

**Пример 4.7.** Для временной диаграммы (рис. 4.15) входная переменная  $u_1$  имеет последовательность исходного, промежуточного и итогового значений 1— $X$ —0; переменная  $u_2$  — последовательность

0—X—1 (значение X используется для отображения неопределенного состояния переменных при их переключении). В соответствии с правилами выполнения операции И в трехзначном алфавите (табл. 4.4) имеем для переменной  $y$  последовательность 0—X—0, что отождествляется со статическим риском сбоя.

*Пятизначное синхронное* моделирование позволяет дополнительно обнаруживать динамические риски сбоя. *Динамический риск сбоя* выражается в возможности многократного изменения некоторой переменной вместо правильного однократного изменения в течение одного такта синхронизации схемы.

Для выявления динамического риска сбоя выполняют двукратное решение системы логических уравнений при промежуточных и итоговых значениях входных переменных. Если у какой-либо изменяющейся переменной последовательность исходного, промежуточного и итогового значений отличается от возможных корректных последовательностей (корректными являются последовательности 0—E—1 и 1—D—0), то в схеме имеет место динамический риск сбоя.

**Пример 4.8.** На рис. 4.16, б приведена временная диаграмма, соответствующая такому сочетанию входных сигналов, при котором в схеме (рис. 4.16, а) проявляется динамический риск сбоя. Этой временной диаграмме соответствуют следующие последовательности значений переменных: для  $u_1$ —1—D—0, для  $u_2$ —0—E—1, для  $u_3$ —1—D—0. Для переменной  $y$  по правилам выполнения логических операций в пятизначном алфавите получаем 0—X—1, что отождествляется с динамическим риском сбоя.

Следует, однако, заметить, что синхронное моделирование указывает на возможность сбоев, которые в действительности происходят лишь при неблагоприятных рассогласованиях моментов переключения входных сигналов. Так, если (рис. 4.15) переключение переменной  $u_1$  произойдет раньше переключения переменной  $u_2$ , то в действительности ложного сигнала на выходе не будет, хотя синхронное моделирование указывает на

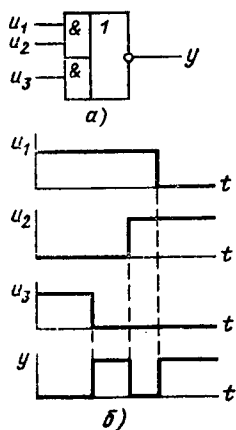


Рис. 4.16. Схема (а) и диаграммы (б) для иллюстрации динамического риска сбоя



риск сбоя. В этом отношении результаты синхронного моделирования аналогичны результатам анализа на наихудший случай — они могут привести к отрицательным заключениям о работоспособности вполне корректных схем.

*Асинхронные модели* схем составляют из асинхронных моделей элементов и применяют для анализа переходных процессов в цифровой РЭА. Время  $t$  в асинхронных моделях дискретизируется и измеряется в количестве тактов. Продолжительность такта достаточно малая — не должна превышать допустимую погрешность расчета временных параметров.

Асинхронную модель схемы можно представить в виде

$$V' = F(V, U), \quad (4.57)$$

где  $U$  — вектор входных переменных;  $V$  — вектор текущих значений внутренних и выходных переменных;  $V'$  — вектор будущих значений тех же переменных. Примером логического выражения, входящего в систему (4.57), является выражение (4.55). Алгоритм асинхронного моделирования заключается в подстановке в текущий момент времени  $t$  известных значений  $V$  и  $U$  в правую часть выражений (4.57), в вычислении новых значений переменных и их задержек, в увеличении модельного времени на длительность такта и корректировке вектора  $V$  с учетом результатов расчета задержек и значений вектора  $V'$ . Далее такты асинхронного моделирования повторяются до исчерпания заданного временного интервала анализа.

Асинхронные модели обычно используют с двузначным или трехзначным представлением переменных. *Трехзначное асинхронное моделирование* позволяет учесть разбросы задержек распространения сигналов в элементах. Пусть в момент времени  $t_1$  на вход элемента приходит сигнал, изменяющий состояние элемента с 0 на 1 с задержкой  $t_3$ , лежащей в интервале  $[t_{3\min}, t_{3\max}]$ . Тогда в асинхронной модели элемента значение выходной переменной

$$y = \begin{cases} 0 & \text{при } t_1 \leq t < t_1 + t_{3\min}; \\ X & \text{при } t_1 + t_{3\min} \leq t < t_1 + t_{3\max}; \\ 1 & \text{при } t \geq t_1 + t_{3\max}. \end{cases}$$

С помощью асинхронных моделей можно проанализировать прохождение сигналов во времени в цифровой РЭА с учетом реальных задержек в элементах при различных последовательностях входных сигналов. Однако асинхронное

моделирование обычно требует заметно бóльших затрат машинного времени по сравнению с синхронным.

**Математические модели функциональных схем цифровой РЭА на регистровом подуровне.** Первая особенность ММ на регистровом подуровне связана с разнообразием типов функциональных узлов, рассматриваемых в качестве элементарных при моделировании. Разнообразие типов элементов влечет за собой разнообразие их математических моделей. В ММ элементов могут использоваться различные типы данных, в частности величины булевы, целые, вещественные. Эти величины могут быть скалярными и векторными. Введение векторных переменных позволяет лаконично описывать многоразрядные счетчики, регистры, их входные и выходные сигналы. С помощью вещественных величин и операций над ними, которые присущи алгоритмическим языкам общего назначения, можно описать разнообразные алгоритмы, реализуемые в функциональных узлах различной сложности.

Математические модели на регистровом подуровне могут быть алгоритмического и схемного типов. Модели алгоритмического типа описывают алгоритмы функционирования устройств без привязки к их схемной реализации. Модели схемного типа отражают связи между переменными на входах и выходах функциональных узлов, составляющих анализируемую схему. Возможны смешанные модели, состоящие из алгоритмических и схемных описаний.

В большинстве случаев модели схемного типа представляют в виде конечных автоматов

$$KA = \{V_0, U, Y, V, \varphi, \Psi\}$$

и называют *автоматными моделями*. Здесь  $V_0$  — исходное состояние автомата;  $U, Y, V$  — векторы входных, выходных и внутренних переменных, принимающих значения в конечных множествах;  $\varphi$  — функция переходов;  $\Psi$  — функция выходов.

Функция переходов задает преобразование наборов входных и внутренних переменных на предыдущем такте в набор внутренних переменных на последующем такте:

$$V(t + t_3) = \varphi(V(t), U(t)),$$

где  $t_3$  — задержка, а функция выходов задает преобразование векторов  $V(t)$  и  $U(t)$  в вектор  $Y(t)$ .

Функции  $\varphi$  и  $\Psi$  могут выражаться с помощью формул, таблиц (матриц), описаний на входных языках прикладных

программ. Задержки  $t_z$  могут быть различными для разных путей прохождения сигналов от входов к выходам и в общем случае функционального узла с  $n$  входами и  $m$  выходами задаются в виде матрицы  $[t_{ij}]$ ,  $i = \overline{1, n}$ ,  $j = \overline{1, m}$ , где  $t_{ij}$  — задержка распространения сигнала от  $i$ -го входа к  $j$ -му выходу.

В отдельных случаях модель функционального узла может быть представлена в виде алгоритма, в котором действия выполняются над переменными  $U$  и  $Y$  вещественного типа. В таком виде удобно представлять сложные устройства, например арифметико-логические, выполняющие действия над числами с плавающей запятой.

Пример модели функционального узла

$$Y: = X1 * X2,$$

где  $Y$  — идентификатор выхода;  $X1$  и  $X2$  — идентификаторы входов;  $*$  — символ выполняемой в функциональном узле операции (алгоритма) над операндами  $X1$  и  $X2$ . Переменные  $Y$ ,  $X1$  и  $X2$  могут быть скалярами или векторами булевого или вещественного типа. Подобным образом описываются шифраторы, сумматоры, схемы контроля четности и т. п.

#### 4.8. МАТЕМАТИЧЕСКИЕ МОДЕЛИ ДЛЯ ЗАДАЧ КОНСТРУИРОВАНИЯ РЭА

При автоматизированном конструировании пользователю приходится принимать решения в условиях неопределенности, которые не имеют ни случайного, ни игрового характера. Эта неопределенность лежит в самом существе процесса принятия решения и происходит от неопределенности условий, в которых необходимо принимать решение.

Интенсивно развивающаяся в настоящее время теория расплывчатых (нечетких) множеств является попыткой создать более естественный и широкий математический аппарат для описания существующих неопределенностей множеств решений.

На практике, особенно при проектировании, приходится иметь дело с множествами, в которых нельзя указать резкую границу, отделяющую элементы, принадлежащие к данному множеству и не принадлежащие к нему. Следует отличать понятия случайности и расплывчатости. Случайность связана с неопределенностью, относящейся к принадлежности или непринадлежности элемента к множеству. Рас-

*плывчатое множество* — это такое множество, в котором могут иметься различные степени принадлежности элементов к данному множеству.

Пусть задано множество  $X = \{x_1, x_2, \dots, x_n\}$ . Тогда расплывчатое множество  $A \in X$  есть совокупность кортежей

$$\tilde{A} = \{ \langle \mu_A(x_i), x_i \rangle \}, x_i \in X, i = \overline{1, n},$$

где  $\mu_A(x_i)$  — степень принадлежности элемента  $x_i$  к  $A$ ;  $\mu_A$  — функция, отображающая  $x_i$  в пространство принадлежности  $M$ . Предполагается, что  $M$  — это интервал  $[0, 1]$ , причем 1 и 0 представляют высшую и низшую степени принадлежности.

Основное положение теории расплывчатых множеств состоит в том, что  $\tilde{A}$  несмотря на «размытость» границ задается точно путем сопоставления каждому элементу  $x \in X$  числа, лежащего между 0 и 1.

Например, пусть  $X = \{2, 4, 6 \dots\}$  — множество неотрицательных четных чисел. Тогда расплывчатое множество  $\tilde{A}$  можно, например, определить как набор кортежей вида

$$\tilde{A} = \{ \langle 0,6; 2 \rangle, \langle 0,2; 4 \rangle, \langle 0,7; 6 \rangle \dots \}.$$

Основой расплывчатого множества  $\tilde{A}$  называют множество элементов  $X$ , для которых  $\mu_A(x)$  положительно.

Носитель  $\tilde{A}$  есть множество элементов  $x \in X$ , для которых  $\mu_A(x)$  положительно. Точкой перехода  $A$  называют элемент  $x$ , для которого  $\mu_A(x) = 0,5$ . *Одноточечным расплывчатым множеством* (иногда используют термины «нечеткий», «размытый») называют множество, носитель которого состоит из единственной точки. Тогда если  $\tilde{A}$  — одноточечное расплывчатое множество, то его обозначают  $\tilde{A} = \mu/x$ , где  $\mu$  — степень принадлежности  $x$  к  $A$ . Тогда обыкновенное множество можно обозначить  $A = 1/x$ .

Расплывчатое множество рассматривают в виде объединения входящих в него одноточечных множеств. Если  $\tilde{A}$  состоит из конечного числа элементов, то

$$\tilde{A} = \mu_1/x_1 \cup \mu_2/x_2 \cup \dots \cup \mu_n/x_n = \bigcup_{i=1}^n \mu_i/x_i.$$

Очевидно, что тогда конечное множество  $X = \{x_1, x_2, \dots$

...,  $x_n$ } можно записать в виде

$$X = 1/x_1 \cup 1/x_2 \cup \dots \cup 1/x_n = \bigcup_{i=1}^n 1/x_i.$$

**Пример 4.9.** Пусть  $X = \{\text{кристалл, плата, панель}\}$ ;  $\tilde{A}$  — расплывчатое множество, определяемое признаком степень интеграции (СИ) элементов. Тогда, например, можно записать:

$A = \text{СИ большая/кристалл} \cup \text{СИ средняя/плата} \cup \text{СИ малая/панель}$ .

Степени принадлежности большая, средняя, малая также являются расплывчатыми подмножествами множества

$$W = 0 \cup 0, 1 \cup 0, 2 \cup, \dots, 0 \cup, 9 \cup 1.$$

А эти подмножества можно определить, например, так:

малая	. . . . .	0,5/0,1	1/0,2	0,8/0,3
средняя	. . . . .	0,6/0,4	1/0,5	0,9/0,6
большая	. . . . .	0,7/0,7	0,8/0,8	1/1

Такая запись означает, что малая степень принадлежности соответствует элементу 0,2, средняя степень принадлежности — элементу 0,5, а большая степень принадлежности — элементу 1.

В практических задачах конструирования функция принадлежности  $\mu_A$  определяется исходя из конкретных условий проектирования.

При проектировании конструкций пользователю удобнее иметь дело с моделями, которые легко образуются, если элементы конструкций принять за точки, а связи между ними — за линии. Такое представление объекта отличается высокой наглядностью, позволяет сосредоточить внимание на наиболее существенных связях, находить оптимальное решение задач проектирования. Использование аппарата теории графов для разработки алгоритмов конструкторского проектирования приводит нас к введению лишь некоторых определений, правил, теорем и положений из общей теории графов, которые будут представлять интерес в дальнейшем изложении.

Объект, состоящий из двух множеств (множества точек и множества линий), которые находятся между собой в некотором отношении, называют *графом*.

Множество точек графа обозначают  $X = \{x_1, x_2, \dots, x_n\}$ ,  $|X| = n$  и называют *множеством вершин*. Множество линий, соединяющих пары вершин  $(x_i, x_j) \in X$ , называют *множеством ребер* или дуг и обозначают  $U = \{u_1, u_2, \dots, u_m\}$ ,  $|U| = m$ . Тогда графом можно считать объект, который обозначается

как  $G = (X, U)$  и состоит из множества вершин  $X$  и множества ребер или дуг  $U$ , находящихся между собой в некотором отношении.

В общем случае множество  $U$  можно представить в виде  $U = \tilde{U} \cup \bar{U} \cup \dot{U}$ , где  $\tilde{U}$  — подмножество неориентированных линий, для которых не существенно направление соединения вершин. Такое подмножество называют *подмножеством ребер*. Каждое ребро  $u_i \in \tilde{U}$  определяется неупорядоченной парой вершин  $x_i, x_j$ , которые оно соединяет, и записывается  $u_k = (x_i, x_j)$  или  $u_k = (x_j, x_i)$ .  $\bar{U}$  — подмножество ориентированных линий, для которых существенно направление соединения вершин. Подмножество  $\bar{U}$  называют *подмножеством дуг*, причем каждая дуга  $u_i \in \bar{U}$  определяется упорядоченной парой (кортежем длины два) вершин  $x_i, x_j$ , которые  $u_k$  соединяет, и записывается  $u_k = \langle x_i, x_j \rangle$ . Подмножество  $\dot{U}$  — подмножество линий, каждая из которых выходит и входит в одну и ту же соответствующую этой линии вершину, и называется подмножеством петель. Каждая петля  $u_i \in \dot{U}$  может определяться упорядоченной или неупорядоченной парой, например, вида

$$u_i = (x_k, x_k) \text{ или } u_i = \langle x_k, x_k \rangle.$$

Граф  $G = (X, U)$  (рис. 4.17), у которого  $\bar{U}, \tilde{U}, \dot{U} \neq \emptyset$ , называют смешанным.

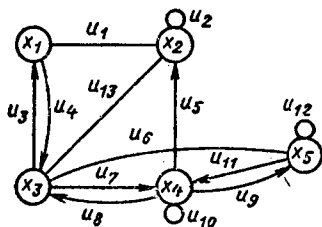


Рис. 4.17. Смешанный граф

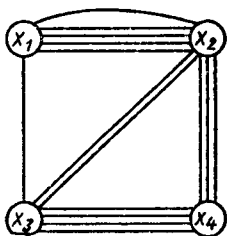


Рис. 4.18. Мультиграф

Здесь  $|X| = 5, |U| = 13, \bar{U} = U \cup \tilde{U} \cup \dot{U}$ ;

$$\bar{U} = \{u_3, u_4, u_5, u_7, u_8, u_9, u_{11}\},$$

$$\tilde{U} = \{u_1, u_6, u_{13}\}, \quad \dot{U} = \{u_2, u_{10}, u_{12}\}.$$

Граф  $G = (X, U)$ , у которого  $U = \bar{U}$ , а  $\bar{U}, \bar{U} = \emptyset$ , называют *ориентированным* или *орграфом*.

Граф  $G = (X, U)$ , у которого  $U = \bar{U}$ , называют *неориентированным* или *неорграфом*.

Заметим, что подмножество петель можно рассматривать как ориентированные, так и неориентированные подмножества ребер. Обычно графы с петлями оговаривают особо.

Рассмотрим неорграфы с петлями и без петель, которые будем называть просто графами. Граф  $G$ , у которого существует хотя бы одна пара вершин, соединяемых  $m$  ребрами  $u_i \in U$ , называют *мультиграфом*, а максимальное  $m$  — *мультичислом графа  $G$*  ( $m \in \{2, 3, \dots\}$ ). Ребра, соединяющие одну и ту же пару вершин, называются *кратными*. На рис. 4.18 показан пример мультиграфа, у которого  $m = 5$ .

Если ребро  $u_k \in U$  графа  $G = (X, U)$  соединяет вершины  $x_i, x_j \in X$ , т. е.  $u_k = (x_j, x_i)$ , то говорят, что ребро  $u_k$  инцидентно вершинам  $x_i$  и  $x_j$ . Аналогично и обратное: вершины  $x_i$  и  $x_j$  инцидентны ребру  $u_k$ .

Любые две вершины  $x_i, x_j \in X$  графа  $G$  называют *смежными*, если существует соединяющее эти вершины ребро  $u_k \in U$ , т. е.  $u_k = (x_i, x_j)$ . Если два ребра  $u_k, u_i \in U$  инцидентны одной и той же вершине, то их называют смежными. Следовательно, отношения инцидентности и смежности могут иметь место как на множестве  $X$ , так и на множестве  $U$ .

Граф  $G = (X, U)$  можно задать в различных формах. Основными из них являются геометрическая и матричная.

Основой геометрической формы задания графа является рисунок. Изображение графа в виде рисунка обладает наглядностью, раскрывает содержательный смысл представляемого объекта.

Граф называют *помеченным*, если его вершины отличаются одна от другой различными метками, например  $x_1, x_2, \dots, x_n$ . Для конструирования необходима нумерация элементов, поэтому в дальнейшем будем рассматривать помеченные графы.

Большинство задач автоматизации конструирования удобно решать при использовании матричной формы задания графа. Квадратную таблицу  $R = \|r_{ij}\|_{n \times n}$  называют *матрицей смежности*, если ее элементы образуются по правилу

$$r_{ij} = \begin{cases} 1, & \text{если вершина } x_i \text{ соединена с } x_j \text{ ребром, т. е. } x_i \\ & \text{смежна } x_j; \\ 0 & \text{в противном случае.} \end{cases}$$

Для мультиграфа

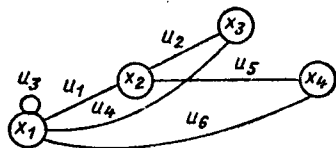
$$r_{ij} = \begin{cases} m, & \text{если вершина } x_i \text{ соединена с } x_j \text{ } m \text{ ребрами;} \\ 0 & \text{в противном случае.} \end{cases}$$

Очевидно, что для рассматриваемых графов  $r_{ij} = r_{ji}$  и для задания графа достаточно использовать половину матрицы  $R$ . Граф  $G$  (рис. 4.19) имеет следующую матрицу смежности:

$$R = \begin{array}{c} \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \left\| \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{array} \right\| \end{array}$$

Строки и столбцы матрицы  $R$  соответствуют вершинам графа  $G$ . На пересечении  $x_i$  строки и  $x_j$  столбца находится элемент  $r_{ij}$ , соответствующий числу ребер, соединяющих

Рис. 4.19. Пример графа



вершины  $x_i$  и  $x_j$ . Заметим, что строки и столбцы матрицы  $R$  также можно нумеровать числами натурального ряда, соответствующими индексам помеченных вершин графа. Петли в графе изображаются элементами  $r_{ii} = r_{jj}$ , расположенными по главной диагонали матрицы  $R$ . Преимущество использования матрицы смежности — это простота и формальность преобразований над графами. Основным недостатком применения матрицы смежности заключается в том, что они занимают большой объем памяти в ЭВМ даже тогда, когда граф содержит только  $O(X)$  ребер. Если  $|U| \leq |X^2|$ , то этот недостаток устраняется, если представить матрицу в виде списка. Такое представление графа требует памяти ЭВМ порядка  $O(|X| + |U|)$ . Прямоугольную таблицу вида  $I = \|i_{ki}\|_{n \times m}$  называют матрицей инцидентности, если ее элементы образуются по правилу



$i_{ki} = \begin{cases} 1, & \text{если вершина } x_k \text{ инцидентна ребру } u_i; \\ 0 & \text{в противном случае.} \end{cases}$

Матрица инцидентности для графа  $G$  (рис. 4.19) имеет вид

$$I = \begin{matrix} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{matrix} & \left\| \begin{array}{cccccc} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array} \right\| \end{matrix}.$$

Строки матрицы  $I$  соответствуют вершинам графа, столбцы — ребрам, а элемент  $i_{ki}$  указывает на инцидентность вершины  $x_k$  и ребра  $u_i$ . В каждом столбце матрицы  $I$  расположено по две единицы, так как каждое ребро соединяет ровно две вершины. При наличии в графе петель соответствующие им столбцы в матрице  $I$  будут иметь по одной единице, так как петля соединяет только одну вершину графа.

Матрицы  $R$  и  $I$  задают однозначно информацию о графе.

Граф  $G$  называется *конечным*, если множество его вершин и ребер — конечные множества. Далее будем рассматривать конечные графы, так как объекты проектирования, представляемые графами, состоят из конечного числа элементов.

Граф  $G$ , у которого  $X \neq \emptyset$ , а  $U = \emptyset$ , называют *нуль-графом*, а его вершины — *изолированными*. Обозначают его  $G_0$ .

Граф  $G = (X, U) |X| = n$  называют *полным*, если между любой парой вершин  $x_i, y_j \in X$  имеется ребро  $u_k \in U$ . Обозначают его  $K_n$ . Число ребер, инцидентных вершине  $x_i \in X$  графа  $G$ , называется *локальной степенью* этой вершины и обозначается  $\rho(x_i)$ . Тогда для графа  $G = (X, U)$ ,  $|X| = n$ ,  $|U| = t$ , число ребер

$$t = \sum_{i=1}^n \rho(x_i)/2.$$

Если в графе имеются петли, то каждую из них считают дважды. Так как для полного графа на  $n$  вершин  $\rho(x_1) = \rho(x_2) = \dots = \rho(x_n)$ , то  $t = n(n-1)/2$ .

*Подграфом* графа  $G$  называют граф, у которого все вершины и ребра принадлежат  $G$ , т. е.  $G' = (X', U')$  — подграф графа  $G = (X, U)$ , если  $X' \subseteq X$ ,  $U' \subseteq U$  и ребра  $U'$  соединяют только вершины  $X'$ .

Суграфом  $G' = (X', U')$  графа  $G = (X, U)$  называют граф, для которого  $X' = X, U' \subseteq U$ .

Граф  $\bar{G}$  называют дополнением графа  $G$  до полного, если он состоит из всех ребер полного графа  $K_n$ , не принадлежащих  $G$ , т. е.

$$\bar{G} = K_n \setminus G.$$

На рис. 4.20, а показан суграф графа, изображенного на рис. 4.19. На рис. 4.20, б приведено дополнение его до полного.

Маршрутом в графе  $G = (X, U)$  называют некоторую конечную последовательность ребер вида  $S = (x_0, x_1), (x_1,$

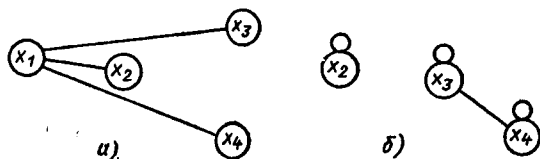


Рис. 4.20. Суграф графа  $G$  (а) и дополнение графа  $G$  до полного (б)

$x_2), \dots, (x_{i-1}, x_i)$ , где  $x_0, x_i$  — начальная и конечная вершины соответственно. В конечном графе  $G$  можно выделить только конечное число маршрутов. Число ребер в маршруте  $S$  называют его длиной.

Маршрут, в котором нет повторяющихся ребер, называют цепью. Замкнутую цепь, в которой  $x_0 = x_i$ , называют циклом. Соответственно цепи и циклы называют простыми, если они не содержат повторяющихся вершин, кроме, разумеется, первой и последней в случае цикла. На рис. 4.21, а изображен неэйлеров граф. Здесь

$S_1 = (x_1 x_2), (x_2 x_3), (x_3 x_5), (x_5 x_2), (x_2, x_1), (x_1, x_4)$  — маршрут,

$S_2 = (x_1 x_2), (x_2 x_3), (x_3 x_5), (x_5 x_2)$  — цепь,

$S_3 = (x_5 x_2), (x_2 x_3), (x_3 x_6), (x_6 x_5)$  — цикл,

$S_4 = (x_1 x_2), (x_2 x_3), (x_3 x_5), (x_5 x_6)$  — простая цепь,

$S_5 = (x_2 x_3), (x_3 x_5), (x_5 x_2)$  — простой цикл.

Матрица циклов графа  $G$  (рис. 4.21, а) имеет вид

$$M_C = \begin{matrix} & & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ \text{I} & \parallel & 1 & 1 & 0 & 1 & 1 & 0 \\ \text{II} & \parallel & 0 & 1 & 1 & 0 & 1 & 0 \\ \text{III} & \parallel & 0 & 1 & 1 & 0 & 1 & 1 \end{matrix} \parallel.$$

Две произвольные вершины  $x_i, x_j \in X$  графа называют *связными*, если существует маршрут  $S$ , в котором концевыми будут вершины  $x_i, x_j$ . Граф  $G$  называют *связным*, если любые две его вершины связаны. В противном случае  $G$  *несвязан*, а каждый из соответствующих его подграфов

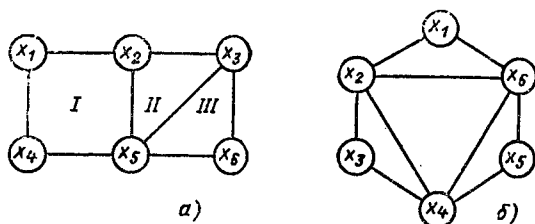


Рис. 4.21. Незэйлеров (а) и эйлеров (б) графы

$G_1, G_2, \dots, G_i$  называют *компонентой связности*. При решении задач конструирования важное значение имеют графы специального вида — эйлеровы и гамильтоновы.

Связный граф  $G = (X, U)$  называется *эйлеровым*, если существует замкнутая цепь, проходящая через каждое его ребро только один раз.

В эйлеровом графе все локальные степени четны. Например, граф  $G$  (рис. 4.21, а) не является эйлеровым, так как степени  $x_2$  и  $x_3$  нечетны. Говорят, что граф имеет эйлеров цикл, если связан и все его локальные степени четны. Эйлеров цикл (ЭЦ) обозначается через  $C_e$ . Для графа  $G$  (рис. 4.21, б)

$$C_e = (x_1 x_2) (x_2 x_3) (x_3 x_4) (x_4 x_2) (x_2 x_6) (x_6 x_4) (x_4 x_5) (x_5 x_6) (x_6 x_1).$$

Цикл, проходящий по всем вершинам графа  $G$  один раз, называют *гамильтоновым*, а  $G$  называют *гамильтоновым графом*. Например, граф  $G$  (рис. 4.20, а) не имеет гамильтонова цикла (ГЦ), а граф  $G$  (рис. 4.21, б) имеет  $G_{ГЦ} = (x_1, x_2, x_3, x_4, x_5, x_6)$ . В отличие от ЭЦ для ГЦ неизвестен общий критерий существования. В основном известны только теоремы, дающие достаточные условия существования ГЦ.

**Теорема.** Если в графе  $G$  с  $n$  вершинами для любой пары вершин  $x_i, x_j$  имеем

$$\rho(x_i) + \rho(x_j) \geq n,$$

то  $G$  имеет ГЦ.

Из теоремы следует результат Дирака, что граф имеет ГЦ, если для каждой его вершины  $\rho(x_i) \geq n/2$ .

Очевидно, что полный граф всегда содержит гамильтонов цикл. Связный граф без циклов называют *деревом* и обозначают  $T=(X, U)$ ,  $|X|=n$ . Любое дерево  $T$  имеет  $n-1$  ребро. Начальную вершину называют *корнем*, из которого выходят ребра, называемые *ветвями* дерева. Очевидно, что в дереве любые две вершины  $x_i, x_j$  дерева связаны единственной цепью. В любом связном графе  $G$  можно выделить произвольное дерево  $T$ . Для задач конструирования РЭА наибольший интерес представляют деревья, у которых число вершин равно числу вершин графа, из которого выделено это дерево. Такие деревья называют *покрывающими*. Для одного и того же связного графа можно выделить некоторое множество покрывающих деревьев.

**Теорема.** Число покрывающих деревьев  $t$  графа  $K_n$ , содержащего  $n$  вершин, составляет

$$t = n^{n-2}.$$

Множество деревьев графа называют *лесом*. Задачи выделения эйлеровых и гамильтоновых циклов и покрывающих деревьев связаны с задачами о лабиринте, коммивояжере и с построением путей минимальной стоимости.

Задача о лабиринте в терминах теории графов формулируется как задача отыскания в связном графе  $G=(X, U)$  такого маршрута, который начинается в заданной вершине  $x_i \in X$  и приводит в искомую  $x_j \in X$ , причем маршрут должен содержать наименьшее число ребер.

Пусть необходимо проложить сеть проводов, связывающих  $n$  терминалов вычислительной аппаратуры, причем так, чтобы из одного терминала можно было связаться с любым другим. Если из экономических соображений требуется, чтобы количество затраченного провода было минимально, то граф, вершины которого соответствуют терминалам, а ребра — соединяющим их проводам, должен быть деревом. Задача состоит в определении одного из  $n^{n-2}$  возможных деревьев, соединяющих терминалы.

Эту задачу можно усилить. Пусть  $G=(X, U)$  — связный граф и каждому ребру  $u_i \in U$  ставится в соответствие некоторое неотрицательное число  $v(u)$ , называемое его мерой. Необходимо найти покрывающее дерево  $T$ , у которого сумма мер, взятая по всем ребрам,

$$\sum_{i=1}^m v(u_i) = \min.$$

Приведем алгоритм решения данной задачи.

1. В связном графе  $G = (X, U)$ ,  $|X| = n$  определяется ребро с наименьшей мерой  $u_1$ .

2. Строят по индукции последовательность ребер  $u_2, u_3, \dots, u_{n-1}$ , выбирая на каждом шаге ребро, не совпадающее с предыдущим, с наименьшей мерой и не образующее циклов с предыдущими ребрами  $u_i$ .

3. Получается подграф  $T$  графа  $G$  с ребрами  $u_1, \dots, u_{n-1}$ , который и является искомым деревом с наименьшей суммой мер. Расстоянием  $d_{i,j}$  между вершинами  $x_i, x_j \in X$  графа  $G = (X, U)$  называют длину кратчайшей цепи, соединяющей эти вершины. Под длиной цепи понимают число входящих в нее ребер.

Функцию расстояний для графа  $G$  удобно задавать матрицей расстояний  $D = \|d_{i,j}\|_{n \times n}$  или ее списком.

Элемент матрицы  $D$  определяется следующим образом:

$$d_{i,j} = \begin{cases} 0, & \text{если } x_i = x_j; \\ d_{i,j}, & \text{если } x_i \neq x_j. \end{cases}$$

Для графа  $G$  (рис. 4.21, б) матрица расстояний имеет вид

$$D = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \left\| \begin{array}{cccccc} 0 & 1 & 2 & 2 & 2 & 1 \\ 1 & 0 & 1 & 1 & 2 & 1 \\ 2 & 1 & 0 & 1 & 2 & 2 \\ 2 & 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 1 & 1 & 0 \end{array} \right\| \end{matrix}.$$

Для задач конструирования представляет интерес нахождение функции расстояний для графов  $G_r$  частного вида, называемых координатной решеткой. В графе  $G_r = (X_r, U_r)$  множество вершин  $X_r$  соответствует узлам решетки, а множество  $U_r$  ребер — горизонтальным и вертикальным отрезкам, соединяющим узлы решетки. Пример графа  $G_r$  — координатной решетки — показан на рис. 4.22.

Расстояние между двумя смежными вершинами в решетке  $G_r$ , называемое шагом решетки, принимают равным единице. Между двумя произвольными вершинами в решетке  $G_r$  расстояние

$$d_{i,j} = |s_i - s_j| + |t_i - t_j|,$$

где  $s_i, s_j$  и  $t_i, t_j$  — координаты вершин  $x_i, x_j \in G_r$ .

Обычно задаются размеры решетки  $p \times q$ , где  $p$  — число узлов решетки по оси  $s$ , а  $q$  — по оси  $t$ .

Например, для графа  $G_r$  (рис. 4.22) расстояние

$$d_{(6,11)} = |s_6 - s_{11}| + |t_6 - t_{11}| = |4 - 0| + |1 - 2| = 5.$$

Если произвольный граф  $G$  отображается в  $G_r$ , так, что любые вершины  $G$  размещаются в узлах решетки, то расстояние между вершинами  $G$  определяется как расстояние между соответствующими узлами решетки  $G_r$ .

Любой граф  $G$  может быть отображен в решетку  $G_r$ . Для подсчета суммарной длины  $L(G)$  ребер графа  $G$ , ото-

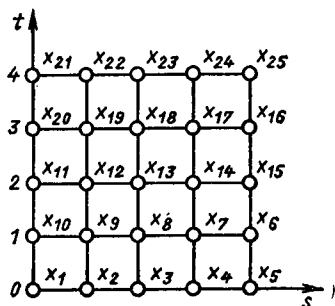


Рис. 4.22. Граф  $G_r$  — координатная решетка

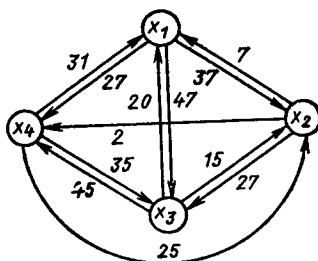


Рис. 4.23. Граф  $G$

браженного в решетку  $G_r$ , введем понятие матрицы геометрии  $D_\gamma$ , представляющей собой часть матрицы расстояний  $D$ , в которой исключены элементы  $d_{i,j}$ , если вершины  $x_i, x_j \in X$  не смежны в графе  $G$ .

Для построения матрицы геометрии  $D_\gamma$  графа  $G$  необходимо каждый элемент матрицы  $D$  умножить на соответствующий элемент матрицы смежности  $R$ , т.е.  $D_\gamma = \|r_{i,j}d_{i,j}\|_{n \times n}$ . Сумма элементов матрицы  $D_\gamma$  определяет удвоенную суммарную длину  $L(G)$  ребер графа  $G$  при данном отображении его в решетку  $G_r$ .

С задачей определения гамильтоновых циклов и кратчайших расстояний тесно связана задача о коммивояжере, суть которой в следующем: имеется  $N$  городов (вершин графа) и заданы расстояния между городами. Коммивояжер находится в городе  $x_1$ . Ему необходимо посетить только по одному разу все остальные  $N-1$  городов и вернуться в  $x_1$ , чтобы общее пройденное расстояние было минимальным.

Рассмотрим описание алгоритма. Сначала строят все кратчайшие пути, соединяющие  $x_1$  с любой вершиной  $l$  ( $l=2, 3, \dots, N$ ). Длина любого из этих путей равна расстоянию  $d_{1,l}$  от  $x_1$  до  $x_l$ . Далее определяют путь минимальной длины, соединяющий  $x_1$  с  $x_l$  и проходящий через одну промежуточную вершину  $x_m$ . Зная длину минимального пути из  $x_1$  в  $x_l$  с одной промежуточной вершиной, можно определить длину пути минимальной длины между  $x_1$  и  $x_l$ , проходящего через две промежуточные вершины, три и т. д.

При увеличении числа промежуточных вершин обязательно будет построен путь минимальной длины из  $x_1$  в любую вершину, проходящий через все остальные вершины. Тогда можно определить минимальный путь с учетом возвращения в  $x_l$ .

**Пример 4.10.** Пусть для графа  $G$  (рис. 4.23) задана матрица расстояний

$$D = \begin{matrix} & & 1 & 2 & 3 & 4 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \left\| \begin{matrix} 0 & 37 & 47 & 27 \\ 7 & 0 & 27 & 2 \\ 20 & 15 & 0 & 45 \\ 31 & 25 & 35 & 0 \end{matrix} \right\| \end{matrix}.$$

Согласно алгоритму определяют кратчайшие пути, ведущие из  $x_1$  в  $x_l$  ( $l=2, 3, 4$ ). Получают  $d_{1,2}=37$ ,  $d_{1,3}=47$ ,  $d_{1,4}=27$ . Далее находят пути минимальной длины, соединяющие  $x_1$  с  $x_2$  и проходящие через некоторую промежуточную вершину. Пусть  $l=2$ , тогда длина пути из  $x_1$  в  $x_2$

$$d_{1,2}^3 = d_{1,3} + d_{3,2} = 47 + 15 = 62;$$

$$d_{1,2}^4 = d_{1,4} + d_{4,2} = 27 + 25 = 52.$$

Аналогичные вычисления выполняют для  $x_3$  и  $x_4$ . Результаты сведены в таблицу:

	$l=2$	$l=3$	$l=4$
$x_1$	$d_{1,2}=37$	$d_{1,3}=47$	$d_{1,4}=27$
$x_2$	$d_{1,2}^3 = 62$	$d_{1,3}^2 = 64$	$d_{1,4}^2 = 39$
	$d_{1,2}^4 = 52$	$d_{1,3}^4 = 62$	$d_{1,4}^3 = 92$
$x_3$	$d_{1,2}^{3,4} = 120,$	$d_{1,3}^{2,4} = 74,$	
	$d_{1,2}^{4,3} = 77$	$d_{1,3}^{4,2} = 83$	

$$d_{1,3}^2 = d_{1,2} + d_{2,3} = 37 + 27 = 64;$$

$$d_{1,3}^4 = d_{1,4} + d_{4,3} = 27 + 35 = 62;$$

$$d_{1,4}^2 = d_{1,2} + d_{2,4} = 37 + 2 = 39;$$

$$d_{1,4}^3 = d_{1,3} + d_{3,4} = 47 + 45 = 92.$$

Длина пути:

из  $x_1$  в  $x_2$ , проходящие через вершины  $x_3$  и  $x_4$ ,

$$d_{1,2}^{3,4} = d_{1,3} + d_{3,4} + d_{4,2} = d_{1,4}^3 + d_{4,2} = 92 + 25 = 117,$$

$$d_{1,2}^{4,3} = d_{1,4} + d_{4,3} + d_{3,2} = d_{1,3}^4 + d_{3,2} = 62 + 15 = 77;$$

из  $x_1$  в  $x_3$ , проходящие через  $x_2$  и  $x_4$ ,

$$d_{1,3}^{2,4} = d_{1,2} + d_{2,4} + d_{4,3} = d_{1,4}^2 + d_{4,3} = 39 + 35 = 74,$$

$$d_{1,3}^{4,2} = d_{1,4} + d_{4,2} + d_{2,3} = d_{1,2}^4 + d_{2,3} = 52 + 27 = 79;$$

из  $x_1$  в  $x_4$ , проходящие через  $x_2$  и  $x_3$ ,

$$d_{1,4}^{2,3} = d_{1,3}^2 + d_{3,4} = 64 + 45 = 109,$$

$$d_{1,4}^{3,2} = d_{1,2}^3 + d_{2,4} = 62 + 2 = 64.$$

Теперь определяют минимальную длину пути, включая возвращение в  $x_1$ . Зная расстояние  $d_{i,1}$  от  $x_2$  до  $x_1$ , можно получить общую длину пути при условии, что из  $x_i$  попадаем в  $x_1$ .

Найдем все пути из  $x_i$  в  $x_1$ :

$$d_{1,2,1}^{4,3} = d_{1,2}^{4,3} + d_{2,1} = 77 + 7 = 84,$$

$$d_{1,3,1}^{2,4} = d_{1,3}^{2,4} + d_{3,1} = 74 + 20 = 94,$$

$$d_{1,4,1}^{3,2} = d_{1,4}^{3,2} + d_{4,1} = 64 + 31 = 95.$$

Минимальная длина пути из вершины  $x_1$  в  $x_1$  равна 84. Порядок обхода следующий:  $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$ , это вытекает из величины  $d_{1,2,1}^{4,3}$ .

Заметим, что оптимизация в алгоритмах конструирования упрощается при использовании таких числовых инвариантных характеристик графа, как цикломатическое число или число внутренней и внешней устойчивости и др.

Наименьшее число ребер, которые необходимо удалить из графа  $G$ , чтобы он стал ациклическим (деревом), называют *цикломатическим числом* графа. Для графа  $G = (X, U)$ ,  $|X| = n$ ,  $|U| = m$ , цикломатическое число

$$\psi(G) = m - n + k,$$



где  $m$  — число ребер;  $n$  — число вершин;  $k$  — число компонент связности графа.

Для графа, состоящего из одной компоненты связности, цикломатическое число  $\gamma(G) = m - n + 1$  показывает только число ребер, которые необходимо удалить из графа.

*Раскраской* вершины графа называют разбиение множества вершин  $X$  на  $l$  непересекающихся классов (подмножеств)

$$\begin{aligned} X_1, X_2, \dots, X_l; \quad X &= \bigcup_{i=1}^l X_i; \quad X_i \cap X_j = \emptyset; \quad i, j \in I = \\ &= \{1, 2, \dots, l\}; \quad i \neq j, \end{aligned}$$

таких, что внутри каждого подмножества не должно содержаться смежных вершин. Если каждому подмножеству  $X_i$  поставить в соответствие определенный цвет, то вершины внутри этого подмножества можно окрасить в один цвет, вершины другого подмножества  $X_j$  в другой цвет и т. д. до раскраски всех подмножеств. В этом случае граф называют  $l$ -раскрашиваемым. Наименьшее число подмножеств, на которое разбивают граф при раскраске, называют *хроматическим числом*  $\kappa(G)$ .

Полный граф  $K_n$  можно раскрасить только  $n$  цветами, следовательно,  $\kappa(K_n) = n$ .

Рассмотрим число *внутренней устойчивости*. Если две любые вершины подмножества  $X'$  не смежны, то это подмножество называют внутренне устойчивым. Подмножество  $\Psi \subseteq X$  графа  $G = (X, U)$  называют максимальным внутренне устойчивым подмножеством (МВУП) или независимым, если добавление к нему любой вершины  $x_i \in X$  делает его не внутренне устойчивым. Подмножество  $\Psi_i$  будет *независимым*, если

$$\forall x_i \in \Psi_i \quad (\Gamma x_i \cap \Psi_i = \emptyset).$$

Независимые подмножества (НП) различаются по числу входящих в них элементов. В произвольном графе  $G$  можно выделить семейство всех НП вида  $\Psi = \{\Psi_1, \Psi_2, \dots, \Psi_n\}$ . Независимые подмножества, содержащие наибольшее число элементов, называют *предельными*. Тогда число внутренней устойчивости  $\eta(G)$  определяется мощностью предельного НП:

$$\eta(G) = |\max_{\Psi_i \in \Psi} \Psi_i|;$$

Число внутренней устойчивости  $\eta(\mathbf{G})$  можно связать с хроматическим числом  $\kappa(\mathbf{G})$  следующим образом:

$$\eta(\mathbf{G}) \kappa(\mathbf{G}) \geq n.$$

Многие задачи контроля схем сводят к различным тождественным преобразованиям заданных графов этих устройств. Тождественные преобразования графов, сводимые только к переобозначению вершин и ребер, приводят к получению изоморфных графов.

Два графа  $\mathbf{G} = (\mathbf{X}, \mathbf{U})$  и  $\mathbf{G}' = (\mathbf{X}', \mathbf{U}')$  называют *изоморфными*, если можно установить взаимно однозначное соответствие  $\mathbf{X} \leftrightarrow \mathbf{X}'$ ,  $\mathbf{U} \leftrightarrow \mathbf{U}'$  такое, что если  $(x_i, x_j) \in \mathbf{X} \leftrightarrow (x'_i, x'_j) \in \mathbf{X}'$ , то ребро  $u = (x_i, x_j) \in \mathbf{U} \leftrightarrow u' = (x'_i, x'_j) \in \mathbf{U}'$ .

Изоморфные графы могут быть получены один из другого путем перенумерации их вершин. Очевидно, что изоморфизм есть отношение эквивалентности на графах. Если изоморфные преобразования проводятся с графом, заданным матрицей смежности, то они сводятся к перестановке местами соответствующих строк и столбцов. Известно, что в общем случае для определения изоморфизма графов необходимо сделать  $n!$  сравнений или перестановок строк и столбцов матрицы, что для графов с  $n > 30$  не под силу даже современной ЭВМ. Поэтому необходимо применить тот или иной эвристический алгоритм поиска по дереву решений.

При конструировании схем к их топологическому чертежу предъявляется требование получения, либо плоского изображения схем, либо плоского изображения частей схем. В этой связи возникает задача определения планарности графа.

Граф  $\mathbf{G} = (\mathbf{X}, \mathbf{U})$  называют *плоским*, если его множество ребер расположено на плоскости таким образом, что ребра имеют общие точки лишь в вершинах. Граф, изоморфный плоскому и расположенный на плоскости с пересечением ребер, называют *планарным*.

Область плоскости, ограниченная ребрами плоского графа, внутри которой нет вершин ребер, называют *гранью*. Ребра грани образуют простой цикл. Заметим, что плоский граф имеет всегда одну бесконечную грань, не ограниченную ребрами. Существует формула Эйлера, позволяющая установить связь между числом граней, числом вершин и числом ребер плоского графа:

$$n - m + f = 2,$$

где  $f$  — число граней плоского графа.

Определять планарность графа можно с помощью различных критериев. Рассмотрим основной из них.

Пусть задан граф  $G = (X, U)$ . Подразбиением ребра  $u_k = (x_i, x_j)$  называют замену его двумя ребрами  $u_{p1} = (x_i, x_p)$  и  $u_{p2} = (x_p, x_j)$  с введением новой вершины  $x_p$ . Два графа называют *гомеоморфными*, если они обладают изоморфными подразделениями.

**Теорема Понтрягина — Куратовского.** Граф планарен тогда и только тогда, когда не содержит подграфов, гомеоморфных полному графу  $K_5$  (рис. 4.24, а) и полному двудольному графу  $K_{3,3}$  (рис. 4.24, б).

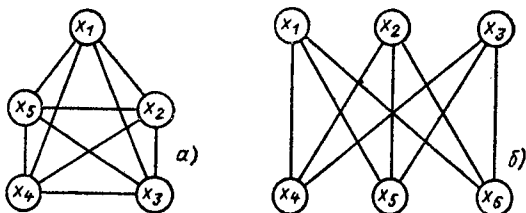


Рис. 4.24. Полный граф  $K_5$  (а) и полный двудольный граф  $K_{3,3}$  (б)

Очевидно, что граф планарен тогда и только тогда, когда планарны все его связные компоненты. Поэтому для определения планарности рассматривают связные графы. Распространенная методика определения планарности заключается в нахождении в графе  $G$  максимального цикла  $C$  (лучше всего гамильтонова) и размещении его на плоскости в виде замкнутой самопересекающейся кривой. Далее в оставшейся части определяют пересекающиеся по ребрам пути и предпринимают попытки разместить каждый из этих путей либо полностью внутри  $C$ , либо полностью вне  $C$ . Если таким образом размещается весь граф, то он планарен, в противном случае не планарен. Основная проблема — иметь возможность генерирования множества путей, выбора областей для планарного размещения и перестановки путей. Сложность алгоритма —  $O(n)$ .

Число попарных пересечений ребер (иногда называется числом скрещиваний) на плоскости обозначается  $P(G)$  и для полного графа определяется выражением

$$P(K_n) \leq \begin{cases} 1/64 (n-1)^2 (n-3)^2, & \text{если } n \text{ нечетно;} \\ 1/64 n (n-2)^2 (n-4), & \text{если } n \text{ четно.} \end{cases}$$

Зная, что  $P(K_5) = 1$ , можно определить минимальное число пересечений для полного графа с любым числом вершин.

Часто при описании элементов системы связи необходимо учитывать направление.

Орграф  $G = (X, \vec{U})$  будем обозначать  $D = (X, U)$ . В графе  $D$  маршрутом считается чередующаяся последовательность вершин и дуг:  $(x_0, u_1, x_1, \dots, u_n, x_n)$ , в которой каждая дуга  $u_i = (x_{i-1}, x_i)$ . *Путь* — это маршрут, в котором все вершины различны. *Контур* — это простой замкнутый маршрут, у которого все вершины различны за исключением первой и последней. Если существует путь из  $x_i$  в  $x_j$ , то говорят, что  $x_j$  достижима из  $x_i$ .

Граф  $D$  называют сильно связным, если любые две его вершины взаимно достижимы.

Матрицей смежности графа  $D$  называют матрицу  $R(D) = \|r_{ij}\|_{n \times n}$ , причем

$$r_{i,j} = \begin{cases} 1, & \text{если } \langle x_i, x_j \rangle \text{ — дуга графа } D; \\ 0 & \text{в противном случае.} \end{cases}$$

Так как  $r_{i,j} \neq r_{ji}$ , то матрица несимметрична относительно главной диагонали.

Заметим, что в графе  $D$  дуга  $u_i$  есть кортеж, т. е. упорядоченное множество из двух вершин. Дуга  $u_i = \langle x_i, x_j \rangle$  считается положительно инцидентной ее конечной вершине  $x_j$ . Число дуг, положительно инцидентных вершине  $x_j$ , называют полустепенью захода и обозначают  $\rho^+(x_j)$ . Отрицательную степень  $x_j$  определяют аналогично, называют полустепенью исхода и обозначают  $\rho^-(x_j)$ .

Очевидно, что  $\rho(x_j) = \rho^+(x_j) + \rho^-(x_j)$ . Так как дуга положительно инцидентна только одной вершине  $x_j$  и отрицательно инцидентна той же самой вершине  $x_j$ , то

$$\sum_{x_i \in X} \rho^+(x_i) = \sum_{x_j \in X} \rho^-(x_j) = |U|.$$

Элементы матрицы инцидентности  $I(D)$  графа  $D$  принимают значения 0, +1, -1. Элемент равен нулю, если вершина не инцидентна дуге, +1, если дуга ориентирована от вершины, и -1 в противном случае.

Заметим, что любой неорграф  $G$  можно перевести в орграф  $D$  путем замены каждого ребра двумя противоположно направленными дугами. Поэтому многие результаты для неорграфов справедливы и для орграфов.

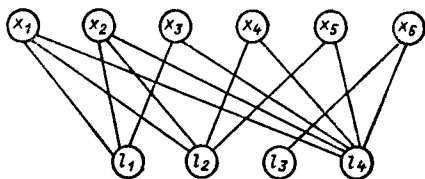
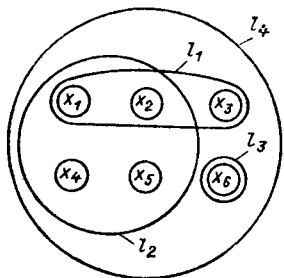


Рис. 4.26. Граф Кенига  $K(H)$

← Рис. 4.25. Гиперграф  $H = (X, E)$

В описанных выше графах использовались бинарные отношения на множестве вершин. Заметим, что в общем случае на множестве вершин можно задать  $k$ -местные отношения. Такое обобщение графа позволяет строить объекты, в которых каждое ребро может соединять не только две вершины, но и любое подмножество множества вершин.

Объект  $H = (X, E)$  будем считать *гиперграфом*, если он состоит из множества вершин  $X$  и множества ребер  $E$ , причем каждое ребро  $l_i \in E$  представляет собой некоторое подмножество вершин, т. е.  $l_i \subseteq X$ . Если  $\forall l_i \in E (|E| = 2)$ , то гиперграф  $H$  преобразуется в граф  $G$  без изолированных вершин. На рис. 4.25 показан пример гиперграфа  $H = (X, E)$ ,  $|X| = 6$ ,  $|E| = 4$ ; ребро  $l_3$  с  $|l_3| = 1$  есть петля. Ребрами являются:  $l_1 = \{x_1, x_2, x_3\}$ ,  $l_2 = \{x_1, x_2, x_4, x_5\}$ ,  $l_3 = \{x_6\}$ ,  $l_4 = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ . В гиперграфе  $H = (X, E)$  две вершины считаются смежными, если существует ребро  $l_i$ , содержащее эти вершины. Соответственно два ребра являются смежными, если их пересечение — непустое подмножество.

Матрицей инцидентности гиперграфа называют матрицу  $I(H) = \|a_{i,j}\|_{m \times n}$ ,  
причем

$$a_{i,j} = \begin{cases} 1, & \text{если } x_j \in l_i; \\ 0, & \text{если } x_j \notin l_i. \end{cases}$$

Для гиперграфа  $H$  (рис. 4.25) матрица инцидентности примет вид

$$I(H) = \begin{vmatrix} & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\ l_1 & 1 & 1 & 1 & 0 & 0 & 0 \\ l_2 & 1 & 1 & 0 & 1 & 1 & 0 \\ l_3 & 0 & 0 & 0 & 0 & 0 & 1 \\ l_4 & 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix}$$

При решении некоторых задач конструирования возникает необходимость в установлении соответствия между гиперграфом  $H = (X, E)$  и графом  $K(H) = (X, E, V)$ , который называют графом Кенига. Граф  $K(H)$  является двудольным, причем  $X$  — это одно подмножество его вершин ( $X$  — множество вершин соответствующего гиперграфа);  $E$  — это второе подмножество его вершин, т. е. множество ребер соответствующего гиперграфа. При этом вершины  $x_i \in X$  и  $l_j \in E$  в  $K(H)$  смежны тогда и только тогда, когда в гиперграфе  $H$  вершина  $x_i$  принадлежит ребру  $l_j$ . На рис. 4.26 приведен граф Кенига для гиперграфа  $H$  (см. рис. 4.25).

В некоторых случаях для построения схем используют расплывчатые графы и гиперграфы.

Граф  $\hat{G} = (X, U)$  называют *расплывчатым*, если для каждой вершины  $x_i \in X$  множество  $U$  является расплывчатым. Множество  $U$  характеризуется функцией принадлежности  $\mu_u(x)$ , принимающей значения из отрезка  $[0, 1]$ . Очевидно, что если  $\mu_u(x)$  для любых  $x, y \in X$  принимает значения 0 или 1, то расплывчатый граф  $\hat{G}$  становится обыкновенным графом. Для расплывчатого гиперграфа функция принадлежности определяется так же, как и для расплывчатого графа.

При решении задач автоматизации основные свойства и характеристики объектов описывают с помощью формальных математических объектов, обеспечивающих адекватность и сохраняющих наглядность и необходимую содержательность. При решении задач с помощью САПР и при разработке компонентов КСАП возникает необходимость построения различных ММ и выбора из них наиболее приемлемой.

Сформулируем основные требования, предъявляемые к ММ: адекватность и простота представления исходного объекта; информационная сложность, т. е. возможность перехода от одной ММ к другой, от объекта к модели и обратно; разумный объем памяти ЭВМ, отводимый для хранения информации о модели; степень разработанности математического аппарата для оперирования с ММ; простота обработки.

Очевидно, что самой лучшей моделью является сам объект. Построение его ММ вызвано попытками формализовать и алгоритмизовать процесс проектирования, а также необходимостью минимизации объема памяти ЭВМ для

представления модели. В этой связи рассматриваемые модели не могут полностью отвечать всем требованиям описания объекта, а могут лишь описывать его части. Для этой цели используют теоретико-множественные подходы.

Для правильного моделирования объекта необходимо определить все свойства объекта и каждый раз подбирать абстрактную формальную модель, чтобы между объектом и его моделью можно было устанавливать различные виды изоморфизма.

В конструкторском проектировании выделяют ММ схем (структурных, функциональных, электрических), монтажного пространства, самих конструкций.

Использование графотеоретических моделей объектов сохраняет всю наглядность и содержательность описываемых объектов и позволяет строить формальные алгоритмы конструирования, которые легко обрабатываются на ЭВМ.

Любая функциональная или принципиальная схема объекта состоит из набора элементов, соединенных между собой заданным образом. Тогда схему ЭВА можно рассматривать как некоторое множество элементов  $x_1, x_2, \dots, x_n$ , соединенных между собой цепями из множества  $E$ .

Такое представление схемы обычно называют схемой соединений или коммутационной схемой. На рис. 4.27 показан условный фрагмент схемы. Каждый элемент схемы имеет некоторое множество соединительных выводов, которые будем называть множеством контактов и обозначать  $C = \{c_1, c_2, \dots, c_p\}$ . Кроме элементов и контактов в схеме имеются внешние контакты  $C_0$ , которые осуществляют связь рассматриваемой схемы с другими.

Два контакта считаются связанными, если объединяются одной электрической цепью. Под электрической цепью

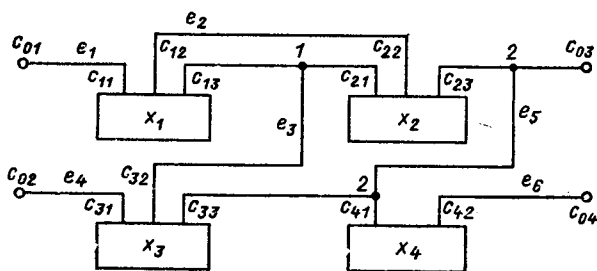


Рис. 4.27. Условный фрагмент схемы

понимается некоторое множество эквипотенциальных контактов.

Рассмотрим несколько способов задания схем РЭА и ЭВА графами, гиперграфами и их матричными и списковыми эквивалентами.

Зададим схему в виде графа  $G = (X \cup E \cup C, U)$ , где  $X$  — вершины графа, соответствующие элементам схемы;  $E$  — вершины, соответствующие цепям схемы;  $C$  — вершины, соответствующие контактам элементов. Множество ребер  $U$  состоит из элементных  $F$  и сигнальных  $W$  ребер, причем  $U = F \cup W$ . Ребра-подмножества  $F$  определяют принадлежность контактов из  $C$  элементам  $X$  и задаются парами  $(x_i,$

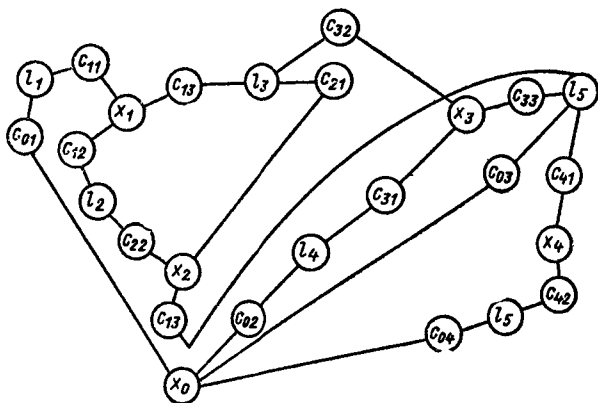


Рис. 4.28. Граф  $G = (X \cup E \cup C, U)$  фрагмента схемы

$c_k$ ). Ребра-подмножества  $W$  задают вхождение контактов из  $C$  в цепи  $E$  и описываются парами  $(c_k, l_j)$ . На рис. 4.28 показан граф схемы рис. 4.27. Обычно граф  $G$  задается в виде двух матриц

$$A_1 = \| \| a_{ij}^1 \| \| |E| \cdot |C| \text{ и } a_{ij}^1 \begin{cases} 1, \text{ если контакт } c_i \in l_j; \\ 0 \text{ в противном случае;} \end{cases}$$

$$A_2 = \| \| a_{ij}^2 \| \| |X| \cdot |C| \text{ и } a_{ij}^2 \begin{cases} 1, \text{ если } c_j \in X_i; \\ 0 \text{ в противном случае.} \end{cases}$$

Граф также задают в виде матрицы цепей  $T = \| t_{ij} \|_{n \times k}$ , строки которой соответствуют элементам схемы, а столбцы — контактам. Если элементы имеют различное число выводов, то в качестве  $k$  принимается  $\max K_i, i = 0, 1, \dots$  Эле-



мент  $t_{ij}$  — номер цепи, связанный с контактом  $c_j$  элемента  $x_i$ . Заметим, что если в матрице  $T$  элементы  $t_{ij} = t_{pq} = \dots = t_{\alpha\beta}$ , то это означает, что контакты  $j, q, \dots, \beta$ , расположенные на элементах с индексом  $i, p, \dots, \alpha$ , принадлежат одной электрической цепи (одному узлу). Для фрагмента схемы рис. 4.27 матрица имеет вид

$$T = \begin{array}{c|cccc} & c_1 & c_2 & c_3 & c_4 \\ x_1 & l_1 & l_2 & l_3 & - \\ x_2 & l_3 & l_2 & l_5 & - \\ x_3 & l_4 & l_3 & l_5 & - \\ x_4 & l_5 & l_6 & - & - \\ x_0 & l_1 & l_4 & l_5 & l_6 \end{array}$$

Например, элемент  $t_{33}$  матрицы  $T$  означает, что цепь  $l_5$  соединена с контактом  $c_3$  в элементе схемы  $x_3$ .

При решении задач компоновки и покрытия на конструкторском этапе проектирования между входами и выходами логических элементов схем устанавливаются различия. Они реализуются путем приписывания ребрам графа схемы направления. Входной сигнал логического элемента исходит из соответствующей вершины, а выходной сигнал направлен к вершине. Каждое ребро имеет вес, равный номеру контакта, что позволяет полностью идентифицировать схему коммутации. Тогда фрагмент схемы рис. 4.27 можно представить в виде двудольного орграфа (рис. 4.29, а).

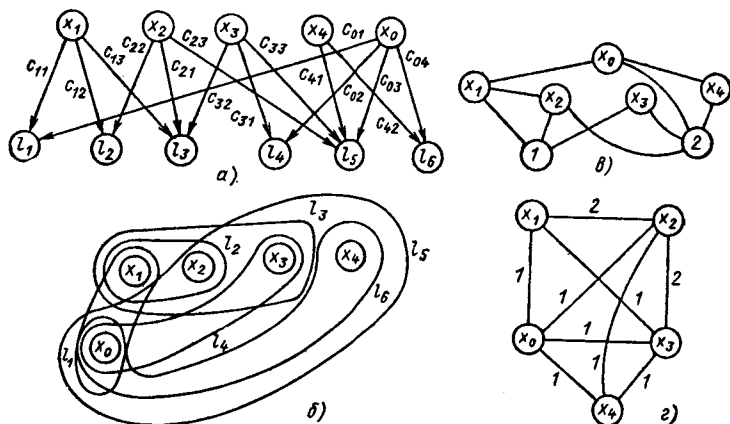


Рис. 4.29. Двудольный орграф (а), гиперграф (б), граф  $G=(X \cup V, U)$  (в) и взвешенный граф (г) фрагмента схемы, изображенной на рис. 4.27

Так как любой двудольный граф может быть представлен гиперграфом, то иногда коммутационные схемы удобно задавать гиперграфами (рис. 4.29, б). Основное преимущество такого задания — «плавающая» информация о цепях, каждая из которых может быть представлена любым из покрывающих деревьев.

Иногда удобно задать схему в виде графа  $G = (X \cup V, U)$ , где  $V$  — узловые точки схемы (рис. 4.29, в).

Основное преимущество всех описанных моделей — адекватное задание в смысле планарности. Доказано, что если один из описанных графов схемы планарен, то планарна и представляемая им схема.

Существует много модификаций описанных методов. Одним из них является построение взвешенного графа схемы. Для этого подсчитывают для каждой пары элементов число соединяющих их цепей. Затем строят граф  $G = (X, U)$ , в котором вершины соответствуют элементам, а ребра  $u_{ij}$  с весами — числу цепей между  $x_i$  и  $x_j$ . Для фрагмента схемы взвешенный граф показан на рис. 4.29, г.

Иногда используют способ перехода от схемы к графу, когда контакты элементов отображаются вершинами графа  $G = (X, U)$ , а соединения между ними — ребрами. При этом полученный граф представляет собой объединение полных подграфов, соответствующих узлам схемы. Если в каждом полном подграфе выделить покрывающее все вершины данного подграфа дерево (выполнить «развязку узлов»), то граф  $G$  изобразится множеством несвязных деревьев, т. е. лесом.

Для более полного описания функциональных и конструктивных возможностей схем применяют в качестве модели гиперграфы с помеченными вершинами и ребрами. Множество вершин  $X$  гиперграфа  $H = (X, E)$  интерпретирует множество элементов исходной схемы и внешние разъемы, множество ребер  $E$  — множество цепей в схеме. Каждой вершине  $x_i \in X$  присваивают метку, характеризующую тип элемента, а каждому ребру  $l_j \in E$  — веса, характеризующие качество контактов, принадлежащих одной цепи, и направление распространения сигнала.

Отметим, что существуют и другие модели, производные от описанных, причем основные критерии использования любой модели — это затраты памяти ЭВМ, расчет вычислительных процедур и полиномиальная оценка алгоритмов.

## Контрольные вопросы и задания

1. Почему в САПР используется множество иерархически организованных моделей?

2. Что такое имитационное моделирование?

3. Приведите примеры полной модели и макромоделей из какой-либо предметной области.

4. Для элементарной дифференцирующей RC-цепочки запишите математическую модель в инвариантной, алгоритмической, аналитической и схемной формах.

5. В чем заключаются основные отличия методов конечных разностей и конечных элементов?

6. Приведите пример математической модели какого-либо объекта на микроуровне.

7. Поясните физический смысл уравнения непрерывности.

8. Что такое область адекватности модели?

9. Для задачи теплопередачи в стержне, описываемой одномерным уравнением теплопроводности, запишите систему разностных уравнений при разделении стержня на  $n$  участков.

10. Для задачи предыдущего пункта разделите стержень на  $n$  конечных элементов. Задайтесь линейной аппроксимацией температуры от  $x$  (направление оси  $x$  выбрано вдоль стержня). Запишите выражения для координатных функций. Выполните алгебраизацию задачи, задавшись видом функционала, характеризующего качество аппроксимации.

11. Приведите примеры компонентных и топологических уравнений для произвольной электронной схемы.

12. Запишите компонентные уравнения преобразовательного элемента, отображающего связь электрической и механической подсистем в электромагните.

13. Составьте эквивалентную схему цилиндра с поршнем.

14. Составьте эквивалентную схему механической системы, состоящей из платформы с закрепленным на ней с помощью упругой связи грузом.

15. В чем отличие метода узловых потенциалов от метода переменных состояния?

16. Как отображаются в моделях связи физически разнородных подсистем?

17. Для схемы, приведенной на рис. 4.10, а составьте матрицу инциденций.

18. Для схемы, показанной на рис. 4.30, составьте математическую модель по методу переменных состояния.

19. Для схемы, данной на рис. 4.30, составьте матрицу Якоби для модели, получаемой по методу узловых потенциалов.

20. В чем сущность моделирования систем по методу огибающей?

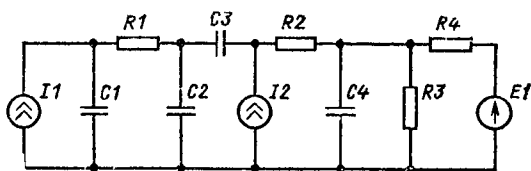


Рис. 4.30. Эквивалентная схема

21. Поясните сущность трехзначного асинхронного моделирования.
22. Проведите сравнительный анализ синхронных и асинхронных моделей логических схем.
23. Составьте синхронную модель для схемы, изображенной на рис. 4.31.

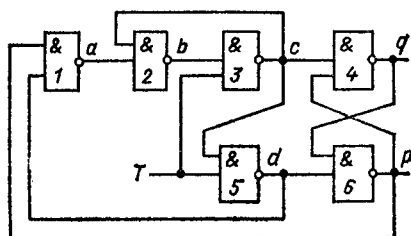


Рис. 4.31. Функциональная схема

24. Дайте основные определения расплывчатого множества.
25. Пусть  $X = (1, 2, 3, 4)$ ; постройте пример расплывчатого множества  $A$  на  $X$ .
26. Определите расплывчатое отношение на множествах  $A = \{ТЭЗ, ИМС\}$ ;  $B = \{\text{плата, стойка}\}$ .
27. Приведите смешанный граф, орграф и неорграф  $G = (X, U)$  с  $|X| = 12$ ,  $|U| = 16$ .
28. Постройте структурные схемы алгоритмов выделения из графа суграфов и подграфов с заданным числом ребер.
29. В полном графе  $K_n$ ,  $n = 7$ , выделите одно из  $n^{n-2}$  покрывающих деревьев.
30. Постройте граф  $G = (X, U)$  с  $n = 5$ ,  $m = 10$ . Задайте его с помощью матриц смежности и инцидентности. Постройте алгоритм перехода от одной матрицы к другой.
31. Подсчитайте число суграфов, включая изоморфные, в графе  $G = (X, U)$ ,  $|X| = n$ .
32. Предложите методы построения графов с эйлеровыми и гамильтоновыми циклами на заданных наборах вершин и ребер.
33. Приведите фрагмент схемы и для него постройте граф  $G = (X \cup E \cup C, U)$ , матрицы  $A_1, A_2, T$ , граф  $G = (X \cup E, U)$ , гиперграф  $H$ , двудольный орграф, мультиграф, граф  $G = (X, U)$ .

## 5.1. ТРЕБОВАНИЯ К МЕТОДАМ АНАЛИЗА

**Многообразие применяемых методов.** Анализ технических объектов в САПР основан на математическом моделировании, т.е. на исследовании проектируемых объектов путем оперирования их математическими моделями.

В предыдущей главе показано, что функциональными моделями проектируемых объектов на макроуровне являются системы ОДУ, которые могут быть представлены в общем виде (4.38), либо предварительно приведены линеаризацией к виду (4.39), либо алгебраизацией и линеаризацией (4.40). К таким же формам уравнений с помощью методов конечных разностей или конечных элементов приводятся ММ объектов на микроуровне.

Таким образом, математические модели объектов проектирования на микро- и макроуровнях сводятся к системам обыкновенных дифференциальных и конечных уравнений (под конечными уравнениями понимаются алгебраические и трансцендентные уравнения). Оперирование такими моделями в процедурах одновариантного анализа означает решение соответствующих уравнений. Поэтому методы одновариантного анализа на этих уровнях суть численные методы решения систем дифференциальных и конечных уравнений. То же относится к моделям и методам анализа аналоговой РЭА на метауровне.

Многовариантный анализ заключается в многократном повторении решения систем названных уравнений при варьировании внутренними и (или) внешними параметрами. Типовыми процедурами многовариантного анализа, реализуемыми в САПР, являются процедуры анализа чувствительности и статистического анализа.

Особенностью цифровой РЭА является использование на функционально-логическом уровне проектирования в качестве моделей систем логических уравнений, а в качестве методов анализа — методов решения этих систем.

В вычислительной математике известно большое количество методов численного решения систем уравнений. Однако применение большинства из них в САПР РЭА оказывается неэффективным, что объясняется особенностями ММ проектируемых объектов. Поэтому при создании мате-

матического обеспечения САПР усилия направлены не только на разработку математических моделей, но и в не меньшей мере на развитие численных методов и алгоритмов анализа. Поскольку эффективность метода зависит от особенностей решаемой задачи, целесообразна реализация в САПР более чем одного метода для каждого класса решаемых уравнений. Выбор метода в большинстве случаев возлагается на пользователя, что требует от него соответствующей подготовки в области численных методов анализа.

При неудачном выборе моделей или методов анализа пользователь САПР может столкнуться с рядом неприятностей: чрезмерной продолжительностью вычислений, несходимостью или неустойчивостью вычислительного процесса, малой точностью получаемых результатов.

В САПР целесообразно применять методы, исключающие возможность возникновения подобных ситуаций, т. е. методы, обладающие свойствами высокой экономичности, надежности и точности. Однако эти требования противоречивы и не всегда удается их одновременное удовлетворение в должной мере, поэтому важно уметь распознавать неблагоприятные ситуации и знать факторы, изменение которых может привести к исправлению положения.

**Экономичность.** Экономичность метода характеризуется затратами вычислительных ресурсов (машинного времени  $T_m$  и машинной памяти  $P_m$ ) на его применение в некоторых заранее оговоренных условиях (например, в тестовых задачах, в среднем по группе задач определенного класса и т. п.). На показатели  $T_m$  и  $P_m$  обычно оказывают влияние многие факторы и в первую очередь размерность решаемой задачи  $N$ . В качестве  $N$  принимают порядок решаемой системы уравнений, число элементов, из которых состоит моделируемый объект, и т. п.

При сравнении методов по экономичности часто не интересуются абсолютными показателями  $T_m$  и  $P_m$  в конкретной ситуации, а исследуют характер зависимости  $T_m$  и  $P_m$  от  $N$ . Наиболее эффективные методы имеют линейную или близкую к линейной зависимость показателей экономичности от сложности задачи. Для многих численных методов характерна полиномиальная зависимость  $T_m$  от  $N$ :

$$T_m \approx cN^\alpha,$$

где  $c$  — коэффициент пропорциональности. При  $\alpha \geq 2$  ме-

тод имеет заметное ограничение по сложности решаемых задач.

**Надежность.** Надежность метода оценивается как вероятность получения правильных результатов при использовании метода для решения задач заданного класса. Обычно условия применимости метода связаны с такими характеристиками ММ анализируемых объектов, которые пользователь не может оценить заранее имеющимися в его распоряжении средствами, поэтому возможны ситуации, когда вычислительный процесс оказывается неустойчивым или отсутствует сходимость, что может выражаться в заикливание или останове ЭВМ из-за переполнения разрядной сетки. В САПР стараются применять надежные методы. Однако высоконадежные методы часто характеризуются недостаточной экономичностью. В этом случае целесообразно комбинирование методов с переходом к трудоемким, но надежным методам только в результате автоматического распознавания ситуаций несходимости или неустойчивости вычислений.

**Точность.** Погрешности решения задачи определяются особенностями используемых моделей, численных методов, ограниченностью разрядной сетки ЭВМ. Каждый источник погрешности должен контролироваться, с тем чтобы погрешности не превысили предельно допустимые. Обычно точность результатов, получаемых с помощью численного метода, зависит от некоторых параметров, выбираемых «по умолчанию» или задаваемых среди исходных данных. С помощью этих параметров можно управлять погрешностями решения, но необходимо помнить, что снижение погрешностей возможно лишь до некоторого отличного от нуля предела и, как правило, сопровождается увеличением затрат машинного времени. Целесообразно в математическом обеспечении САПР иметь не один, а несколько методов одинакового целевого назначения, но с различными возможностями компромиссного удовлетворения противоречивых требований точности и экономичности.

Пользователь САПР должен также знать, что явления заикливания вычислений или переполнения разрядной сетки могут происходить не только из-за недостатков выбранного численного метода, но и из-за ошибок в задании исходных данных. Некоторые ошибки, связанные с нарушением формальных правил грамматики входного языка, распознаются автоматически. Однако ряд ошибок не может быть выявлен формальными средствами без участия

пользователя. Примерами таких ошибок могут быть ошибки в задании численных значений параметров или в задании соединений в анализируемой схеме. Если эти ошибки приводят к получению модели самовозбуждающейся схемы, то возможны явления заикливания и переполнения разрядной сетки.

**Направления повышения эффективности методов анализа.** Высокие размерности задач проектирования, необходимость выполнения многих вариантов решения систем уравнений при проектировании ЭВМ и других сложных технических объектов обуславливают большие затраты вычислительных ресурсов. Поэтому повышение экономичности методов анализа при соблюдении требований точности является актуальной задачей создания и совершенствования математического обеспечения САПР. Эта задача решается на основе идей и методов, группируемых в несколько направлений.

**Декомпозиция** — деление модели проектируемого объекта на части и отдельный анализ получающихся частей. Если  $T_m = cN^\alpha$ , то после деления модели на  $m$  равных частей затраты машинного времени приближенно оцениваются величиной  $ct(N/m)^\alpha$ , т. е. уменьшаются в  $q \approx m^{\alpha-1}$  раз. Однако отдельный анализ происходит в условиях принятия упрощающих предположений о взаимном влиянии частей, т. е. сопровождается увеличением погрешностей расчетов. Декомпозиция составляет основу блочно-иерархического подхода к проектированию. Это направление широко используют как в автоматизированных, так и в неавтоматизированных методах проектирования.

**Диакоптика** — направление исследования сложных систем по частям, отличающееся от декомпозиции тем, что отдельный анализ осуществляется без упрощающих предположений о влиянии частей друг на друга. Экономичность диакоптических методов соизмерима с экономичностью обычных декомпозиционных методов, а точность выше.

**Учет разреженности матриц** — направление экономичной организации операций над разреженными матрицами. Матрицу называют разреженной, если в ней преобладают нулевые элементы. Отказ от хранения нулевых элементов и реализация алгоритмов, в которых игнорируются арифметические действия над нулевыми элементами, могут дать значительную экономию  $T_m$  и  $P_m$ .

**Учет событийности** — направление, называемое также учетом временной разреженности моделей и основанное на



исключении из вычислительного процесса действий над неактивными переменными. *Неактивной* на интервале  $[t, t + \Delta t]$  переменной называют величину, изменения которой на этом интервале не превышают достаточно малого заранее заданного значения. В моделях сложных систем в каждый момент модельного времени большинство переменных неактивно. Моделирование, основанное на учете событийности, принято называть *событийным моделированием*. В алгоритмах событийного моделирования необходимо реализовать критерии своевременного включения переменных и соответствующих им частей моделей в группу неактивных (латентных) и своевременного их исключения из этой группы.

*Комбинирование моделей и методов* — одновременное использование при решении конкретной задачи нескольких разнотипных моделей или методов анализа одинакового целевого назначения. Комбинирование может быть пространственным, если разнотипные модели или методы применяют в разных частях общей модели, или временным, если их применяют на разных этапах вычислительного процесса. Пространственное комбинирование является частным случаем диакоптического подхода, так как подразумевает разделение модели на части (фрагменты). Повышение эффективности при комбинировании моделей и методов основано на использовании наиболее подходящих моделей и методов для данного фрагмента и данного этапа вычислений. Пространственное комбинирование моделей, относящихся к разным иерархическим уровням, называют *многоуровневым (или смешанным) моделированием*.

## 5.2. МЕТОДЫ АНАЛИЗА СТАТИЧЕСКИХ РЕЖИМОВ

Численные методы решения систем конечных уравнений. Большинство методов численного решения конечных уравнений

$$F(V) = 0 \quad (5.1)$$

относится к итерационным. В этих методах последовательные приближения к корню  $V^*$  системы (5.1) производятся по итерационной формуле

$$V_{i+1} = V_i + \Delta V_i,$$

начиная с выбранного исходного приближения  $V_0$ , где  $\Delta V_i$  — поправка на  $(i+1)$ -й итерации. Сходящийся вычи-

слительный процесс заканчивается при выполнении условия

$$\|V_{i+1} - V_i\| < \varepsilon_1 \text{ или } \|F(V_i)\| < \varepsilon_2,$$

где  $\varepsilon_1 > 0$ ,  $\varepsilon_2 > 0$ . В случае несходимости вычисления следует прекращать, если число итераций  $N$  превысит заданное  $M$ .

Численные методы решения (5.1) различаются способом вычисления поправки  $\Delta V_i$ . В методе простой итерации

$$\Delta V_i = hF(V_i),$$

где  $h$  — положительная константа (шаг);  $F(V_i)$  — вектор невязок.

Условие сходимости метода при решении системы линейных уравнений

$$|1 + h\lambda_j| < 1 \quad (5.2)$$

должно выполняться для всех  $j=1, 2, \dots, n$ . Здесь  $\lambda_j$  и  $n$  —  $j$ -е собственное значение и порядок матрицы коэффициентов системы (5.1).

Если система (5.1) нелинейна, то условие (5.2) используется как приближенное, фигурирующие в нем собственные значения относятся к матрице Якоби системы (5.1)

$$Я = \partial F(V) / \partial V.$$

Метод простой итерации характеризуется медленной сходимостью. Если система (5.1) плохо обусловлена, то значение  $h$ , при котором обеспечивается сходимость, мало и требуется большое число итераций

$$N \geq -0,5C \lg \varepsilon,$$

где  $C$  — спектральное число обусловленности матрицы коэффициентов;  $\varepsilon$  выбирают в диапазоне  $10^{-1}$ — $10^{-5}$ .

В большинстве задач скорость сходимости удается увеличить, применяя релаксационные методы, когда требуется такое упорядочение уравнений, чтобы диагональные элементы матрицы Якоби были отличны от нуля. На очередной  $(i+1)$ -й итерации вектор неизвестных

$$V_{i+1} = \omega \tilde{V}_i + (1 - \omega) V_i,$$

где  $\omega$  — параметр релаксации, выбирается в пределах 0—2 [при  $\omega > 1$  имеем метод последовательной верхней релаксации (ПВР), при  $\omega = 1$  — метод Зейделя, при  $\omega < 1$  — метод последовательной нижней релаксации (ПНР)].

Однако не во всех случаях релаксационные методы оказываются эффективнее метода простой итерации];  $\tilde{v}_{ki} = v_{ki} - Hf_k(\tilde{v}_{1i}, \tilde{v}_{2i} \dots \tilde{v}_{k-1,i}, v_{ki} \dots v_{ni})$  — элементы вектора  $V_i$ ;  $H = h/a_{kk}$ ,  $f_k(V) = 0$  —  $k$ -е уравнение упорядоченной системы (5.1);  $a_{kk}$  —  $k$ -й диагональный элемент матрицы Якоби.

В методе Ньютона поправка

$$\Delta V_i = -Y_i^{-1} F(V_i),$$

где  $Y_i^{-1}$  — обратная матрица Якоби, вычисленная на  $i$ -й итерации.

Метод Ньютона, характеризуемый высокой скоростью сходимости, широко распространен в процедурах автоматизированного проектирования. Однако по сравнению с предыдущими методами реализация метода Ньютона связана с увеличенными затратами памяти, требующимися для размещения матрицы Якоби. Кроме того, увеличивается трудоемкость вычислений на одной итерации.

В алгоритмах решения системы конечных уравнений по методу Ньютона для вычисления поправки  $\Delta V_i$  вместо обращения матрицы Якоби используют решение системы линейных алгебраических уравнений

$$Y_i \Delta V_i = -F(V_i),$$

т. е. метод Ньютона по своей сути есть метод линеаризации решаемой системы уравнений.

Важным фактором, управляя которым, можно добиться выполнения условий сходимости метода Ньютона, является близость точки начального приближения  $V_0$  к точке корня  $V^*$ . Это обстоятельство привело к появлению метода, повышающего вероятность сходимости метода Ньютона и называемого *методом продолжения решения по параметру*. В этом методе в решаемой системе уравнений выделяют параметр, влияющий на положение точки корня в пространстве фазовых переменных. Например, при анализе электронной схемы таким параметром может быть напряжение источника питания. Система (5.1) решается методом Ньютона многократно при ступенчатом изменении параметра. Пусть параметр  $E$  выбран так, что при  $E \rightarrow 0$  имеем  $V^* \rightarrow 0$ . Тогда при первом решении выбираем  $V_0 = 0$  и находим значение корня  $V_1^*$ , соответствующее начальному значению параметра  $E_1$ . Далее увеличиваем  $E$  и решаем систему уравнений при начальном приближении  $V_0 = V_1^*$

и т. д. вплоть до решения уравнений при истинном значении параметра  $E$ .

Следует отметить, что анализ статических состояний можно рассматривать как частный случай анализа переходных процессов, при котором определяется установившееся состояние объекта. Метод анализа статических состояний с помощью интегрирования системы дифференциальных уравнений, описывающей переходные процессы, называемый *методом установления*, широко используют в программах анализа проектируемых объектов.

**Численные методы решения систем линейных алгебраических уравнений (ЛАУ).** Процедура решения системы ЛАУ

$$AV = B \quad (5.3)$$

занимает важное место в математическом обеспечении САПР. Решение системы ЛАУ требуется при анализе статических режимов на каждой итерации метода Ньютона. Эта же процедура многократно выполняется при анализе переходных процессов после приведения математической модели объекта к алгебраизованной и линеаризованной форме (4.40). При этом эффективность решения названных задач анализа в значительной мере зависит от эффективности используемых методов решения ЛАУ.

Для решения систем ЛАУ в большинстве проектных процедур анализа используют *метод Гаусса* или его разновидности. Вычисления по методу Гаусса состоят из прямого и обратного ходов. При прямом ходе из уравнений последовательно исключают неизвестные, т. е. исходную систему приводят к виду, в котором матрица коэффициентов становится треугольной. Такое приведение основано на  $k$ -кратном применении формулы пересчета коэффициентов

$$a_{ij} := a_{ij} - a_{ik} a_{kj} / a_{kk} \quad (5.4)$$

ко всем элементам матрицы коэффициентов с  $i > k$  и  $j > k$ ,  $k = 1, 2, \dots, (n-1)$ , где  $n$  — порядок системы уравнений. При пересчете столбец свободных членов рассматривают как  $(n+1)$ -й столбец преобразуемой матрицы. Обратный ход заключается в вычислении неизвестных, начиная с определения из  $n$ -го уравнения единственной фигурирующей в нем неизвестной  $v_n$ . Далее из  $(n-1)$ -го уравнения находят неизвестную  $v_{n-1}$ , так как в этом уравнении кроме  $v_{n-1}$  фигурирует только неизвестная  $v_n$ , значение которой

к данному моменту вычислено, и т. д. вплоть до нахождения  $v_1$ .

**Методы разреженных матриц.** Если выполнять вычисления, пользуясь (5.4), для всех элементов матрицы коэффициентов, то экономичность метода Гаусса характеризуется кубической зависимостью затрат машинного времени  $T_m$  от порядка системы уравнений  $n$ . Это приводит к ограничению области целесообразного применения метода Гаусса значениями  $n$  в несколько десятков. Однако во многих практических задачах  $n$  имеет порядок сотен или тысяч. Применение метода Гаусса к таким задачам оказывается эффективным, если учитывать свойство разреженности матрицы коэффициентов в системе решаемых уравнений (5.3).

*Разреженной* называют ту матрицу, в которой преобладают элементы, равные нулю. Разреженность  $S$  оценивается отношением числа нулевых элементов к общему числу элементов матрицы. Анализ показывает, что в математических моделях большинства порегируемых объектов число ненулевых элементов пропорционально первой степени  $n$ . Поэтому если учитывать разреженность матрицы, то  $T_m$  можно сделать линейной функцией  $n$  и существенно расширить пределы эффективного применения метода Гаусса. Учет разреженности при этом заключается в том, что арифметические действия по (5.4) не производят, если выполняется хотя бы одно из условий  $a_{ik}=0$  или  $a_{kj}=0$ .

Следует различать исходную и итоговую разреженность матрицы. Исходная разреженность определяется числом ненулевых элементов, имеющих в матрице до начала выполнения исключений неизвестных по методу Гаусса. Такие *нулевые элементы* называют *первичными*. Однако в процессе вычислений по (5.4) некоторые коэффициенты  $a_{ij}$ , бывшие нулевыми, могут стать ненулевыми. Такие коэффициенты называют *вторичными ненулевыми элементами*. Итоговая разреженность определяется суммарным числом первичных и вторичных ненулевых элементов, и эффективность учета разреженности матрицы тем выше, чем больше итоговая разреженность.

Итоговая разреженность  $S_{ит}$  матрицы зависит от ее исходной разреженности и от порядка, в каком расположены уравнения и неизвестные (строки и столбцы матрицы). Существует некоторый оптимальный порядок, при котором  $S_{ит}$  максимальна. Процедура упорядочения строк и столбцов матрицы с целью максимизации  $S_{ит}$  имеет

важное значение для эффективности вычислений и включается в алгоритм метода Гаусса. В программах анализа используют приближенные алгоритмы максимизации  $S_{ит}$ . Простейший из них — алгоритм, по которому строки матрицы коэффициентов располагаются в порядке увеличения числа ненулевых элементов в них.

Эффективность учета разреженности матриц проявляется и в экономии оперативной памяти, так как в алгоритмах, учитывающих разреженность, требуется хранение в памяти ЭВМ только ненулевых элементов.

**Метод прогонки.** Примерами сильно разреженных матриц являются матрицы Якоби в системах конечных уравнений, получаемых по методам конечных разностей или конечных элементов из дифференциальных уравнений в частных производных. Если алгебраизация дифференциального уравнения производится на основе регулярной сетки, то разреженная матрица Якоби оказывается ленточной, т. е. матрицей, у которой ненулевые элементы располагаются только на  $k$  главных диагоналях. Специфические особенности структуры ленточных матриц можно использовать для упрощения алгоритмов учета разреженности.

Для решения систем ЛАУ с трехдиагональными матрицами коэффициентов используют разновидность метода Гаусса, называемую *методом прогонки*. Нетрудно заметить, что в трехдиагональных матрицах при исключении очередной неизвестной  $v_{i-1}$  из системы уравнений пересчет по (5.4) следует производить только в отношении диагонального элемента  $a_{ii}$  и свободного члена  $i$ -го уравнения  $b_i$ . Обозначим преобразованные по (5.4) значения  $a_{ii}$  и  $b_i$  через  $r_i$  и  $q_i$  соответственно. Тогда прямой ход по методу Гаусса сводится к расчету коэффициентов  $r_i$  и  $q_i$ ,  $i=2, 3, \dots, n$ , по рекуррентным формулам, являющимся частными случаями (5.4):

$$r_i = a_{ii} - a_{i,i-1} a_{i-1,i} / r_{i-1}; \quad q_i = b_i - a_{i,i-1} q_{i-1} / r_{i-1}.$$

Начальные значения коэффициентов  $r_1 = a_{11}$  и  $q_1 = b_1$ . Обратный ход в методе прогонки также выполняют по очевидной рекуррентной формуле

$$v_i = (q_i - a_{i,i+1} v_{i+1}) / r_i,$$

при этом принимают  $v_{n+1} = 0$ .

Коэффициенты  $r_i$  и  $q_i$  называют прогоночными. В качестве прогоночных коэффициентов часто используют

$$l_i = -a_{i-1,i} / r_{i-1} \quad \text{и} \quad p_i = q_{i-1} / r_{i-1},$$

тогда обратный ход осуществляют по формуле

$$v_i = p_{i+1} + l_{i+1} v_{i+1}.$$

Алгоритмы метода прогонки в отличие от более общих алгоритмов учета разреженности матриц с нерегулярной структурой характеризуются большей простотой программной реализации.

**Выбор метода решения системы алгебраических уравнений.** Решение систем алгебраических уравнений (АУ) имеет место во многих проектных процедурах и прежде всего в процедурах функционального проектирования. Эффективность решения этих задач вносит существенный вклад в общую эффективность выполнения проектных процедур, поэтому необходимо правильно выбрать метод решения системы АУ. Такой выбор приходится осуществлять разработчику пакета прикладных программ (ППП) для подсистем функционального проектирования. Если же пакет выполнен открытым по отношению к численным методам решения систем АУ и, следовательно, содержит ряд модулей, реализующих альтернативные методы, то выбор метода возлагается на пользователя.

На эффективность применения метода оказывают влияние не только особенности самого метода, но и в не меньшей мере особенности решаемой задачи и используемой ЭВМ. Среди наиболее существенных особенностей задач, называемых ниже факторами, отметим размерность  $n$  (порядок системы уравнений), число обусловленности  $C$  и разреженность  $S$  матрицы Якоби, а среди особенностей ЭВМ — быстродействие  $B$ , определенное для класса научно-технических задач, емкость оперативной памяти и разрядность машинного слова. Разработчик ППП должен ориентироваться на некоторые диапазоны значений этих факторов, характерные для моделей проектируемых объектов в соответствующей предметной области. Эти диапазоны должны быть либо указаны в техническом задании на разработку ППП, либо спрогнозированы самим разработчиком на основе исследования статистических данных об имеющихся или предполагаемых моделях.

Собственно выбор метода основан на установлении взаимосвязи между факторами и характеристиками используемых ЭВМ, с одной стороны, и такими показателями эффективности применения метода, как время решения, вероятность получения правильного результата и его точность, с другой стороны.

Рассмотрим, как устанавливаются подобные взаимосвязи для основных методов решения АУ. Для этих методов

$$T_m = n\gamma I/B,$$

где  $T_m$  — затраты машинного времени;  $n$  — порядок решаемой системы АУ, принимаемый за оценку сложности задачи;  $\gamma$  — среднее число арифметических операций, приходящихся на единицу сложности задачи, на одной итерации;  $I$  — среднее число итераций.

Для решения системы линейных алгебраических уравнений (ЛАУ) вида  $AV=B$  выбирают либо метод Гаусса, либо итерационные методы.

Для метода Гаусса  $I=1$ , и если не учитывать разреженность матрицы коэффициентов  $A$ , то  $\gamma \approx 2(n^2/3 + 2n)$ . Неучет разреженности ограничивает целесообразность применения метода Гаусса решением задач только невысокой размерности. При  $n > 50$  учет разреженности становится необходимым. Для метода Гаусса при учете разреженности и оптимальном упорядочении строк и столбцов матрицы  $A$  в задачах проектирования технических объектов имеем  $\gamma = \text{const}$ . Так, для моделей переключательных электронных схем  $\gamma \approx 25$ , а для распределенных моделей с трехдиагональной матрицей коэффициентов при применении метода прогонки  $\gamma \approx 8$ .

Для решения систем ЛАУ итерационными методами с учетом разреженности матрицы коэффициентов имеем  $I > 1$ , а  $\gamma = 2Qn$ , где  $Q = 1 - S$  — насыщенность матрицы. Так как  $Q = K/n$ , где  $K$  — среднее арифметическое для числа ненулевых элементов в одной строке матрицы  $A$ , то  $\gamma = 2K$ . Так, для моделей переключательных электронных схем получаем по результатам статистических исследований  $\gamma \approx 7,8$ , т. е. одна итерация выполняется быстрее, чем по методу Гаусса. Однако из-за того, что  $I \gg 1$ , итерационные методы по показателю  $T_m$  практически всегда проигрывают методу Гаусса.

Решение систем нелинейных АУ выполняется итерационными методами, при этом на требуемое число итераций  $I$  в методе Ньютона решающее влияние оказывает выбор начального приближения, а в остальных итерационных методах — число обусловленности  $C$  матрицы Якоби решаемой системы уравнений.

В методе Ньютона, применяемом в рамках методов установления или продолжения решения по параметру, обыч-



но  $I$  не превышает трех. В случаях, если  $I$  превышает некоторый порог  $I_{\text{пр}}$  (например,  $I_{\text{пр}}=7$ ), лучше уменьшать значения коэффициентов, управляющих процессом установления, чем продолжать итерации при  $I > I_{\text{пр}}$ . Следует отметить, что при решении нелинейных АУ величина  $\gamma$  растет, так как при ее подсчете должны быть учтены затраты на вычисление элементов матрицы Якоби.

В методе простых итераций  $I$  может достигать неприемлемо больших значений, поэтому целесообразно ввести на  $I$  ограничение  $I_{\text{гр}}$  сверху. Если принять  $I_{\text{гр}}=1,5 \cdot 10^4$ , то из соотношения  $I_{\text{гр}} = -0,5 \zeta \lg \varepsilon$  при  $\varepsilon = 10^{-3}$  получаем, что метод простых итераций можно применять только к решению системы уравнений, у которых матрица Якоби имеет  $\zeta < 10^4$ . Методы Зейделя, Якоби, последовательной верхней релаксации (ПВР) имеют аналогичный характер зависимости  $I$  от  $\zeta$ , хотя скорость сходимости у них часто оказывается несколько выше, чем в методе простых итераций.

Экономичность метода решения систем АУ определяется также затратами оперативной памяти. При неучете разреженности только на хранение матрицы Якоби нужно  $n^2$  ячеек памяти. Поэтому если для одного слова используется 8 байт, то при  $n=100$  для хранения требуется 80 кбайт, а при  $n=500$  — уже 2 Мбайт. Итак, подтверждается вывод о необходимости учета разреженности при решении задач с  $n > n_{\text{пр}}$ , где  $n_{\text{пр}}$  зависит от характеристик используемой ЭВМ и, как правило, составляет несколько десятков. В задачах анализа распределенных моделей, в которых  $n$  может превышать  $10^4$ , экономичность метода по затратам машинной памяти становится одной из важнейших характеристик. В таких случаях применяют либо релаксационные методы, либо метод Ньютона с использованием на каждой итерации метода Гаусса, но в рамках рассматриваемого ниже диакоптического подхода.

На точность решения задачи оказывают влияние задаваемые пользователем в исходных данных значения допустимых погрешностей  $\varepsilon_1$  или  $\varepsilon_2$ , а также обусловленность модели. Однако задаваемые значения  $\varepsilon_1$  или  $\varepsilon_2$  могут вообще оказаться недостижимыми или из-за несходимости, или из-за слишком медленной сходимости вычислительного процесса. Поэтому если создаваемый ППП ориентирован на решение систем уравнений с широким диапазоном значений  $\zeta$ , то нужно принимать специальные меры по обеспечению точности решения. При реализации метода Гаусса

нужно перейти к представлению чисел в ЭВМ с повышенной разрядностью (например, с удвоенной разрядностью, если используется ЭВМ единой серии, в которой разрядность машинного слова равна 32), в случае метода простых итераций — к уменьшению шага  $h$ .

Надежность применения метода определяется не только фактом принципиальной сходимости к корню, но и тем, каковы затраты времени  $T_m$  на получение решения с требуемой точностью. Неадекватность итерационных методов проявляется либо при неудачном выборе начального приближения к корню (метод Ньютона), либо при плохой обусловленности задачи (методы релаксационные и простых итераций), либо при повышенных требованиях к точности решения (метод простых итераций), либо при высокой размерности задач (метод Гаусса при неучете разреженности). Поэтому при создании узкоспециализированных программ необходимы предварительный анализ особенностей ММ заданного класса задач (значений  $n$ ,  $C$ , допустимых погрешностей) и соответствующий выбор конкретного метода. При создании ППП с широким спектром решаемых задач необходима реализация средств автоматической адаптации метода решения к конкретным условиям. Такая адаптация в современных ППП чаще всего применяется в рамках методов установления или продолжения решения по параметру.

### 5.3. МЕТОДЫ АНАЛИЗА ПЕРЕХОДНЫХ ПРОЦЕССОВ

Классификация методов численного интегрирования систем обыкновенных дифференциальных уравнений (ОДУ). Методы численного интегрирования ОДУ являются методами преобразования дифференциальных уравнений в алгебраические. После дискретизации независимой переменной  $t$  система ОДУ

$$\Psi(\dot{V}, V, t) = 0$$

в каждой точке  $t_k$  этой переменной представляется в виде системы  $n$  алгебраических уравнений

$$\Psi(\dot{V}_k, V_k, t_k) = 0 \quad (5.5)$$

с  $2n$  неизвестными  $\dot{V}_k = \dot{V}(t_k)$  и  $V_k = V(t_k)$ . Система (5.5) доопределяется уравнениями

$$\dot{V}_k = f(V_k), \quad (5.6)$$

задаваемыми выбранным методом численного интегрирования. Система алгебраических уравнений (5.5), (5.6) решается в каждой точке  $t_k, k=1, 2, \dots, III$ , где  $III$  — число точек дискретизации (шагов интегрирования).

Формулу численного интегрирования (5.6), в которой в качестве неизвестных величин фигурируют  $\dot{V}_k$  и  $V_k$ , и соответствующие этой формуле методы интегрирования называют *неявными*. В неявных формулах кроме  $\dot{V}_k$  и  $V_k$  могут присутствовать значения переменных  $\dot{V}$  и (или)  $V$  в  $p$  предыдущих точках дискретизации  $t_{k-i}, i=1, 2, \dots, p$ . При  $p \geq 2$  метод интегрирования называют *многошаговым*. Следует отметить, что к моменту решения системы (5.5), (5.6) значения  $\dot{V}_{k-i}$  и  $V_{k-i}$  для  $i \geq 1$ , фигурирующие в (5.6), уже вычислены на предыдущих шагах. Название метода «многошаговый» происходит из-за использования в формуле интегрирования результатов нескольких предыдущих шагов. Величину  $p$  при этом называют *порядком многошагового метода*. Вместо  $\dot{V}_{k-i}$  или  $V_{k-i}, i \geq 2$ , в формуле интегрирования могут присутствовать производные  $V$  по  $t$  порядка выше первого или заменяющие их результаты некоторых дополнительных вычислений на данном шаге. В этом случае метод называется *одношаговым*, а порядок одношагового метода совпадает с порядком старшей из используемых производных.

Систему алгебраических уравнений, решаемых на каждом шаге численного интегрирования, можно записать также в следующем виде:

$$\Psi(\dot{V}_{k-1}, V_{k-1}, t_{k-1}) = 0, \quad (5.7)$$

$$\dot{V}_{k-1} = f(V_k), \quad (5.8)$$

где  $\dot{V}_{k-1}$  и  $V_k$  — неизвестные величины;  $V_{k-1}$  вычислены на предыдущем шаге.

Формулу численного интегрирования (5.8) и соответствующие ей методы интегрирования называют *явными*. Явные методы по аналогии с неявными могут быть одно- и многошаговыми, аналогично определяются порядки явных методов.

Очевидно, что необязательно на каждом шаге интегрирования численно решать систему из  $2n$  конечных уравнений. В большинстве случаев выполняют предварительное исключение неизвестного вектора  $\dot{V}_k$  из (5.5) или  $\dot{V}_{k-1}$  из (5.7) с помощью формул интегрирования (5.6) или (5.8)

в общем виде и на каждом шаге численно решают систему  $n$  уравнений с неизвестным вектором  $V_k$ .

**Методы численного интегрирования ОДУ, применяемые в САПР.** В практике машинных вычислений наиболее распространены для решения ОДУ методы Гира, Адамса и Рунге — Кутты.

Общий вид формул интегрирования в неявных методах Гира

$$\dot{V}_k = \left( \sum_{i=0}^p a_i V_{k-i} \right) / h_k,$$

где  $h_k = t_k - t_{k-1}$  — величина  $k$ -го шага интегрирования;  $a_i$  — коэффициенты, значения которых зависят от порядка  $p$  метода и величин порядков последних шагов. Формулы Гира называют также *формулами дифференцирования назад* (ФДН) по той причине, что в них аппроксимация производных в точке  $t_k$  производится с помощью значений функций, относящихся к данному и предыдущим моментам времени. Формула Гира при  $p=1$  совпадает с *неявной формулой Эйлера*

$$\dot{V}_k = (V_k - V_{k-1}) / h_k.$$

Часто применяют формулу Гира второго порядка, называемую также *формулой Шихмана*, которая при  $h = \text{const}$  имеет вид

$$\dot{V}_k = \left( \frac{3}{2} V_k - 2V_{k-1} + \frac{1}{2} V_{k-2} \right) / h.$$

Общий вид формул интегрирования в *явных методах Адамса* при  $p \geq 2$

$$\dot{V}_{k-1} = a_1 (V_k - V_{k-1}) / h_k + \sum_{i=2}^p a_i \dot{V}_{k-i};$$

в *неявных методах Адамса* при  $p \geq 2$

$$\dot{V}_k = a_0 (V_k - V_{k-1}) / h_k + \sum_{i=1}^{p-1} a_i \dot{V}_{k-i}.$$

Явная формула Адамса при  $p=1$  называется также *явной формулой Эйлера*:

$$\dot{V}_{k-1} = (V_k - V_{k-1}) / h_k;$$

явная формула Адамса при  $p=2$

$$\dot{V}_{k-1} = \frac{2}{3} (V_k - V_{k-1}) / h_k + \dot{V}_{k-2} / 3.$$

Неявный метод Адамса второго порядка точности называют также *методом трапеций*, ему соответствует формула интегрирования

$$\dot{V}_k = 2(V_k - V_{k-1})/h_k - \dot{V}_{k-1}. \quad (5.9)$$

Рассмотренные методы при  $p \geq 2$  являются многошаговыми. К одношаговым методам относится метод Рунге — Кутты.

В САПР распространены неявные методы трапеций и Гира, а в отдельных случаях применяют явный метод Эйлера.

Использование методов возможно, если порождаемый ими вычислительный процесс является устойчивым. Неустойчивость вычислений может возникнуть в связи с катастрофическим ростом погрешностей. Различают локальную погрешность интегрирования, допущенную на данном шаге интегрирования, и погрешность, накопленную к моменту  $t_k$  за все предыдущие шаги. В неустойчивых методах погрешность решения увеличивается от шага к шагу, что приводит к полному искажению результатов и, возможно, к переполнению разрядной сетки.

Среди рассмотренных методов интегрирования имеются А-устойчивые и ограниченно устойчивые методы. А-устойчивым называют метод, при применении которого к интегрированию системы линейных ОДУ

$$\dot{V} = AV \quad (5.10)$$

с начальными условиями  $V_0 \neq 0$  погрешность решения стремится к нулю при любом значении постоянного шага  $h > 0$  и при  $t_k \rightarrow \infty$ . Здесь  $A$  — постоянная матрица, собственные значения которой имеют отрицательные действительные части. К А-устойчивым относятся неявные методы Гира и Адамса первого и второго порядков точности.

Ограниченно устойчивыми являются остальные из рассмотренных методов, для них характерно сохранение устойчивости вычислений только при выполнении ограничений, накладываемых на значение шага интегрирования. Так, для явного метода Эйлера при  $h = \text{const}$  в задаче (5.10) условие устойчивости имеет вид неравенства

$$|1 + h\lambda_j| < 1, \quad (5.11)$$

которое должно выполняться для всех собственных значений  $\lambda_j$  матрицы  $A$ . Для матрицы  $A$  с отрицательными вещественными собственными значениями  $\lambda_j$  условие (5.11) мо-

жно представить в виде

$$0 < h < -2/\lambda_j.$$

Если при этом система уравнений (5.10) есть модель динамической системы (например, электронной схемы), то величины  $-1/\lambda_j$  принято называть постоянными времени  $\tau_j$ . Тогда условие устойчивости явного метода Эйлера приводится к виду

$$0 < h < 2\tau_{\min}, \quad (5.12)$$

где  $\tau_{\min}$  — минимальная среди постоянных времени моделируемого объекта.

Условия (5.11) или (5.12) устойчивости методов интегрирования в применении к нелинейным системам ОДУ можно рассматривать как приближенные, при этом под  $\lambda_j$  понимают собственные значения матрицы Якоби  $\mathbf{Y} = \partial \dot{\mathbf{V}} / \partial \mathbf{V}$ . Так как в нелинейных задачах элементы матрицы Якоби непостоянны, то непостоянны и ее собственные значения. Поэтому априорный выбор значения постоянного шага  $h$ , удовлетворяющего условиям устойчивости на всем интервале интегрирования  $[0, T_{\text{кон}}]$ , оказывается практически невозможным (случай гарантированного выполнения условий устойчивости за счет выбора  $h \ll \tau_{\min}$  неприемлем, так как приводит к чрезмерным затратам машинного времени).

Интегрирование с постоянным шагом нецелесообразно и в А-устойчивых методах, так как  $h$  влияет на точность и время решения. Влияние  $h$  на точность решения по-разному проявляется на различных участках моделируемого переходного процесса. Поэтому минимизация затрат машинного времени при соблюдении точностных ограничений возможна только в условиях интегрирования с переменным шагом.

Большинство алгоритмов автоматического выбора шага основано на контроле локальных погрешностей интегрирования. Локальные погрешности включают в себя погрешности методические, обуславливаемые приближенностью формул интегрирования, и округления, обуславливаемые представлением чисел с помощью ограниченного количества разрядов. Локальная методическая погрешность многошагового метода порядка  $p$ , допущенная на  $k$ -м шаге интегрирования, зависит от значения шага  $h_k$  и оценивается по формуле

$$\varepsilon_k = c \| \mathbf{V}^{(p+1)}(\tau) \| h_k^{p+1}, \quad (5.13)$$

где  $c$  — коэффициент, зависящий от характера и порядка

метода;  $\|V^{(p+1)}(\tau)\|$  — норма  $(p+1)$ -х производных  $V$  по  $t$ , вычисленных в точке  $\tau \in [t_{k-p}, t_k]$ . В алгоритмах интегрирования задаются значением допустимой локальной погрешности  $\delta$ , а значение шага  $h_k$  ограничивают сверху

$$\varepsilon_k < \delta, \quad (5.14)$$

при этом вектор  $(p+1)$ -х производных приближенно оценивают с помощью вектора  $(p+1)$ -х конечных разностей, вычисленных по уже найденным значениям  $V_{h-i}$ ,  $i=1, 2, \dots, (p+2)$ . Однако выполнение условия (5.14) с большим запасом приводит к заниженным значениям  $h_k$  и увеличивает  $T_m$ . Поэтому наряду с (5.14) вводят также ограничение на допущенную погрешность  $\varepsilon_k$  снизу и значение шага выбирают автоматически, например по следующему правилу:

$$h_{k+1} = \begin{cases} m_1 h_k & \text{при } \varepsilon_k > \delta_2; \\ h_k & \text{при } \delta_2 \geq \varepsilon_k \geq \delta_1; \\ m_2 h_k & \text{при } \varepsilon_k < \delta_1, \end{cases}$$

где  $m_1 < 1$ ;  $m_2 > 1$ ;  $\delta_2/\delta_1 = 2$ . (Значения  $m_1, m_2, \delta_2$  выбирают, исходя из имеющегося опыта вычислений по данному алгоритму.)

Для того чтобы сделать несущественным влияние погрешностей округления, предусматривают в соответствующих частях алгоритма вычисления с удвоенным числом разрядов, а также не допускают снижения значения шага ниже некоторого минимального уровня  $h_{\min}$ .

**Сравнение методов и обоснование их выбора для конкретных задач автоматизированного проектирования.** Эффективность метода численного интегрирования оценивается его влиянием на экономичность и точность вычислений.

Качественно характер влияния значения шага на погрешность интегрирования для различных методов можно представить с помощью графиков, изображенных на рис. 5.1. В области малых  $h < h_{\min}$  значительны погрешности

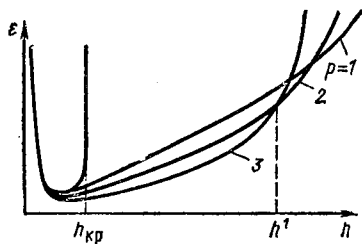


Рис. 5.1. Зависимость погрешности интегрирования от величины шага и порядка метода

округления. В явных методах наблюдается резкий рост погрешностей при  $h > h_{кр}$ , где  $h_{кр}$  — максимально допустимая по условиям устойчивости величина шага интегрирования. С ростом порядка метода снижаются погрешности интегрирования, но только при умеренных значениях  $h < h'$ . Поскольку величина  $h'$  зависит не только от порядков сравниваемых методов, но и от особенностей переходного процесса, что подтверждается формулой (5.13), то заранее определить, метод какого порядка точности даст оптимальный компромисс между показателями точности и экономичности, затруднительно. В связи с этим в программах анализа электронных схем распространен метод ФДН, основанный на автоматическом выборе не только значения шага, но и порядка  $p$  неявной формулы Гира. Если на предшествующем  $(k-1)$ -м шаге использовалась формула порядка  $p_{k-1}$ , то на следующем шаге выбор производился между формулами порядков  $p_{k-1}-1$ ,  $p_{k-1}$  или  $p_{k-1}+1$ . Выбирается та формула, использование которой при заданной погрешности обеспечивает наибольшую величину шага  $h_k$ . При этом порядок  $p_k$  формул Гира не должен выходить за пределы диапазона  $[1, 6]$ .

Среди неявных методов интегрирования при  $p = \text{const}$  применяют методы Эйлера, трапеций, Шихмана. Их положительными особенностями являются А-устойчивость и сравнительно малый объем памяти, требующийся для хранения результатов интегрирования, полученных на предыдущих шагах. Однако метод Эйлера не обеспечивает необходимой точности при анализе переходных процессов в слабодемпфированных системах. Метод трапеций в его первоначальном виде (5.9) имеет недостаток, заключающийся в появлении в численном решении ложной колебательной составляющей уже при сравнительно умеренных значениях шагов, поэтому метод трапеций удобен только при принятии мер, устраняющих ложные колебания. Значительное уменьшение ложных колебаний, но при несколько больших погрешностях, дает формула Шихмана.

В целом затраты машинного времени на анализ переходных процессов неявными методами существенно зависят от экономичности алгоритмов численного решения конечных уравнений, применяемых на каждом шаге интегрирования. Обычно для решения конечных уравнений используют метод Ньютона, тогда

$$T_m \approx bN^\alpha \text{ ИШ}, \quad (5.15)$$

где  $b$  — коэффициент пропорциональности, зависящий в ос-



новном от быстродействия используемой ЭВМ;  $N$  — показатель сложности анализируемого объекта;  $I$ ,  $III$  — число ньютоновских итераций на одном шаге и шагов интегрирования;  $\alpha \in [1, 3]$  и зависит от свойств выбранного алгоритма решения системы линейных алгебраических уравнений (ЛАУ) на каждой ньютоновской итерации. Если разреженность матрицы Якоби не учитывается, то  $\alpha = 3$  и возможности применения неявных методов ограничиваются задачами сравнительно малой размерности. Поэтому в САПР сложных объектов (таких, как БИС) необходим учет разреженности матриц. При этом  $\alpha$  в (5.15) оказывается в интервале  $[1, 2]$  и существенно повышает эффективность неявных методов. Величины  $I$  и  $III$  зависят от особенностей ММ схемы и характера анализируемых процессов. По классу задач анализа реакции электронных схем на одиночное импульсное воздействие получены средние типичные значения  $I = 1 \div 3$ , и значения  $III$ , равные нескольким десяткам—сотням.

Явные методы интегрирования целесообразно применять к решению систем ОДУ, представленных в нормальной форме Коши

$$\dot{\mathbf{V}} = \varphi(\mathbf{V}, t). \quad (5.16)$$

Тогда отпадает необходимость решения систем конечных уравнений на каждом шаге. Например, подставляя формулу Эйлера  $\mathbf{V}_{k-1} = (\mathbf{V}_k - \mathbf{V}_{k-1})/h_k$  в  $\dot{\mathbf{V}}_{k-1} = \varphi(\mathbf{V}_{k-1}, t_{k-1})$ , получаем явное относительно искомого вектора  $\mathbf{V}_k$  выражение

$$\mathbf{V}_k = \mathbf{V}_{k-1} + h_k \varphi(\mathbf{V}_{k-1}, t_{k-1}) \quad (5.17)$$

(явность этого выражения и послужила причиной названия явных методов интегрирования). Вычисления по (5.17) фактически сводятся к расчету правых частей системы ОДУ (5.16) и соответствуют значению  $\alpha$ , близкому к единице, в формуле (5.15). Малый объем вычислений на одном шаге и малый объем требующейся оперативной памяти — положительная черта явных методов. Однако в них могут стать недопустимо большими значения  $III$ . В самом деле,  $III = T_{\text{кон}}/h_{\text{ср}}$ , где  $T_{\text{кон}}$  — отрезок интегрирования, соизмерим с максимально допустимой постоянной времени  $\tau_{\text{max}}$ ;  $h_{\text{ср}}$  — средняя величина шага, из-за соображений устойчивости соизмерима с  $\tau_{\text{min}}$ . Теперь (5.15) принимает вид

$$T_{\text{м}} \approx bN\tau_{\text{max}}/\tau_{\text{min}}.$$

Очевидно, что явные методы применяются к анализу переходных процессов только в объектах с умеренными значениями  $C = \tau_{\text{max}}/\tau_{\text{min}}$  (обычно при  $C$  не более  $10^4$ — $10^5$ ). В то

же время в реальных задачах анализа фрагментов БИС значения  $\mathcal{C}$  могут оказаться значительно больше.

Таким образом, сравнение явных и неявных методов интегрирования ОДУ свидетельствует о большей универсальности последних. Поэтому неявные методы являются основными методами анализа переходных процессов в подсистемах схмотехнического проектирования современных САПР БИС и РЭА. Явные методы могут давать лучшие результаты только в отдельных случаях анализа объектов с хорошо обусловленными ММ и играют в современных САПР вспомогательную роль.

#### 5.4. МЕТОДЫ АНАЛИЗА ПОВЫШЕННОЙ ЭФФЕКТИВНОСТИ

**Диакоптические методы.** Диакоптические методы основаны на фрагментации модели сложного объекта, организации отдельных вычислений по фрагментам с периодическим согласованием результатов, получаемых в отдельных фрагментах. Диакоптические методы применяют для решения систем различных уравнений совместно с традиционными численными методами.

Для решения систем линейных алгебраических уравнений (ЛАУ)  $AV=B$  применяют *диакоптический вариант метода Гаусса*, основанный на приведении матрицы коэффициентов к блочно-диагональному виду с окаймлением (БДО). При анализе электронных схем этот вариант называют *методом подсхем*. В методе подсхем исходную схему разбивают на фрагменты (подсхемы). Фазовые переменные (например, узловые потенциалы) делят на внутренние переменные фрагментов и граничные переменные. Вектор фазовых переменных

$$V = (V_1, V_2 \dots V_l, V_L),$$

где  $V_i$  — подвектор внутренних переменных  $i$ -го фрагмента;  $l$  — число фрагментов;  $V_L$  — подвектор граничных переменных ( $L=l+1$ ). Матрица коэффициентов имеет структуру БДО

$A_{11}$	0	0	.	0	$A_{1L}$
0	$A_{22}$	0	.	0	$A_{2L}$
0	0	$A_{33}$	.	0	$A_{3L}$
0	0	0	.	$A_{lL}$	$A_{lL}$
$A_{L1}$	$A_{L2}$	$A_{L3}$	.	$A_{LL}$	$A_{LL}$

Прямой ход метода Гаусса состоит из  $L$  этапов. На  $i$ -м этапе исключаются переменные  $V_i$ , при этом пересчет коэффициентов по формуле Гаусса производится только в подсистеме уравнений

$$\begin{bmatrix} A_{ii} & A_{iL} \\ A_{Li} & A_{LL} \end{bmatrix} \begin{bmatrix} V_i \\ V_L \end{bmatrix} = \begin{bmatrix} B_i \\ B_L \end{bmatrix},$$

где  $V_i$  и  $V_L$  — подвекторы правых частей соответствующих уравнений.

Следовательно, решение одной сложной задачи заменяется решением  $L$  задач меньшей размерности, что при  $\alpha > 1$  в формуле (5.15) дает снижение  $T_m$ . Кроме того, при нехватке оперативной памяти в методе подсхем можно минимизировать число обменов информацией между оперативной и внешней памятью.

Для решения систем нелинейных конечных уравнений используют *диакоптический вариант метода Ньютона* с контролем сходимости итерационного процесса отдельно по выделенным фрагментам. Выполнение условий сходимости в  $i$ -м фрагменте является основанием для прекращения вычислений по уравнениям этого фрагмента. Очевидно, что раздельное интегрирование означает и раздельное решение подсистем ЛАУ, относящихся к отдельным фрагментам.

Раздельное по фрагментам интегрирование дифференциальных уравнений довольно просто организуется лишь при использовании явных методов. Покажем это на примере решения методом Эйлера системы ОДУ, представленной в нормальной форме Коши и разделенной на две подсистемы:

$$\dot{V} = \varphi_1(V, Z, t); \quad \dot{Z} = \varphi_2(V, Z, t),$$

где  $V$  и  $Z$  — подвекторы фазовых переменных, относящиеся к разным фрагментам (граничные переменные предполагаются включенными в один из подвекторов  $V$  или  $Z$ ). Обозначая величины шагов в подсистемах  $h$  и  $H$  и применяя формулу Эйлера для момента времени  $t_k$

$$\dot{V}(t_k) = \frac{V(t_k + h) - V(t_k)}{h}; \quad \dot{Z}(t_k) = \frac{Z(t_k + H) - Z(t_k)}{H},$$

имеем

$$V(t_k + h) = V(t_k) + h\varphi_1(V(t_k), Z(t_k), t_k),$$

$$Z(t_k + H) = Z(t_k) + H\varphi_2(V(t_k), Z(t_k), t_k),$$

откуда вычисляем  $V(t_k + h)$  и  $Z(t_k + H)$ . Пусть  $h < H$ . Для

выполнения очередного шага по переменным  $V$  нужно вычислить  $Z(t_k+h)$ , что осуществляют с помощью интерполяционного полинома степени  $p$ , построенного по  $p+1$  последним рассчитанным точкам кривой  $Z(t)$ . Здесь  $p$  — порядок метода интегрирования. При применении метода Эйлера  $p=1$ , и значение  $Z(t_k+h)$  вычисляют на основе линейной интерполяции по известным точкам  $Z(t_k)$  и  $Z(t_k+H)$ . После осуществления шага по переменным  $V$  шаги повторяют — очередной шаг делают по переменным  $V$  или  $Z$  в зависимости от того, в каком фрагменте моделируемое время имеет меньшее значение.

Интегрирование подсистем ОДУ с оптимальным для каждого фрагмента значением шага может привести к существенной экономии затрат машинного времени, особенно при применении неявных методов интегрирования. Однако организация неявного пофрагментного интегрирования оказывается более сложной, чем явного. Примеры методов пофрагментного неявного интегрирования — методы однонаправленных моделей и релаксации формы сигнала (РФС).

**Методы однонаправленных моделей и релаксации формы сигнала.** Модели многих сложных элементов являются однонаправленными. В них могут быть выделены входные и выходные фазовые переменные, причем выходные не влияют на входные. Примерами однонаправленных моделей служат большинство моделей логических элементов.

При представлении макромодели логического элемента в схемной форме ее типичная структура имеет вид, показанный на рис. 5.2. Входной блок  $I$  отражает входные ха-

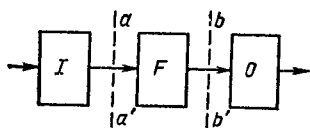


Рис. 5.2. Структура модели логического элемента

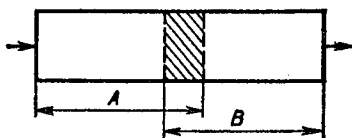


Рис. 5.3. Фрагментация схемы с перекрытием

рактеристики схемы, выходной блок  $O$  — выходное сопротивление, блок  $F$  — функциональные преобразования сигналов и задержку их распространения. Передача сигналов через линии  $a-a'$  и  $b-b'$  происходит только в направлении слева направо.

*Метод однонаправленных моделей* применяется при вы-

делении фрагментов в схеме в соответствии со следующими правилами:

- 1) границы фрагментов проходят через схемы однонаправленных моделей по линиям  $a-a'$  или  $b-b'$ ;
- 2) контур любой обратной связи должен полностью находиться в пределах одного фрагмента.

После выделения фрагментов необходимо их ранжирование — упорядочение в соответствии с последовательностью прохождения сигналов. Ранг 1 получают фрагменты, на входы которых поступают только внешние возбуждения. Ранг 2 получают фрагменты, на входы которых поступают сигналы с выходов фрагментов ранга 1 и, возможно, также внешние возбуждения, и т. д. Другими словами, ранг  $r$  присваивают фрагменту, на входы которого поступают сигналы с выходов только ранжированных фрагментов (внешние возбуждения рассматриваются как сигналы с выходов фрагментов ранга 0), причем старший из рангов предшествующих фрагментов равен  $r-1$ .

После фрагментации и ранжирования выполняют раздельное численное интегрирование подсистем дифференциальных уравнений, относящихся к различным фрагментам в порядке увеличения их рангов. Интегрирование выполняют на всем заданном отрезке интегрирования  $T_{\text{кон}}$ . При интегрировании уравнений фрагмента с рангом  $r$  в качестве входных воздействий используют результаты интегрирования уравнений фрагментов с более низкими рангами.

Раздельное интегрирование позволяет организовать вычисления в каждом фрагменте с оптимальным для фрагмента значением шага, что может привести к значительной экономии вычислительных затрат. Однако метод однонаправленных моделей имеет ограниченное применение из-за необходимости соблюдения указанных правил фрагментации. Эти ограничения устраняются в методе РФС.

*Метод РФС* является итерационным методом раздельного интегрирования дифференциальных уравнений. Условие однонаправленности моделей снимается благодаря введению фрагментации схем с перекрытием, поясняемой рис. 5.3. Заштрихованный участок соответствует подсхеме, включаемой при раздельном интегрировании и в фрагмент  $A$ , и в фрагмент  $B$ . Чем шире зона перекрытия, тем точнее учитывается нагрузка для фрагмента  $A$  и точнее рассчитываются входные сигналы для фрагмента  $B$ . Если в схеме нет межфрагментных обратных связей, то достаточно ранжирования фрагментов и выполнения одной итерации пофрагментного

интегрирования на интервале  $[0, T_{\text{кон}}]$ . При наличии межфрагментных обратных связей требуется выполнение нескольких итераций, с помощью которых форма сигналов на выходах фрагментов приближается к истинной (происходит релаксация формы сигналов).

Очевидно, что выполнение пофрагментного интегрирования на всем заданном интервале  $[0, T_{\text{кон}}]$  может привести к излишне большому числу итераций и к колебаниям фазовых переменных в процессе релаксации с излишне большой амплитудой, что потребует значительных затрат машинного времени. Во избежание этих неприятностей можно использовать *метод прогнозируемой реакции*. В отличие от метода РФС здесь пофрагментное интегрирование производится на отрезках  $[t_h, t_h + H_j]$ , где  $H_j$  — шаг прогноза, соизмеримый с длительностью задержки прохождения сигнала по  $j$ -му контуру обратной связи. Значение  $H_j$  определяют автоматически аналогично тому, как рассчитываются значения шагов интегрирования.

**Комбинированные методы и алгоритмы анализа.** При решении задач анализа в САПР получило достаточно широкое распространение временное комбинирование численных методов. Наиболее известны рассмотренные выше *алгоритмы ФНД* для численного интегрирования ОДУ, являющиеся алгоритмами комбинирования формул Гира. Другим примером временного комбинирования методов служат *циклические алгоритмы неявно-явного интегрирования* ОДУ. В этих алгоритмах циклически меняется формула интегрирования — следом за шагом неявного интегрирования следует шаг явного интегрирования. В базовом алгоритме неявно-явного интегрирования используют формулы первого порядка точности — формулы Эйлера. Такой комбинированный алгоритм оказывается реализацией А-устойчивого метода второго порядка точности, повышение точности объясняется взаимной компенсацией локальных методических погрешностей, допущенных на последовательных неявном и явном шагах. Следует отметить, что в качестве результатов интегрирования принимаются только результаты неявных шагов, поэтому в алгоритме комбинированного неявно-явного интегрирования устраняются ложные колебания, присущие наиболее известному методу второго порядка точности — методу трапеций.

Комбинирование неявных и явных формул интегрирования успешно применяют для повышения эффективности решения нестационарных двумерных задач на микроуровне в

рамках метода, называемого *методом переменных направлений*. Если на очередном временном шаге использовать неявные формулы аппроксимации производных по обеим пространственным координатам  $x_1$  и  $x_2$ , то потребуется решать систему конечных уравнений порядка  $pq$ , где  $p$  и  $q$  — уменьшенное на единицу число участков дискретизации по осям  $x_1$  и  $x_2$ . Однако можно циклически менять характер аппроксимации, например на  $i$ -м временном шаге для производных по  $x_1$  использовать неявную, для производных по  $x_2$  — явную формулы, а на  $(i+1)$ -м шаге, наоборот, для производных по  $x_1$  — явную, для производных по  $x_2$  — неявную формулы. Тогда на каждом шаге по времени нужно решать  $q$  систем уравнений порядка  $p$  или  $p$  систем уравнений порядка  $q$ .

**Учет латентности фрагментов.** Локальные погрешности интегрирования зависят от значения шага интегрирования  $h$  и от характера переходных процессов. Если фазовые переменные претерпевают быстрые изменения, то погрешность не выше заданной обеспечивается при малых  $h$ . Если же фазовые переменные меняются медленно, то значения  $h$  при тех же погрешностях могут быть существенно больше. В сложных схемах ЭВА, как правило, большинство фрагментов в любой момент времени относится к неактивным (латентным), т. е. к таким, в которых не происходит изменений фазовых переменных, причем отрезки латентности  $T_{\text{лат}}$  могут быть довольно продолжительными. В латентных фрагментах допустимо увеличивать шаг интегрирования вплоть до значения  $T_{\text{лат}}$ , что эквивалентно исключению уравнений фрагментов из процесса интегрирования на период их латентности. Такое исключение выполняется в алгоритмах учета латентности, относящихся к алгоритмам *событийного моделирования*. Основу этих алгоритмов составляет проверка условий латентности. Примером таких условий может служить

$$\|\Delta V\| < \varepsilon, \quad (5.18)$$

где  $\Delta V$  — вектор изменений фазовых переменных;  $\varepsilon$  — малая положительная константа.

При проверке (5.18) у нелатентных фрагментов в вектор  $\Delta V$  входят изменения как внешних, так и внутренних переменных фрагмента, и если условие (5.18) выполняется на протяжении нескольких шагов интегрирования подряд, то фрагмент включается в группу латентных. При проверке (5.18) у латентных фрагментов вектор  $\Delta V$  состоит из изме-

нений только внешних фазовых переменных, происшедших за время с начала латентности, и фрагмент исключается из числа латентных, если условие (5.18) нарушается.

Следует отметить, что в таких методах, как РФС, латентность учитывается естественным образом, в них не требуется специальных проверок статуса латентности фрагментов.

**Адаптивное моделирование.** *Адаптивное моделирование* — метод автоматического выбора подходящих моделей для фрагментов в процессе анализа сложной системы. Метод адаптивного моделирования составляют способы решения следующих основных вопросов:

1. Фрагментация.
2. Критерий смены моделей.
3. Определение начальных значений переменных во включаемых моделях.

Рассмотрим один из возможных вариантов метода адаптивного моделирования. В нем используются модели трех уровней сложности:  $M1$  — простейшая макро модель фрагмента;  $M2$  — промежуточная по сложности макро модель, в которой блоки  $I$  и  $O$  (см. рис. 5.2) адекватно отражают входные и выходные цепи схемы,  $M3$  — полная модель фрагмента. Модели  $M1$  и  $M2$  имеют блочную структуру (см. рис. 5.2). Фрагментацию используют двоякую: во-первых, функциональную фрагментацию, выделяющую фрагменты, которым соответствуют сменяемые модели; во-вторых, диакоптическую фрагментацию, выделяющую отдельно интегрируемые фрагменты. Диакоптическая фрагментация выполняется с перекрытием, зону перекрытия составляют блоки  $I$  макро моделей. При функциональной фрагментации нужно стремиться к получению максимально возможного числа типовых фрагментов, под которыми понимаются фрагменты, для которых заранее разработаны и включены в библиотеку макро модели типов  $M1$  и  $M2$  вместе с рассчитанными для макро моделей типа  $M1$  областями адекватности (ОА). Диакоптический фрагмент есть функциональный фрагмент вместе с зоной перекрытия, которой соответствует блок  $I$  из модели схемы-нагрузки.

Предполагается, что анализ начинается при представлении всех фрагментов моделями  $M1$ . При дальнейшем анализе возможны замены моделей  $M1$  моделями  $M3$ . Такие замены выполняются для тех фрагментов, в которых происходит нарушение условий адекватности, обнаруживаемое при выходе вектора внешних переменных фрагмента за пределы ОА. Одновременно с включением полной модели  $M3$



для некоторого фрагмента А происходит замена блоков  $O$  в моделях схем, являющихся источниками сигналов для фрагмента А, и замена блоков  $I$  в моделях схем, являющихся нагрузками для фрагмента А. Эта замена — включение блоков из моделей  $M_2$  вместо блоков из моделей  $M_1$ .

Включение новой модели сопровождается появлением новых фазовых переменных. Очевидно, что для продолжения анализа должны быть определены значения этих переменных в момент  $t_{\text{вкл}}$  включения модели. Расчет этих значений целесообразно выполнять повторным интегрированием подсистемы дифференциальных уравнений включаемой модели на интервале  $[0, t_{\text{вкл}}]$ .

## 5.5. МЕТОДЫ АНАЛИЗА ЛОГИЧЕСКИХ И ФУНКЦИОНАЛЬНЫХ СХЕМ ЭЛЕКТРОННОЙ ВЫЧИСЛИТЕЛЬНОЙ АППАРАТУРЫ

**Методы решения логических уравнений.** Анализ переходных процессов в логических схемах выполняют с помощью асинхронных моделей (4.56), т. е. на основе *асинхронного моделирования*. К началу очередного такта  $t_i$  известны значения векторов внутренних  $V_i = (v_{1i}, v_{2i}, \dots, v_{ni})$  и входных  $U_i$  переменных. Подставляя  $V_i$  и  $U_i$  в правую часть выражений (4.57), получаем новые значения  $v'_{ki}$ , которые примут внутренние переменные в моменты времени  $t_i + \tau_k$ , где  $\tau_k$  — внутренняя задержка распространения сигнала  $v_k$  в соответствующем элементе схемы. Далее переходим к следующему такту, в котором вычисления по (4.57) повторяются со значениями векторов  $V$  и  $U$ , соответствующими новому моменту времени  $t_{i+1} = t_i + 1$  (напомним, что время измеряется в количестве тактов). Асинхронное моделирование называют *потактовым*.

Если длительность такта превышает задержку в некоторых элементах, т. е. для некоторых  $k$  имеем  $\tau_k \approx 0$ , то в модели (4.57) последовательностной схемы появляются отдельные неявные относительно  $v_k$  выражения, а это приводит к необходимости решать подсистемы логических уравнений в пределах каждого такта. Если задержки не учитывать во всех элементах, то имеем синхронную модель (4.56), с помощью которой анализируются установившиеся состояния в схеме и могут определяться статические и динамические риски сбоя. Синхронная модель — это система логических уравнений

$$V = F(V, U). \quad (5.19)$$

Для синхронного моделирования (решения систем логических уравнений) используются итерационные методы простой итерации и Зейделя.

Алгоритм метода простой итерации при решении (5.19) совпадает с алгоритмом асинхронного моделирования при  $\tau_k = 1$ . На первой итерации (такте) выбирается начальное приближение  $V_0$  и подставляется в правую часть (5.19), при этом определяется новое приближение  $V_1$ . На второй итерации рассчитывается  $V_2$  при подстановке  $V_1$  в правую часть (5.19) и т. д. Условием прекращения итераций является или совпадение результатов двух последних итераций  $V_i = V_{i-1}$ , где  $i$  — номер итерации, или выполнение равенства  $i = i_{\max}$ , где  $i_{\max}$  — предельно допустимое число итераций. В последнем случае делается заключение о несходимости вычислительного процесса из-за генерирования колебаний в моделируемой схеме.

Итерации по методу Зейделя отличаются от простых итераций тем, что новое значение  $k$ -го элемента вектора  $V_i$  сразу же после его вычисления на  $i$ -й итерации заменяет старое значение и используется для вычисления новых значений следующих элементов вектора  $V_i$  на той же  $i$ -й итерации, т. е.

$$v_{ki} = F(v_{1i}, v_{2i}, \dots, v_{k-1,i}, v_{k,i-1}, \dots, v_{n,i-1}).$$

В правильно спроектированной комбинационной схеме каждому входному вектору  $U$  соответствует одно определенное решение  $V^*$  системы логических уравнений (5.19). Корень  $V^*$  может быть найден при произвольном начальном приближении как методом простой итерации, так и методом Зейделя. Однако в последовательностной схеме система уравнений (5.19) имеет несколько решений, соответствующих различным устойчивым состояниям схемы. Обычно требуется найти то состояние, в которое переходит схема из заданного исходного состояния при заданном входном воздействии. Пусть заданному исходному состоянию соответствует значение  $V_{\text{исх}}$  вектора  $V$ . Тогда для решения сформулированной задачи рекомендуется выбрать начальное приближение  $V_0 = V_{\text{исх}}$  и, кроме того, в методе Зейделя необходимо привести в соответствие последовательности вычислений по уравнениям модели и прохождения сигналов в схеме. Последнее условие выполняется с помощью алгоритмов ранжирования, что способствует определению искомого корня за малое число итераций. Например, для комбинационной схемы решение получается за одну итерацию.

**Ранжирование логических схем и уравнений.** Алгоритмы ранжирования комбинационных схем аналогичны алгоритмам ранжирования электронных схем в методе однонаправленных моделей. Ранг 0 присваивается входным цепям, по которым поступают сигналы на схему извне. Ранг  $r$  присваивается элементам, все входы которых ранжированы и старший из рангов равен  $r-1$ . Ранг  $r$  присваивается также выходам элементов ранга  $r$ . После этого логические уравнения упорядочиваются по значениям рангов соответствующих элементов схемы.

В последовательностных схемах ранжирование может быть выполнено таким же образом после преобразования схемы в комбинационную с помощью разрывов обратных связей. Ранжирование последовательностей схемы начинается так же, как и комбинационной, но проранжировать таким образом удастся только часть элементов. Далее нужно выполнить разрывы контуров обратной связи. Контур обратной связи ищутся среди элементов, которые проранжировать не удалось. Выбирается любой из этих элементов, например  $A$ , к нему добавляется элемент  $B$ , выход которого связан с не имеющим ранга входом элемента  $A$ . Очевидно, что такой элемент  $B$  имеется, иначе элемент  $A$  был бы проранжирован. Далее таким же образом присоединяется элемент  $C$  к элементу  $B$  и т. д., пока в цепочке из элементов  $A, B, C, \dots$  не встретится повторно какой-либо из элементов, уже включенных в цепочку. Эта ситуация означает обнаружение контура обратной связи. Далее одна из связей в контуре обратной связи обрывается. Входы элементов от разорванных связей, называемые псевдовходами, получают ранг 0. После этого ранжирование возобновляется. Операции выделения контуров и ранжирования повторяются до тех пор, пока в схеме не останется непроранжированных элементов.

**Повышение эффективности моделирования логических и функциональных схем.** Для повышения эффективности решения уравнений методом Зейделя целесообразно использовать *диакоптический подход*, в рамках которого итерации выполняются отдельно по фрагментам логической схемы. Введем следующие понятия: составной элемент — множество контуров обратной связи, имеющих попарно общие связи; фрагмент логической схемы — составной элемент или комбинационная схема, состоящая из взаимосвязанных логических элементов, не вошедших в составные элементы.

Очевидно, что после обнаружения контуров обратной

связи нетрудно найти составные элементы и, следовательно, представить моделируемую схему, состоящую из фрагментов. В такой схеме нет межфрагментных обратных связей и, следовательно, система логических уравнений может быть решена путем однократного обращения к проранжированным подсистемам уравнений. При этом решение подсистем комбинационных фрагментов будет выполнено за одну итерацию Зейделя, а решение подсистем составных элементов — за число итераций, определяемое особенностями данного составного элемента и способом разрыва в нем обратных связей. Это число итераций можно уменьшить, если осуществить разрывы всех контуров обратной связи в минимальном количестве межэлементных связей. Пофрагментное выполнение итераций существенно сокращает общие затраты машинного времени.

Другим способом повышения эффективности является *параллельное моделирование*, основанное на том, что для представления логической переменной достаточно  $k$  разрядов, где  $k=1$  в двузначном алфавите и  $k=2$  в трехзначном. Тогда моделирование одной и той же схемы можно выполнять одновременно для  $m=s/k$  различных наборов входных сигналов, где  $s$  — количество разрядов в разрядной сетке ЭВМ. Подобное параллельное моделирование эффективно используется при синтезе тестов для проверки логических схем, где требуется определить реакцию схемы на большое количество входных тестовых наборов.

Наиболее общим направлением повышения эффективности математического обеспечения как синхронного, так и асинхронного моделирования является *учет событийности*. При анализе логических и функциональных схем *событием* называют изменение состояния любого элемента или, что то же самое, изменение значения любой переменной состояния. В процессе событийного моделирования вычисления производят только по уравнениям активных элементов, т. е. таких элементов, на входах которых на данном такте или итерации произошли события.

Рассмотрим алгоритм асинхронного событийного моделирования. В алгоритме используются списки текущих и будущих событий. Все события привязаны к моментам дискретного модельного времени. Ссылки на события, происходящие в текущий момент, находятся в списке текущих событий, а ссылки на те события, наступление которых можно предвидеть, помещаются в список будущих событий. Моделирование текущих событий означает обращение к моде-

лям соответствующих элементов, после чего становятся известными изменяющиеся выходные переменные и задержки в их изменении по отношению к текущему моменту модельного времени. Эти изменения суть события, и сведения о них заносятся в список будущих событий. После того как список текущих событий полностью отработан, модельное время увеличивается и соответствующая часть списка будущих событий становится списком текущих событий. Так как в сложных схемах на каждом такте активными являются не более нескольких процентов элементов, то учет событийности позволяет существенно уменьшить затраты машинного времени.

В маршрутах проектирования БИС и СБИС к числу основных проектных процедур относятся верификация логических и функциональных схем, синтез и анализ тестов. В этих процедурах требуется многократное выполнение моделирования логических схем. Однако высокая размерность задач логического моделирования (СБИС насчитывают десятки—сотни тысяч вентиляей) существенно ограничивает возможности многовариантного анализа. Так, современные программы анализа логических схем на универсальных ЭВМ могут обеспечить скорость моделирования приблизительно  $10^3$  вентиляей в секунду (т. е. на анализ реакции схемы из  $10^3$  вентиляей на один набор входных воздействий затрачивается 1 с машинного времени), что значительно ниже требуемого уровня. Преодоление затруднений, обусловливаемых чрезмерной трудоемкостью вычислений, происходит в двух направлениях. Первое из них основано на использовании общих положений блочно-иерархического подхода и выражается в переходе к представлениям подуровня регистровых передач, рассмотренным в § 4.7. Второе направление основано на применении специализированных вычислительных средств логического моделирования, называемых спецпроцессорами или машинами логического моделирования (МЛМ). Важно отметить, что появление СБИС не только порождает потребности в таких спецпроцессорах, но и обуславливает возможности их создания с приемлемыми затратами. Разработанные к настоящему времени МЛМ функционируют совместно с универсальными ЭВМ и обеспечивают скорость моделирования  $10^6$ — $10^9$  вентиляей в секунду.

**Логико-электрическое моделирование.** Логико-электрическое моделирование представляет собой разновидность многоуровневого моделирования, при котором в анализиру-

емой схеме одновременно используются модели и методы, относящиеся как к схемотехническому, так и к функционально-логическому иерархическому уровню.

В анализируемой схеме выделяются подсхемы, подлежащие анализу с помощью логических и электрических моделей. Сопряжение моделей подсхем осуществляется с помощью специальных переходных моделей элементов и алгоритмов синхронизации событий в логической и электрической частях. Переходные модели служат для отображения процессов в элементах с преобразованием аналоговых переменных в логические и наоборот.

Логико-электрическое моделирование является незаменимым способом анализа сложных цифроаналоговых схем, поскольку позволяет резко сократить размерность задач, благодаря использованию логических моделей для цифровой части при сохранении необходимой точности анализа, благодаря использованию электрических моделей для аналоговой части схемы. Логико-электрическое моделирование позволяет повысить эффективность решения также ряда задач проектирования цифровой аппаратуры, если в анализируемой логической схеме имеются отдельные фрагменты, требующие для своего адекватного представления моделей схемотехнического уровня.

## 5.6. МЕТОДЫ МНОГОВАРИАНТНОГО АНАЛИЗА

**Анализ чувствительности.** Анализ чувствительности входит составной частью в алгоритмы решения многих задач, в частности в алгоритмы оптимизации градиентными методами. Для анализа чувствительности задаются ММ объекта и вектор тех внутренних и внешних параметров  $X$ , влияние которых на вектор выходных параметров  $Y$  требуется определить.

В большинстве случаев анализ чувствительности выполняют на основе численного дифференцирования, при котором поочередно задают приращения  $\Delta x_i$  элементам вектора  $X$  и определяют получающиеся при этом изменения  $\Delta y_j$  выходных параметров. Тогда абсолютный коэффициент влияния (коэффициент чувствительности)  $i$ -го элемента  $x_i$  вектора  $X$  на  $j$ -й выходной параметр  $y_j$  определяется по формуле

$$a_{ji} = \partial y_j / \partial x_i \approx \Delta y_j / \Delta x_i;$$

относительный коэффициент влияния

$$b_{ji} = a_{ji} x_{i\text{ном}} / y_{j\text{ном}},$$

где  $x_{iном}$  и  $y_{iном}$  — номинальные значения параметров  $x_i$  и  $y_i$  соответственно.

Такой метод анализа чувствительности называют *методом приращений*. Если  $n$  — размерность вектора  $\mathbf{X}$ , то в методе приращений требуется  $n+1$  раз выполнить одновариантный анализ. Сравнительно большие трудоемкость и погрешности вычислений, присущие численному дифференцированию, относятся к недостаткам этого метода, а универсальность метода — к его преимуществам.

Повысить точность вычислений можно, если, определяя  $a_{ji}$ , производить одновариантный анализ при значениях параметра  $x_i$ , равных  $x_{iном} + \Delta x_i$  и  $x_{iном} - \Delta x_i$  (остальные внутренние параметры при этом сохраняют номинальные значения). Тогда  $a_{ji}$  есть отношение разности значений  $y_j$ , полученных в этих двух вариантах, к  $2\Delta x_i$ . Однако в этом методе увеличивается трудоемкость — нужно  $2n$  раз выполнить одновариантный анализ.

Применяют также методы, имеющие менее универсальный характер, например методы анализа чувствительности функционалов зависимостей  $\mathbf{V}(t)$ , получающихся при интегрировании систем ОДУ. В этих методах или сокращается число вариантов интегрирования уравнений, или упрощается система интегрируемых уравнений.

**Статистический анализ.** Статистический анализ имеет целью получение информации о распределении вектора выходных параметров  $\mathbf{Y}$  при заданном законе распределения случайного вектора  $\mathbf{X}$  внутренних параметров объекта.

Основным методом статистического анализа в САПР является метод *статистических испытаний (метод Монте-Карло)*. Каждое  $k$ -е статистическое испытание заключается в присвоении элементам  $x_i$  вектора  $\mathbf{X}$  случайных значений  $x_{ik}$  и расчете вектора выходных параметров  $\mathbf{Y}_k$  с помощью одновариантного анализа. После выполнения запланированного числа  $N$  статистических испытаний их результаты  $\mathbf{Y}_k$  обрабатываются с целью оценки числовых характеристик распределений выходных параметров.

В основе алгоритма задания случайных значений параметрам лежит формула

$$\xi = \int_{-\infty}^{x_{ik}} p(x_i) dx_i,$$

с помощью которой случайное значение  $\xi$  величины, равномерно распределенной в интервале  $[0, 1]$ , преобразуется в

случайное значение  $x_{ik}$  величины  $x_i$ , имеющей плотность распределения  $p(x_i)$ . Для выработки значений  $\xi$  используют стандартные подпрограммы, имеющиеся в программном обеспечении любой универсальной ЭВМ. Выработанное значение  $\xi$  интерпретируется как значение  $F(x_{ik})$  функции распределения величины  $x_i$ , так как

$$F(x_{ik}) = \int_{-\infty}^{x_{ik}} p(x_i) dx_i.$$

Функция распределения  $F(x_i)$  монотонная, обычно представлена в памяти ЭВМ в табличной форме, и, следовательно, преобразование  $\xi$  в  $x_{ik}$  сводится к поиску нужного интервала таблицы и определению результата с помощью интерполяции.

Вместо значений  $\xi$  равномерно распределенной случайной величины можно использовать значения  $u_k$  нормированного нормально распределенного вектора  $\mathbf{U} = (u_1, u_2, \dots, u_n)$  и в алгоритме осуществлять преобразование  $u_{ik}$  в  $x_{ik}$ . Но элементы вектора  $\mathbf{U}$  некоррелированы, поэтому если требуется выработка значений коррелированных случайных величин  $x_i$ , то вводятся промежуточный вектор коррелированных нормальных величин  $\mathbf{Z}$  и матрица  $\mathbf{A}$  преобразования  $\mathbf{U}$  в  $\mathbf{Z}$ . Тогда алгоритм задания случайных значений параметрам  $x_i$  сначала вырабатывает  $n$  случайных значений  $u_{ik}$ , затем преобразует их в вектор  $\mathbf{Z}_k = \mathbf{A}\mathbf{U}_k$  и далее значения этого вектора — в искомые значения  $x_{ik}$  коррелированных параметров элементов.

Матрица преобразования  $\mathbf{A}$ , так же как и таблица преобразования  $\mathbf{Z}$  (или  $\mathbf{U}$ ) в  $\mathbf{X}$ , определяется на основе обработки результатов предварительно выполненных измерений параметров на партии тестовых образцов либо изделий данного или аналогичного типа.

Результаты статистических испытаний  $\mathbf{Y}_k$  используются для построения гистограмм, подсчета математических ожиданий и дисперсий выходных параметров. Можно рассчитать также коэффициенты корреляции между выходными  $y_j$  и внутренними  $x_i$  параметрами, которые используются для определения коэффициентов регрессии  $y_j$  на  $x_i$ . Поскольку относительные коэффициенты регрессии являются аналогами коэффициентов влияния  $x_i$  на  $y_j$ , регрессионный анализ, совмещаемый со статистическим анализом, следует рассматривать как возможный подход к анализу чувствительности.



Точность и трудоемкость статистических испытаний зависят от их числа  $N$ . Для получения большинства интересных разработчика результатов статистического анализа с приемлемой погрешностью требуется выбирать  $N=50-200$ . Однако получение некоторых результатов (таких, как вероятность выхода годных изделий при значениях этой вероятности, близких к единице или нулю) с приемлемой точностью требует значительно большего числа испытаний. Отсюда следует вывод о значительной трудоемкости статистического анализа. Именно по этой причине статистический анализ проводят лишь на заключительных итерациях процесса проектирования изделий.

**Разработка тестов цифровой аппаратуры.** Проектирование цифровой аппаратуры включает в себя разработку тестов, с помощью которых проверяется отсутствие неисправностей в изготавливаемой или эксплуатируемой аппаратуре.

Назовем тестируемый объект блоком. Входное воздействие обозначим вектором  $X=(x_1, x_2, \dots, x_n)$ , а выходную реакцию — вектором  $Y=(y_1, y_2, \dots, y_m)$ , где  $x_i$  — булева переменная на  $i$ -м входе;  $y_j$  — то же на  $j$ -м выходе. Определенному значению  $X_k$  вектора  $X$  в исправном блоке соответствует значение  $Y_k$  вектора  $Y$ . Пару  $\{X_k, Y_k\}$  называют элементарной проверкой.

С помощью одной элементарной проверки нельзя выявить все неисправности даже ограниченного класса и тем более нельзя локализовать неисправность. Поэтому тест представляет собой множество элементарных проверок. Число элементарных проверок в тесте называют его длиной. Тест, предназначенный только для установления факта неисправности блока, называют *контролирующим (проверяющим)*, а тест, с помощью которого дополнительно устанавливается место неисправности в блоке, — *диагностическим*. Тест, который выявляет все неисправности заданного класса, называют полным, а тест, из которого нельзя исключить ни одну элементарную проверку без изменения его полностью, — *неизбыточным*.

Очевидно, что тест, в который включены всевозможные комбинации значений входных сигналов, является полным. Однако громадное число таких комбинаций, равное  $2^n$  в комбинационном блоке и  $2^{n+q}$  в последовательностном, делает нереальным применение подобных полных тестов (здесь  $q$  — число элементов памяти в блоке). Поэтому возникают задачи синтеза и анализа тестов. При синтезе гене-

рируются входные наборы  $X_k$ , а при анализе устанавливаются характеристики теста.

Обычно тесты рассчитывают на выявление только оди-ночных константных неисправностей. Это класс устойчивых неисправностей, возникающих поодиночке в единственном элементе блока. К ним относятся неисправности «констант-ный нуль», «константная единица», связанные с наличием постоянного низкого или высокого уровня напряжения, и «инверсная неисправность», связанная с появлением не-предусмотренного инвертирования сигнала.

Для синтеза тестов применяют вероятностные и детер-минированные методы. В вероятностных методах наборы генерируются с помощью датчиков случайных чисел. Ос-новные затраты машинного времени приходятся при этом на анализ проверяющих возможностей генерируемых набo-ров. Анализ каждого набора состоит в расчете реакции на воздействие  $X_k$  как исправного блока, так и всех его воз-можных разновидностей. Если блок состоит из  $N$  элементов, то имеем  $3N$  таких разновидностей и общее число вариан-тов моделирования блока окажется пропорциональным про-изведению  $sN$ , где  $s$  — число проверяемых входных наборов. Практика показывает, что при заданной полноте теста  $s$  зависит от  $N$  и в результате затраты машинного времени  $T_m$  оказываются пропорциональными  $N^2$ , где  $\alpha = 2 \div 3$ .

Большие значения  $T_m$  обуславливают применение для анализа тестов наиболее экономичных методов моделирова-ния логических и функциональных схем. Обычно используют параллельное синхронное трехзначное моделирование. Трехзначный алфавит целесообразен для отбраковки вход-ных векторов  $X_k$ , приводящих к состязаниям сигналов в бло-ке, из-за которых результаты применения теста могут стать неопределенными.

В детерминированных методах синтеза тестов для каж-дой неисправности из заданного списка подбирают свой входной набор. Одним из наиболее распространенных алго-ритмов, реализующих детерминированный подход к синтезу тестов, является алгоритм *Рота*. В соответствии с этим ал-горитмом для очередной неисправности, связанной с эле-ментом  $\mathcal{E}$  тестируемой схемы, подбираются входные для  $\mathcal{E}$  воздействия, такие что выходы исправного и неисправного элементов  $\mathcal{E}$  будут иметь неодинаковые значения. После этого ищется путь транспортировки неисправности (т. е. путь передачи информации о наличии неисправности) к вы-ходам схемы, которые доступны для наблюдения. Поиск

пути заключается в установлении цепочки элементов между элементом Э и выходом схемы и в определении таких значений входов элементов цепочки, при которых неисправность транспортируется к выходу. Очевидно, что не любая цепочка может служить путем транспортировки, поскольку требуемые для транспортировки значения входов могут оказаться противоречивыми. Поэтому в алгоритме Рота осуществляется последовательное наращивание цепочки с проверкой непротиворечивости условий транспортировки для ранее и вновь включаемых элементов. После установления пути требуемые значения входов у элементов цепочки пересчитываются в значения тех входов тестируемой схемы, которые доступны для управления. Эти значения входов и представляют собой очередной искомый набор теста.

В программах синтеза тестов обычно используют комбинирование вероятностных и детерминированных методов. С помощью первых генерируется и анализируется некоторое заранее заданное число наборов. Если полнота теста после этого недостаточна, то выбирают неисправности, не выявляемые уже полученными наборами, и для этих неисправностей ищут входные наборы детерминированным методом.

### Контрольные вопросы и задания

1. Что понимается под эффективностью метода анализа? Назовите основные направления повышения эффективности.
2. Что такое «многоуровневое моделирование»?
3. Подсчитайте разреженность для матрицы табл. 4.3.
4. Можно ли применять метод прогонки к решению системы уравнений (4.52)? Одинаковы или нет исходная и итоговая разреженности матрицы Якоби в этой системе?
5. Поясните причины возникновения неустойчивости вычислений при интегрировании обыкновенных дифференциальных уравнений.
6. Проверьте наличие сходимости при решении уравнения  $x-10=0$  методом простой итерации с  $h=0,5$  и  $h=4$ , задавшись начальным приближением  $x_0=0$ .
7. Почему интегрирование обыкновенных дифференциальных уравнений при анализе процессов в проектируемых объектах нужно выполнять с переменным шагом?
8. В чем заключается сущность метода ФДН?
9. С какой целью применяют комбинирование методов и алгоритмов анализа?
10. Почему метод Эйлера нецелесообразно применять для анализа процессов в слабодемпфированных системах?

11. Решите систему логических уравнений  $p = \overline{m \wedge g}$ ,  $h = \overline{f \wedge g}$ ,  $g = \overline{c \wedge d}$ ,  $f = \overline{a \wedge b}$ ,  $m = \overline{h \wedge l}$  методом простой итерации и методом Зейделя. Выполните ранжирование системы уравнений и сравните число итераций для решений различными методами. Задайтесь начальным приближением  $f=1$ ,  $g=1$ ,  $h=0$ ,  $m=1$ ,  $p=0$  и значениями входных переменных  $a=1$ ,  $b=1$ ,  $c=0$ ,  $d=1$ ,  $l=1$ .

12. На основе предыдущего примера покажите возможности снижения трудоемкости вычислений с помощью событийного моделирования.

13. С какой целью применяют логико-электрическое моделирование?

14. В каких проектных процедурах применяют результаты анализа чувствительности?

15. Обоснуйте выбор для анализа тестов параллельного синхронного трехзначного моделирования.

## ГЛАВА 6

### СИНТЕЗ ТЕХНИЧЕСКИХ ОБЪЕКТОВ В САПР

#### 6.1. ПОСТАНОВКА ЗАДАЧИ СИНТЕЗА

Понятие «синтез» технического объекта в широком смысле слова близко по содержанию к понятию «проектирование». Задача синтеза технического объекта состоит в том, чтобы по заданному функциональному назначению объекта или по закону его функционирования получить проектное решение в виде некоторого описания проектируемого объекта.

Синтез технических объектов нацелен на создание новых вариантов, а анализ используется для оценки этих вариантов, т. е. синтез и анализ выступают в процессе проектирования в диалектическом единстве.

При синтезе заранее заданы допустимый набор используемых элементов (электрорадиоэлементов при синтезе электронных схем, набор балок и блочных конструкций при проектировании строительных сооружений и т. д.), возможные правила их соединения между собой и способы определения по синтезированной структуре объекта функции, которую он реализует.

Под *структурой объекта* понимается набор составляющих его элементов и связей между ними. Структура определяет, как устроен объект проектирования (ЭВМ, автоматическая система, радиотехнический комплекс и т. д.), из каких физических частей он состоит и как эти части связаны друг с другом.

К объектам проектирования относятся структура и кон-

струкция, причем под *конструкцией объекта* понимается материализованная совокупность соединенных между собой элементов, выполняющих заданные функции. В частности, *конструкция ЭВМ* — это совокупность электрически и механически соединенных элементов, в которой реализуется электрическая схема данной машины.

На структуру и конструкцию любого проектируемого объекта всегда накладывается множество различных ограничений. При этом одна группа ограничений относится к методу решения задачи и охватывает такие вопросы, как наличие знаний, сроки и имеющиеся в распоряжении технические средства проектирования. Другая группа ограничений связана с требованиями ТЗ на параметры проектируемого объекта, с требованиями стандартов и технологии изготовления узлов и различных элементов объекта. Третья группа ограничений формируется физическими принципами реализации закона функционирования объекта и получения его предельно желаемых характеристик. Дополнительные ограничения накладываются способами и формами взаимодействия проектируемого объекта с внешней средой, а также методами организации взаимодействия человека с проектируемым объектом в процессе функционирования и эксплуатации.

При проектировании любого технического объекта выбирается или задается в ТЗ на проектирование критерий оптимальности. В зависимости от характера и назначения проектируемого объекта в качестве критерия оптимальности может быть принята его стоимость, КПД, потребляемая мощность или другой, более сложный критерий. Вопросы выбора критериев оптимальности рассматривались в § 1.3.

При проектировании технических объектов важное значение имеет определение оптимальных вариантов структур и конструкций машин и устройств, параметров схем, режимов работы технологического оборудования и т. д. Под *оптимальным* будем понимать такой вариант структуры или конструкции, параметры которой удовлетворяют всем системным, конструктивным, технологическим, электрическим и экономическим требованиям ТЗ, а критерий оптимальности, описывающий качество проектируемой структуры или конструкции, принимает наилучшее (минимальное или максимальное) значение.

Получение оптимальных решений при проектировании стало возможным и достижимым как по срокам проектирования, так и по стоимости реализации проектных процедур

только при автоматизированном проектировании, когда появляется возможность синтезировать и исследовать множество вариантов структур и конструкций, а также проводить количественное изучение проектируемых технических объектов, которые в прошлом изучались лишь качественно.

Примерами задач оптимального проектирования являются: определение структуры ЭВМ максимальной производительности при заданных массогабаритных ограничениях, надежности, потребляемой мощности и другом; расчет элементов конструкций летательного аппарата максимальной грузоподъемности при заданных мощности двигателя и ограничениях на другие параметры аппарата; определение конструктивных параметров электрических двигателей, оптимальных по критерию минимальной стоимости, и др.

Любому варианту проектируемого объекта соответствуют свои структура и конструкция. При автоматизированном проектировании для порождения множества альтернативных структур технического объекта, эквивалентных по функциональному назначению, но различных по тактико-техническим характеристикам, необходима разработка математической модели объекта, представляющей собой формальное описание проектируемого объекта на принятом уровне детализации.

При оптимальном проектировании необходимо обосновать критерий оптимальности проектируемого объекта и определить множество показателей  $\theta = (\theta_1, \dots, \theta_n)$ , на которые наложено множество ограничений  $V = (V_1, \dots, V_n)$ .

Для решения задачи синтеза технических объектов выделяют некоторую совокупность независимых переменных  $X = (x_1, \dots, x_m)$ , фиксация значений которых определяет один из вариантов объекта и его количественные характеристики, в том числе значение критерия оптимальности, а также показателей, принятых в качестве ограничений.

Переменные  $x_1, \dots, x_m$  называют *переменными проектирования* и в зависимости от физической природы объекта проектирования они имеют различную интерпретацию, в частности могут характеризовать количество узлов каждого типа в объекте, указывать на включение или на невключение того или иного узла в структуру объекта, представлять геометрические размеры изделия и т. д.

Критерий оптимальности  $F(X) = F(x_1, \dots, x_m)$  и показатели  $\theta_i(X) = \theta_i(x_1, \dots, x_m)$ , на значения которых наложены ограничения, являются функциями независимых переменных  $x_1, \dots, x_m$ .

В реальных задачах проектирования функция критерия оптимальности  $F(\mathbf{X})$ , функции ограничений  $\theta_i(\mathbf{X})$ , как правило, нелинейно зависят от значений множества переменных  $\mathbf{X}$ . Кроме того, во многих задачах проектирования на значения переменных накладываются условия целочисленности.

В формализованном виде задача синтеза технических объектов заключается в определении значений независимых переменных  $x_1, \dots, x_m$ , при которых критерий оптимальности проектируемого объекта

$$F(\mathbf{X}) = F(x_1, x_2, \dots, x_m) \quad (6.1)$$

принимает экстремальное (максимальное или минимальное) значение при условиях

$$\theta_i(x_1, \dots, x_m) \leq 0, \quad i = \overline{1, n}; \quad (6.2)$$

$$a_j \leq x_j \leq b_j, \quad j = \overline{1, m}. \quad (6.3)$$

Ограничения вида

$$\theta_i(x_1, \dots, x_m) \geq 0 \quad (6.4)$$

путем умножения левой части неравенства на минус единицу могут быть приведены к выражению (6.2).

Ограничения вида (6.2) и (6.4) в своей исходной постановке могут быть заданными в виде уравнений; в случае задания системы неравенств всегда можно перейти от неравенств к уравнениям

$$\theta_i(x_1, \dots, x_m, x_{m+i}) = 0 \quad (6.5)$$

путем введения дополнительных переменных  $x_{m+i}$ , причем для неравенств вида (6.2)  $x_{m+i} > 0$ , а для неравенств вида (6.4)  $x_{m+i} < 0$ . На ряд переменных может быть наложено условие целочисленности, т. е.

$$x_p \text{ — целые числа, } p = \overline{1, q}, q \leq m.$$

В задаче оптимизации, в которой ограничения имеют вид уравнений, количество ограничений  $n$  не может быть больше числа переменных  $m$ , т. е.  $n \leq m$ . Разность  $m - n$  определяет число степеней свободы в данной задаче. Только  $m - n$  переменных берутся произвольными, значения же остальных переменных определяются из системы ограничений. Если  $m = n$ , то число степеней свободы равно нулю и задача в этом случае является алгебраической. Оптимизация целевой функции  $F(\mathbf{X})$  при этом не требуется.

Задача оптимального проектирования в виде (6.1) — (6.5) представляет собой задачу математического программиро-

вания. При этом если целевая функция (6.1) и все ограничения линейны, то задачу оптимизации называют задачей линейного программирования, если же целевая функция или хотя бы одно ограничение нелинейны, то задача оптимизации является задачей нелинейного программирования.

Во многих задачах математического программирования некоторые переменные могут принимать лишь определенные дискретные значения (например, диаметр обмоточного провода, выбираемый из определенного сортамента, номиналы конденсаторов и т. д.) либо только целочисленные значения (например, число выпускаемых станков, самолетов и т. д.). В этом случае задача проектирования может быть сформулирована в терминах дискретного программирования.

При проектировании технических объектов с использованием моделей и методов математического программирования оказывается удобной геометрическая иллюстрация процесса получения оптимального решения. Рассмотрим геометрическую интерпретацию задачи математического программирования с линейной целевой функцией и с системой ограничений, образующих выпуклую оболочку области существования задачи оптимизации, т. е. пусть имеется система уравнений

$$\theta_i(x_1, \dots, x_m) = 0, \quad i = \overline{1, m}, \quad (6.6)$$

$$a_j \leq x_j \leq b_j, \quad j = \overline{1, n}, \quad (6.7)$$

и целевая функция вида

$$F(\mathbf{X}) = \sum_{j=1}^n c_j x_j. \quad (6.8)$$

Требуется найти значения переменных, удовлетворяющих ограничениям (6.6) и (6.7) и обращающих в минимум целевую функцию (6.8).

В данной задаче  $m-n$  переменных могут быть выбраны произвольно, в частности взяты равными нулю. Это дает возможность проиллюстрировать задачу геометрически в  $m-n$ -мерном пространстве  $E^{m-n}$ . В этом пространстве каждой точке  $\mathbf{X}^{(r)}$  соответствует совокупность чисел  $x_1^r, \dots, x_{m-n}^r$ , равных проекции вектора, проведенного из начала координат в точку  $\mathbf{X}^{(r)}$ , на координатные оси пространства  $E^{m-n}$ .

Функция  $F(x_1, \dots, x_m)$  в каждой точке пространства имеет определенное значение, следовательно, пространство  $E^{m-n}$  является скалярным полем критерия оптимальности  $F(\mathbf{X})$  и функций ограничений  $\theta_i(\mathbf{X})$ . Функциям ограничений (6.6) соответствуют граничные гиперповерхности (в частном случае — гиперплоскости). Ограничениям (6.7) соответствуют гиперплоскости, выделяющие в пространстве определенную пространственную область. Если ограничения (6.6) и (6.7) представляют собой выпуклую область, то решения задачи оптимизации будут со-



ответствовать такой точке этой области со скалярным полем критерия  $F(\mathbf{X})$ , в которой критерий  $F(\mathbf{X})$  принимает минимальное значение. Поясним вышесказанное на примере.

**Пример 6.1.** Имеется следующая задача нелинейного программирования: минимизировать целевую функцию

$$F(\mathbf{X}) = x_1 - x_2 \quad (6.9)$$

при ограничениях

$$x_1^2 + x_2^2 \leq 4; \quad -2x_1 + x_2 \leq 1; \quad x_1 - 2x_2 \leq 1. \quad (6.10)$$

Введя дополнительные переменные  $x_3, x_4$  и  $x_5$ , (6.10) превратим в систему уравнений, которую разрешим относительно этих введенных переменных. При этом получим

$$\begin{aligned} x_2 &= 4 - x_1^2 - x_3^2; \quad x_3 \geq 0; \\ x_4 &= 1 + 2x_1 - x_2; \quad x_4 \geq 0; \\ x_5 &= 1 - x_1 + 2x_2; \quad x_5 \geq 0. \end{aligned} \quad (6.11)$$

В данной задаче число переменных  $m=5$ , число уравнений  $n=3$ , тогда  $m-n=2$ , что дает возможность дать геометрическую интерпретацию задачи в пространстве  $E^2$ , т. е. на плоскости.

Поскольку все переменные должны быть положительными:

$$x_j \geq 0, \quad j = \overline{1,5}, \quad (6.12)$$

каждое из неравенств (6.12) определяет некоторую допустимую область в пространстве  $E^2$ . Так, неравенство  $x_1 \geq 0$  определяет верхнюю полуплоскость, неравенство  $x_4 \geq 0$  — полуплоскость, лежащую по одну сторону от прямой  $2x_1 - x_2 + 1 = 0$ , а именно ту, которая содержит начало координат. Область, соответствующая  $x_4 < 0$ , является запрещенной, и ее удобно отметить (на рис. 6.1, а она отмечена штриховкой).

Подобные построения для всех  $x_i$  приведены на рис. 6.1, б. Областью существования задачи является область  $OABCD$ . В рассматриваемом примере функция (6.9) принимает минимальное значение на границе допустимой области в точке  $B$ .

Отметим ряд особенностей задачи нелинейного программирования с нелинейной целевой функцией. В общем случае заранее нельзя сказать о расположении точки, в которой функция  $F(\mathbf{X})$  принимает максимальное или минимальное значение. Эта точка может находиться как на границе допустимой области, так и внутри нее. Функция  $F(\mathbf{X})$  может достигнуть экстремального значения как в одной точке, так и на некотором множестве (гиперлинии или гиперповерхности).

Другая особенность задачи математического программирования состоит в том, что в общем случае нелинейная це-

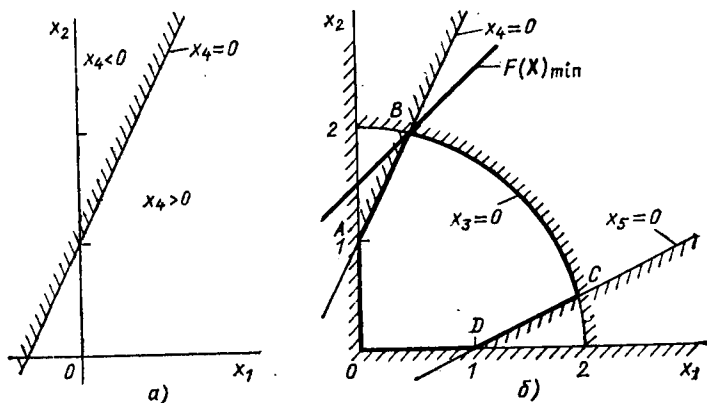


Рис. 6.1. Допустимая и запрещенная полуплоскости (а) и область существования задачи нелинейного программирования (б)

левая функция  $F(\mathbf{X})$  может иметь несколько локальных экстремумов в допустимой области, включая ее границу. Поэтому при оптимальном проектировании структур и конструкций технических объектов возникает проблема определения глобального экстремума задачи оптимизации, что в практических задачах проектирования нередко вызывает значительные трудности при вычислениях.

В общем виде задача нелинейного программирования пока не имеет строгого математического решения. Однако в связи с тем что данный класс задач довольно часто встречается в практических задачах проектирования, разработано большое число методов и эвристических алгоритмов решения конкретных задач нелинейного программирования.

## 6.2. СТРУКТУРНЫЙ СИНТЕЗ И ПАРАМЕТРИЧЕСКАЯ ОПТИМИЗАЦИЯ

Задача синтеза технического объекта включает в себя создание структуры проектируемого объекта и расчет его параметров. Эти две части синтеза называют *структурным и параметрическим синтезом*.

**Структурный синтез.** Задача структурного синтеза заключается в поиске оптимальной или рациональной структуры (схемы) технического объекта для реализации заданных функций в рамках выбранного принципа действия. Результаты структурного синтеза могут быть представлены в виде перечня элементов вместе с таблицей соединений, схе-

мы расположения элементов с указаниями их типов, схемы алгоритмов и т. п.

Задачи структурного синтеза относят к задачам проектирования, наиболее сложным с точки зрения возможностей формализации. Сложность формализованного синтеза структур заключается прежде всего в наличии большого числа факторов, влияющих на разновидности, свойства и параметры синтезируемых структур, а также в трудностях решения задач оптимизации большой размерности при высокой степени детализации описания синтезируемых объектов.

Так, при структурном синтезе специализированных ЭВМ и микропроцессорных систем возникают трудности решения ряда специфичных проблем, связанных с согласованием структур и алгоритмов функционирования ЭВМ и систем с характеристиками решаемых задач. Результатами структурного синтеза ЭВМ являются номенклатура входящих в состав ЭВМ блоков, число блоков каждого типа, топология информационных и управляющих взаимосвязей между блоками, а также расписание функционирования каждого блока при организации вычислительного процесса.

Большая размерность задач проектирования сложных технических систем и объектов делает целесообразным блочно-иерархический подход, при котором процесс проектирования разбивается на взаимосвязанные иерархические уровни. Структурный синтез составляет существенную часть процесса проектирования и также организуется по блочно-иерархическому принципу. Это означает, что синтезируется не вся сложная система целиком, а на каждом уровне в соответствии с выбранным способом декомпозиции синтезируются определенные функциональные блоки с соответствующим уровнем детализации. Существуют различные способы классификации задач структурного синтеза. Так, в частности, в зависимости от стадии проектирования различают следующие процедуры структурного синтеза: выбор основных принципов функционирования проектируемой системы, выбор технического решения в рамках заданных принципов функционирования, выпуск технической документации. В зависимости от типа синтезируемых структур различают задачи одномерного, схемного и геометрического синтеза. В зависимости от возможностей формализации различают задачи, в которых возможен полный перебор известных решений, задачи, которые не могут быть решены путем полного перебора за приемлемое время, задачи по-

иска вариантов структур в счетном множестве допустимых вариантов и задачи синтеза, решение которых является проблематичным.

В самом общем случае для решения задачи необходим перебор вариантов, и сокращение перебора является актуальной проблемой. Иерархический подход уменьшает число вариантов на каждом уровне и делает решение задачи определения оптимальной структуры технического объекта реальным.

Отметим существенное различие между задачами синтеза оптимальных структур и задачами анализа качества структур технических объектов. В анализе необходимо убедиться, что решение существует, а численные методы анализа устойчивы. При структурном синтезе не гарантировано даже существование номинальной структуры, удовлетворяющей всем требованиям ТЗ на проектируемый объект. Существующие и разрабатываемые ММ синтезируемых технических объектов, как правило, оказываются довольно чувствительными к начальным условиям, к размерности задачи оптимизации, к виду целевых функций и ограничений. Поэтому необходимым условием для решения задач синтеза оптимальных структур технических объектов различной природы является использование методов и средств автоматизированного проектирования. Естественно, что формализованные модели и методы для САПР, с одной стороны, должны характеризоваться высокой степенью общности и достоверности, а с другой стороны, должны быть разрешимыми с вычислительной точки зрения.

Существо получения ММ объектов проектирования для решения задач структурного синтеза поясним на примерах компоновки, размещения и трассировки, довольно часто встречающихся в задачах конструирования ЭВА, распределения оборудования по производственным цехам, размещения цехов по территории завода, при проектировании линий электропередачи транспортных средств и т. п.

**Пример 6.2.** Задача компоновки. Под задачами компоновки понимают задачи разбиения множества  $D = \{d_1, \dots, d_n\}$  из  $n$  элементов на ряд непересекающихся подмножеств  $D_k, k = \overline{1, N}$ , 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:

$$F(X) \rightarrow \min;$$

$$\forall k, l \in \{1, 2, \dots, N\} (D_k \cap D_l = \emptyset), \bigcup_{k=1}^N D_k = D, \quad (6.13)$$

где  $D_k \subset D$  — множество элементов, принадлежащих  $k$ -му подмножеству разбиения.

Общее число разбиений

$$\prod_{k=1}^N C_{n - \sum_{q=1}^{k-1} n_q}^{n_k} = \frac{n!}{n_1! n_2! \dots n_N!}$$

при условии, что мощность каждого подмножества  $D_k$  является задан-

ной, т. е.  $|D_k| = n_k, \sum_{k=1}^N n_k = n$ .

Поэтому при реальном проектировании (при  $n > 100$ ) получить решение задачи компоновки путем перебора всех вариантов разбиения даже с использованием современных ЭВМ практически невозможно. Для уменьшения перебора задачу компоновки можно сформулировать в терминах целочисленного программирования. Пусть требуется распределить  $n$  компонентов электронной схемы между  $N$  блоками таким образом, чтобы суммарное число связей между блоками было минимально. Введем вектор  $X$  переменных проектирования, компоненты  $x_{ik} (i = \overline{1, n}, k = \overline{1, N})$  которого указывают на включение или невключение элемента  $d_i \in D$  в подмножество  $D_k$ , т. е.

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

Содержательно функция качества  $F(X)$  может характеризовать число связей между подмножествами  $D_i$  (при заданном числе  $N$  подмножеств), число подмножеств  $N$ , число типов подмножеств  $D_i$ , определяемых данным разбиением. Очевидно, все перечисленные функции качества  $F(X)$  следует минимизировать.

Пусть  $F(X)$  характеризует общее число связей между подмножествами  $D_k, k = \overline{1, N}$ . Тогда задача компоновки формулируется следующим образом: минимизировать целевую функцию

$$F(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^{N-1} \sum_{l=k+1}^N x_{ik} x_{jl} \pi_{kl} \quad (6.14)$$

при ограничениях

$$\sum_{k=1}^N x_{ik} = 1, i = \overline{1, n}; \sum_{i=1}^n x_{ik} = n_k, k = \overline{1, N}; \quad (6.15)$$

$$\sum_{i=1}^n \theta_i^{(S)} x_{ik} \leq V_S^{(k)}. \quad (6.16)$$

Здесь  $\pi_{ij}$  — число связей между элементами  $d_i$  и  $d_j$ ;  $\vartheta_i^{(S)}$  — значение параметра  $S$  для элемента  $d_i$ ;  $V_S^{(k)}$  — ограничение по параметру  $S$ , накладываемое на подмножество  $D_k$ , причем под параметром  $\vartheta_i^{(S)}$  элемента  $d_i$  может подразумеваться любой показатель, подчиняющийся свойству аддитивности (объем, масса, стоимость, энергоемкость и т. д.).

Целевая функция (6.14) является квадратичной, поэтому задача компоновки, сформулированная в виде задачи (6.14)—(6.16), является квадратичной задачей дискретного программирования.

**Пример 6.3.** Задача размещения. После того как решена задача компоновки, требуется определенным образом расположить компоненты, входящие в один блок. От того, как будут размещены микросхемы на определенной печатной плате, зависит длина соединительных проводников, от которой в свою очередь зависят уровень помех и время распространения сигналов. Подобные задачи получили название задач размещения. В общем случае требуется найти такое размещение компонентов  $d_1, d_2, \dots, d_n$  на множестве  $q_1, q_2, \dots, q_m$  ( $m \geq n$ ) позиций монтажного пространства, при котором суммарная длина электрических соединений между компонентами была бы минимальной. Введем псевдобулевы переменные

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

Тогда задача размещения может быть сформулирована в следующем виде: минимизировать целевую функцию

$$F(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \sum_{k=1}^m \sum_{s=1}^m x_{ik} x_{js} l_{ks} m_{ij} \quad (6.17)$$

при ограничениях

$$\sum_{k=1}^m x_{ik} = 1, \quad i = \overline{1, n}; \quad \sum_{i=1}^n x_{ik} \leq 1, \quad k = \overline{1, m}. \quad (6.18)$$

Здесь  $l_{ks}$  — расстояние между позициями  $q_k$  и  $q_s$ ;  $m_{ij}$  — число связей между компонентами  $d_i$  и  $d_j$ .

Первое ограничение гарантирует, что каждый компонент разместится только на одной позиции; второе ограничение гарантирует, что на каждую позицию будет назначено не более одного компонента.

Следует отметить, что среди известных критериев размещения наибольшее распространение получили минимум суммарной длины соединительных проводников, минимум наибольшей длины из всех длин соединительных проводников, минимум числа пересечений проводников и др.

**Пример 6.4.** Задача трассировки. Задача заключается в определении трасс соединений между компонентами схемы с учетом конструктивных ограничений, причем *трассой* называют множество связанных

отрезков, соединяющих точки электрической цепи. Задачи трассировки встречаются при конструировании печатных плат, при разработке систем водоснабжения, канализации, электроснабжения и т. д.

Критериями оптимальности в задачах трассировки могут выбираться минимум суммарной длины трасс, минимум числа соединений трасс длины больше заданной, минимум числа переходов между слоями в многослойных печатных платах и др.

Рассмотрим формальную постановку одной из разновидностей задачи трассировки, а именно — задачи построения связывающих сетей минимальной длины для цепей  $\alpha_k$ . Соединяемые по цепи  $\alpha_k$  точки образуют множество  $U_k$  мощностью  $|U_k| = n_k$ , в котором каждому элементу  $u_i \in U_k$  в монтажном пространстве соответствует одна точка. Введем псевдобулевы переменные:

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

Задача построения минимальной связывающей сети имеет вид: минимизировать целевую функцию

$$L(\mathbf{x}) = \sum_{i=1}^{n_k-1} \sum_{j=i+1}^{n_k} l_{ij} x_{ij} \quad (6.19)$$

при ограничениях

$$\sum_{i=1}^{n_k} x_{ij} \leq K_0, \quad i = \overline{1, n_k}; \quad (6.20)$$

$$y_{ij}^{(s)} = \sum_{r=1}^{n_k} y_{ir}^{(s-1)} x_{rj}; \quad y_{ij}^{(1)} = x_{ij}; \quad (6.21)$$

$$\sum_{s=1}^{n_k-1} y_{ij}^{(s)} \geq 1, \quad i = \overline{1, n_k}, \quad j = \overline{2, n_k}, \quad s = \overline{2, n_k-1}. \quad (6.22)$$

Здесь  $K_0$  — максимально допустимое число соединений, инцидентных одной точке цепи;  $y_{ij}^{(s)}$  — вспомогательные переменные.

Условия (6.21), (6.22) гарантируют связность определяемой сети.

Приведенные примеры показывают, что во многих случаях задачи структурного синтеза являются экстремальными комбинаторными задачами, которые могут быть сведены к задачам дискретного программирования. Оценка трудоемкости получения точных решений задач этого класса позволяет сделать вывод, что при реальном проектировании получение точных решений либо невозможно, либо требует больших затрат машинного времени. Поэтому для структурного синтеза каждого класса технических объектов необхо-

дима разработка специальных приближенных методов, позволяющих получать эффективные решения, близкие к оптимальным, а точные методы при этом служат для оценки качества получаемых решений.

**Параметрический синтез.** Задача параметрического синтеза заключается в определении наилучших значений параметров для выбранной структуры объекта с учетом всех требований ТЗ на проектируемый объект.

Функционирование любой проектируемой технической системы подчиняется определенным физическим законам. Закон функционирования технической системы описывается аналитическими соотношениями между входными, внутренними и выходными переменными системы. Эти переменные связаны определенными соотношениями с переменными проектирования  $X$ , под которыми понимаются внутренние переменные, допускающие варьирование. В процессе параметрического синтеза варьирование переменных проектирования  $X$  ведет к изменению выходных параметров  $Y$  системы.

Для пояснения сущности задач параметрического синтеза используют геометрическую интерпретацию, связанную с введением  $m$ -мерного пространства  $E^m$  пространства параметров проектирования (управляемых параметров) и  $k$ -мерного пространства  $E^k$  выходных параметров. Каждой точке пространства  $E^m$  и  $E^k$  соответствуют векторы  $X$  и  $Y$  значений переменных проектирования и выходных параметров соответствующего варианта проектируемого объекта.

Для постановки и решения задачи параметрического синтеза необходимо формирование целевой функции  $F(X)$ , отражающей качество функционирования проектируемой системы или объекта. Векторный характер критериев оптимальности (многокритериальность) в задачах проектирования обуславливает сложность проблемы постановки задач оптимизации.

Формально задачу параметрического синтеза можно представить как задачу нахождения вектора  $X \in E^m$ , который минимизирует целевую функцию

$$F(X) \rightarrow \min \quad (6.23)$$

при ограничениях

$$g(X) = 0 \text{ и } h(X) \leq 0, \quad (6.24)$$

где  $g(X)$  и  $h(X)$  — векторные функции от  $X$ , описывающие систему ограничений на параметры проектирования  $X$ .



В качестве целевой функции целесообразно выбирать один из параметров, наиболее полно характеризующих свойства проектируемого объекта. Если частный критерий при проектировании выбрать затруднительно, то прибегают к формированию обобщенных критериев. В зависимости от целей проектирования и типов математических моделей объекта проектирования целевые функции могут задаваться по-разному, для чего в САПР обычно предусмотрена библиотека целевых функций. Примерами целевых функций могут служить целевая функция максимального модуля отклонения характеристик объекта от заданных

$$F(\mathbf{X}) = \max_j |y_j^{(0)} - y_j^{(p)}(\mathbf{X})|, \quad j = \overline{1, m}, \quad (6.25)$$

и среднеквадратичная целевая функция

$$F(\mathbf{X}) = \sqrt{\sum_{j=1}^m (y_j^{(0)} - y_j^{(p)}(\mathbf{X}))^2}, \quad (6.26)$$

где  $y_j^{(0)}$  — заданное значение параметра  $y_j$ ;  $y_j^{(p)}$  — реальное значение этого параметра.

Очевидно, что функции вида (6.25) и (6.26) в задачах оптимизации следует минимизировать. Оптимизация может быть осуществлена различными методами, включающими весьма сложные аналитические и численные математические процедуры.

Если в задачах оптимального проектирования все переменные проектирования и состояний являются непрерывными, то для решения задач параметрического синтеза могут быть использованы методы решения задач нелинейного программирования, основанные на хорошо разработанных процедурах поиска экстремума функций. Однако не всегда все элементы в проектируемых объектах могут принимать любые значения в пределах некоторой допустимой области. Это связано прежде всего со стандартизацией и унификацией комплектующих изделий в различных областях техники. Так, в радиотехнике параметры резисторов и конденсаторов могут принимать только определенные значения из разрешенной шкалы номиналов, в строительстве плиты перекрытия, балки и другие комплектующие изделия имеют ряд определенных стандартных размеров. Кроме того, на параметры разрабатываемых объектов также накладывается ряд ограничений, учитывающих условия стандартизации и унификации. Так, в электротехнике и радиоэлектронике разрешается использовать только определенные

значения питающих напряжений, в вычислительной технике существуют стандартные градации емкости устройств памяти и коммутационного оборудования.

Поэтому для решения задач оптимизации при проектировании объектов с дискретными значениями параметров методы оптимизации непрерывных объектов непосредственно неприменимы. Эти задачи относятся к задачам дискретного программирования. Если при оптимизации часть параметров дискретна, а часть имеет непрерывный характер, то задача должна решаться методами частично дискретного программирования. Из-за недифференцируемости выходных параметров в задачах дискретного программирования довольно часто возникают трудности при вычислениях. Рассмотрим пример задачи параметрического синтеза.

**Пример 6.5.** Проектирование трехстержневой фермы. Цель проектирования — выбор конструкции трехстержневой фермы (рис. 6.2) минимальной массы. Проектирование сводится к выбору площадей поперечных сечений отдельных стержней  $x_1$ ,  $x_2$  и  $x_3$  (переменные проектирования), так чтобы ферма была по возможности легкой и удовлетворялись ограничения на напряжение, устойчивость при продольном изгибе, смещение и размеры стержней.

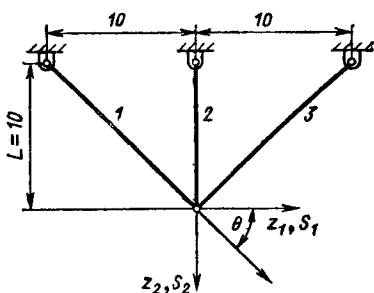


Рис. 6.2. Трехстержневая ферма

Целевая функция  $F(\mathbf{X})$  представляет собой массу конструкции (при  $\theta = 45^\circ$ ):

$$F(\mathbf{X}) = \rho g (10\sqrt{2}x_1 + 10x_2 + 10\sqrt{2}x_3), \quad (6.27)$$

где  $\rho g$  — плотность материала, из которого изготовлена ферма.

Горизонтальное и вертикальное смещения  $z_1$  и  $z_2$  общего узла получены из линейных соотношений теории упругости. Для данной конструкции эти уравнения имеют вид

$$\mathbf{K}(\mathbf{x})\mathbf{Z} - \mathbf{S} = 0, \quad (6.28)$$

где  $\mathbf{Z} = [z_1, z_2]^T$ ;

$$\mathbf{K}(\mathbf{x}) = \frac{\sqrt{2}E}{40} \begin{bmatrix} (x_1 + x_3) & (x_1 - x_3) \\ (x_1 - x_3) & (x_1 + x_3 + 2\sqrt{2}x_2) \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} P \cos \theta \\ P \sin \theta \end{bmatrix} \quad (6.29)$$

— положительно определенная матрица жесткости и вектор нагрузки

соответственно;  $E$  — модуль Юнга;  $\theta$  — угол приложения нагрузки, отсчитываемый от горизонтали.

Вычисляя деформации  $\varepsilon$  в зависимости от смещения узла и применяя закон Гука, найдем напряжение в каждом стержне:

$$\sigma_1 = E(z_1 + z_2)/20, \quad \sigma_2 = Ez_2/10, \quad \sigma_3 = E(z_2 - z_1)/20. \quad (6.30)$$

С помощью допустимых напряжений  $\sigma_i^{(0)}$ ,  $i = \overline{1, 3}$ , определенных для каждого стержня, ограничения на напряжения примут вид

$$\frac{E}{20} |z_1 + z_2| - \sigma_1^{(0)} \leq 0; \quad (6.31)$$

$$\frac{E}{10} |z_2| - \sigma_2^{(0)} \leq 0; \quad (6.32)$$

$$\frac{E}{20} |z_2 - z_1| - \sigma_3^{(0)} \leq 0. \quad (6.33)$$

Чтобы наложить ограничения по устойчивости, необходимо задать вид зависимости момента инерции от площади поперечного сечения для каждого стержня. Общей при инженерных расчетах является зависимость вида  $I = \beta x^2$ , где  $\beta$  — безразмерная постоянная. Подобная зависимость получается, если зафиксировать форму поперечного сечения и все его размеры менять в одинаковой пропорции. Осевые усилия имеют вид  $\Phi_i = \sigma_i x_i$ ,  $i = \overline{1, 3}$ , растяжения стержней считаются положительными. Ограничения по устойчивости имеют вид

$$-\Phi_i \leq \pi^2 EI_i / L_i^2, \quad i = \overline{1, 3}. \quad (6.34)$$

Используя равенства (6.28), можно записать:

$$\begin{aligned} -E(z_1 + z_2)x_1/20 - \pi^2 E\beta x_1^2/200 &\leq 0; \\ -Ez_2x_2/10 - \pi^2 E\beta x_2^2/100 &\leq 0; \\ -E(z_2 - z_1)x_3/20 - \pi^2 E\beta x_3^2/200 &\leq 0. \end{aligned} \quad (6.35)$$

Ограничения на смещения могут быть сформулированы в виде

$$|z_1| - z_1^{(0)} \leq 0, \quad |z_2| - z_2^{(0)} \leq 0, \quad (6.36)$$

где  $z_1^{(0)}$  и  $z_2^{(0)}$  — заданные верхние границы для  $z_1$  и  $z_2$  соответственно.

Наконец, требуется, чтобы площади поперечных сечений были неотрицательны, так что следует наложить ограничения

$$-x_1 \leq 0, \quad -x_2 \leq 0, \quad -x_3 \leq 0. \quad (6.37)$$

Теперь задачу проектирования можно рассматривать как задачу выбора таких переменных проектирования  $x_1$ ,  $x_2$  и  $x_3$ , которые мини-

мизируют  $F(X)$  и удовлетворяют ограничениям (6.28), (6.31)—(6.37). Заметим, что смещения  $z_1$  и  $z_2$  играют в данной формулировке важную роль. Их определяют из уравнения для конструкции (6.28), коль скоро заданы переменные проектирования. Эти переменные представляют собой отклик системы на приложенную нагрузку, их называют *переменными состояния*, а уравнения (6.28) — *уравнениями состояния*.

### 6.3. РАЗНОВИДНОСТИ ЗАДАЧ ОПТИМИЗАЦИИ

В задачах оптимального проектирования технических объектов вектор переменных проектирования  $X = (x_1, \dots, x_n)$  выбирают в результате определения экстремума целевой функции  $F(X)$  в допустимой области, заданной системой ограничений на параметры проектируемого объекта. В самом общем виде целевая функция и ограничения являются нелинейными функциями переменных проектирования  $X$ .

Задачи, в которых экстремум ищут в пределах неограниченного пространства переменных проектирования, относятся к задачам безусловной оптимизации. Найденные при этом *экстремумы* называют *безусловными*. Наличие ограничений любого вида приводит к задачам условной оптимизации, решение которых дает *условный экстремум*.

При решении задач оптимизации первоначально проверяют условия, которым должен удовлетворять вектор переменных проектирования  $X$ , минимизирующий (максимизирующий) критерий качества  $F(X)$ . Эти условия проверяют для отыскания стационарных точек, среди которых находится искомый вектор  $X$ .

Функция  $F(X)$ , определенная в  $E^m$ , имеет абсолютный минимум в  $X^* \in E^m$ , если

$$F(X^*) \leq F(X) \quad (6.38)$$

для всех  $X \in E^m$ . Минимум является строгим, если в (6.38) стоит знак строгого неравенства для  $X \neq X^*$ .

Функция  $F(X)$  имеет в  $X^*$  относительный минимум, если существует  $\varepsilon > 0$  такое, что для любой точки  $X \in S_\varepsilon(X^*)$  выполняется неравенство

$$F(X) - F(X^*) > 0, \quad (6.39)$$

где  $S_\varepsilon(X^*)$  — окрестность точки  $X^*$ . При определении максимума  $F(X)$  (6.39) должно быть заменено на неравенство

$$F(X) - F(X^*) < 0.$$

Если  $F(\mathbf{X})$  имеет в точке  $\mathbf{X}^*$  абсолютный минимум, то эта функция имеет там также и относительный минимум. Обратное не обязательно верно. Часто относительный минимум называют *локальным*, а абсолютный минимум — *глобальным*.

Рассмотрим необходимые и достаточные условия экстремума. Классические методы оптимизации используют тогда, когда известно аналитическое выражение функции  $F(\mathbf{X})$  и известно, что она по крайней мере дважды дифференцируема по переменным проектирования. Тогда для определения экстремума используют необходимые и достаточные условия безусловного экстремума. Эти условия легко получить с помощью разложения  $F(\mathbf{X})$  в окрестностях экстремальной точки  $\mathbf{X}^*$  в ряд Тейлора:

$$F(\mathbf{X}) - F(\mathbf{X}^*) = \sum_{i=1}^m \frac{\partial F(\mathbf{X}^*)}{\partial x_i} (x_i - x_i^*) + \frac{1}{2} \sum_{i=1}^m \sum_{k=1}^m \frac{\partial^2 F(\mathbf{X}^*)}{\partial x_i \partial x_k} (x_i - x_i^*) (x_k - x_k^*). \quad (6.40)$$

Если  $\mathbf{X}^*$  — точка максимума, то линейные члены в (6.40) равны нулю, тогда равны нулю составляющие вектора — градиента функции  $F(\mathbf{X})$ . Следовательно, необходимым условием экстремума является условие

$$\text{grad } F(\mathbf{X}) = 0,$$

где

$$\text{grad } F(\mathbf{X}) = \frac{\partial F(\mathbf{X})}{\partial \mathbf{X}} = \left( \frac{\partial F(\mathbf{X})}{\partial x_1}, \frac{\partial F(\mathbf{X})}{\partial x_2}, \dots, \frac{\partial F(\mathbf{X})}{\partial x_m} \right).$$

Все точки, где выполняется это условие, называются *стационарными*.

Достаточное условие максимума на основании (6.39) и (6.40) заключается в том, что сумма членов со вторыми частными производными должна быть отрицательной при любых  $\mathbf{X} \in S_\varepsilon(\mathbf{X}^*)$ , т. е.

$$(\Delta \mathbf{X})^T \Gamma(\Delta \mathbf{X}) < 0, \quad (6.41)$$

где  $\Delta \mathbf{X} = \mathbf{X} - \mathbf{X}^*$  — вектор-столбец;  $((\Delta \mathbf{X})^T = (\mathbf{X} - \mathbf{X}^*)^T$  — вектор-строка;

$$\Gamma = \begin{bmatrix} \frac{\partial^2 F(\mathbf{X})}{\partial x_1^2} & \frac{\partial^2 F(\mathbf{X})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F(\mathbf{X})}{\partial x_1 \partial x_m} \\ \frac{\partial^2 F(\mathbf{X})}{\partial x_1 \partial x_2} & \frac{\partial^2 F(\mathbf{X})}{\partial x_2^2} & \dots & \frac{\partial^2 F(\mathbf{X})}{\partial x_2 \partial x_m} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 F(\mathbf{X})}{\partial x_1 \partial x_m} & \frac{\partial^2 F(\mathbf{X})}{\partial x_2 \partial x_m} & \dots & \frac{\partial^2 F(\mathbf{X})}{\partial x_m^2} \end{bmatrix}$$

— матрица Гессе;  $\Gamma$  — знак транспонирования матрицы.

Условие (6.41) есть достаточное условие максимума. Матрицу  $\Gamma$ , удовлетворяющую условию (6.41) при любых  $\Delta \mathbf{X}$ , называют *отрицательно определенной*, а в случае  $(\Delta \mathbf{X})^T \Gamma (\Delta \mathbf{X}) > 0$  для любых  $\Delta \mathbf{X}$  — *положительно определенной*. Поэтому достаточные условия экстремума можно представить как требование отрицательной определенности матрицы Гессе для максимума или положительной определенности для минимума в экстремальной точке.

Если достаточные условия не выполняются, то имеем не экстремальную, а седловую точку.

При наличии ограничений на переменные проектирования  $\mathbf{X}$  решение задачи оптимизации ищут в некоторой допустимой области  $S$  согласно следующему направляющему принципу. Если  $F(x_1, \dots, x_m)$  является функцией нескольких переменных, определенных на допустимой области  $S$ , то максимальное значение  $F(\mathbf{X})$ , если оно существует, достигается в одной или более точках, которые могут принадлежать следующим множествам:

$K_1 = \{(x_1, \dots, x_m) \mid (x_1, \dots, x_m) \text{ — точка в области } S, \text{ в которой функция } F \text{ стационарна}\};$

$K_2 = \{(x_1, \dots, x_m) \mid (x_1, \dots, x_m) \text{ — точка границы области } S\};$

$K_3 = \{(x_1, \dots, x_m) \mid \text{ — функция } F(\mathbf{X}) \text{ не дифференцируема в точке } (x_1, \dots, x_m)\}.$

На рис. 6.3 приведен пример геометрической интерпретации многоэкстремальной задачи оптимального проектирования. На рисунке показаны линии равного уровня целевой функции  $F(\mathbf{X})$  ( $a_3 > a_2 > a_1 > a_0$ ) и видны три локальных оптимума, которые находятся в областях, определяемых общим направляющим принципом (точки  $\mathbf{X}_{1\text{лок}}$ ,  $\mathbf{X}_{2\text{лок}}$ ,  $\mathbf{X}_{3\text{лок}}$  являются точками локальных оптимумов, причем точка  $\mathbf{X}_{3\text{лок}}$  совпадает с глобальным оптимумом).

К сожалению, отсутствуют формальные признаки многоэкстремальных ситуаций. Исключением являются зада-

чи, где целевая функция выпуклая (в задачах минимизации) или вогнутая (при максимизации).

Следует напомнить, что функцию  $F(X)$  с числовыми значениями, определенными на выпуклом множестве  $S$ , называют *вогнутой*, если для любой пары точек  $X_1, X_2 \in S$  и для всех чисел  $\lambda$  ( $0 \leq \lambda \leq 1$ ) выполняется неравенство

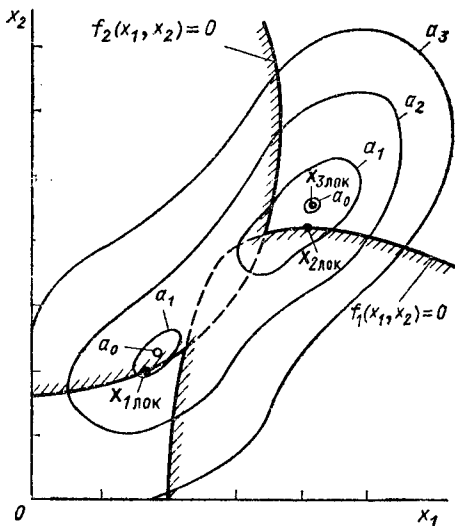


Рис. 6.3. Многоэкстремальная задача оптимального проектирования

$F(\lambda X_1 + (1-\lambda) X_2) \geq \lambda F(X_1) + (1-\lambda) F(X_2)$ . Если  $F(\lambda X_1 + (1-\lambda) X_2) \leq \lambda F(X_1) + (1-\lambda) F(X_2)$ , то функцию  $f(X)$  называют *выпуклой*. Если имеют место строгие неравенства, то говорят, что функция *строго вогнута* или *строго выпукла*.

Для дважды дифференцируемой функции критерий вогнутости или выпуклости формируется следующим образом. Дифференцируемая функция  $F(X)$  строго вогнута в некоторой окрестности точки  $X^{(0)} = (x_1^{(0)}, \dots, x_m^{(0)})$ , если выполняются условия

$$F_{11}(X^{(0)}) < 0, \begin{vmatrix} F_{11}(X^{(0)}) & F_{12}(X^{(0)}) \\ F_{21}(X^{(0)}) & F_{22}(X^{(0)}) \end{vmatrix} > 0,$$

$$\begin{vmatrix} F_{11}(X^{(0)}) & F_{12}(X^{(0)}) & F_{13}(X^{(0)}) \\ F_{21}(X^{(0)}) & F_{22}(X^{(0)}) & F_{23}(X^{(0)}) \\ F_{31}(X^{(0)}) & F_{32}(X^{(0)}) & F_{33}(X^{(0)}) \end{vmatrix} < 0, \dots,$$

т. е. если знаки этих определителей чередуются указанным образом. Здесь  $F_{ij}(\mathbf{X}^{(0)})$  — частная производная второго порядка, вычисленная в точке  $\mathbf{X}^{(0)}$ .

Функция  $F(\mathbf{X})$  строго выпукла в малой окрестности точки  $\mathbf{X}^{(0)}$ , если все определители, указанные выше, положительны.

*Достаточные условия для определения максимума или минимума* формулируются следующим образом: для того, чтобы в точке  $\mathbf{X}^{(0)}$  достигался внутренний локальный максимум, достаточно равенства нулю всех частных производных и строгой вогнутости функции в некоторой окрестности этой точки; для того чтобы в точке  $\mathbf{X}^{(0)}$  достигался внутренний локальный минимум, достаточно, чтобы все частные производные обращались в нуль и чтобы в малой окрестности этой точки функция была строго выпуклой.

В большинстве задач проектирования при отсутствии аналитического задания целевых функций проверка  $F(\mathbf{X})$  на выпуклость или вогнутость, как правило, невозможна, поэтому для решения задач оптимального проектирования используют методы поисковой оптимизации, основанные на исследовании малой окрестности оптимальной точки в допустимой области. Основные требования, предъявляемые к методу поиска, — высокая алгоритмическая надежность, приемлемые затраты машинного времени и требуемой памяти.

Методы поиска экстремума классифицируются по следующим признакам: в зависимости от характера экстремума существуют методы условной и безусловной, локальной и глобальной оптимизации; по числу переменных проектирования различают методы одномерного и многомерного поиска, а по характеру информации о виде целевой функции — методы нулевого, первого и второго порядков, причем в методах первого порядка используют градиент целевой функции, поэтому эти методы называются *градиентными*, в методах второго порядка применяют вторые производные, а в методах нулевого порядка производные не используют.

#### 6.4. ПОКАЗАТЕЛИ ЭФФЕКТИВНОСТИ И ВЫБОР МЕТОДОВ ПОИСКА ЭКСТРЕМУМА

В задачах оптимального проектирования в самой общей постановке целевая функция и ограничения являются нелинейными функциями переменных проектирования.



В общем случае эти задачи многоэкстремальны, поэтому для их решения следует применять методы определения поиска глобального экстремума, которые включают в себя один из способов генерации начальных точек, а также поиска локальных оптимумов.

Поиск локального оптимума состоит из следующих этапов определения: направления движения к оптимуму, длины шага поиска, окончания поиска.

Алгоритмы поиска локального оптимума  $X^*$  являются, как правило, итеративными, т. е. порождают последовательность векторов  $\{X^{(k)}\} = X_1, X_2, \dots, X_k$ , сходящуюся к вектору  $X^*$ .

Говорят, что вектор  $X^*$  является пределом сходящейся последовательности  $\{X^{(k)}\}$ , если для любого  $\epsilon > 0$  найдется такой номер  $N$ , что при  $k > N$  выполняется неравенство  $|X_k - X^*| < \epsilon$ . Отсюда следует, что допустимая область  $S$  должна вместе с любой сходящейся последовательностью содержать и ее предел. Такую область называют *замкнутой*. Примером замкнутой области может служить множество всех точек, удовлетворяющих ограничениям (6.2) и (6.3).

Множество всех точек пространства  $E^n$ , которые не содержатся в замкнутой области  $S(E^n \setminus S)$ , называют *открытым*. В замкнутой области  $S$ , если она не совпадает со всем пространством  $E^n$ , всегда можно найти точки, в  $\epsilon$ -окрестности которых имеются точки из  $E^n \setminus S$ . Такие точки области называют *граничными*. Множество всех граничных точек образует границу области  $S$ . В частности, если область  $S$  определяется условиями (6.2) и (6.3), его границу составляют те точки, в которых хотя бы одно из ограничений выполняется как строгое равенство.

Эффективность методов поиска локального оптимума определяется скоростью их сходимости к  $X^*$ , а критериями оценки качества выбора направления являются:

улучшение значения критерия оптимальности во вновь выбранной точке по сравнению с его величиной в данной точке;

наиболее быстрое убывание (возрастание) критерия в окрестности данной точки;

наиболее вероятное расположение экстремума с учетом кривизны гиперповерхности, представляющей критерий оптимальности.

Использование каждого из трех критериев выбора направления движения к оптимуму требует различного числа

обращений к модели. Необоснованное усиление критерия выбора направления поиска может привести к резкому возрастанию числа обращений к модели, а ослабление — к беспорядочному блужданию в окрестности оптимума. В обоих случаях возрастают затраты машинного времени на проектирование.

Проведем краткий анализ методов поиска экстремума. Особенности методов будем иллюстрировать примерами их применения к поиску экстремума функции  $F(\mathbf{X})$  в двумерном пространстве переменных проектирования.

**Методы безусловной оптимизации.** Для решения задачи безусловной оптимизации используют итерационные процессы вида

$$\mathbf{X}_k = \mathbf{X}_{k-1} + \alpha_k \Delta \mathbf{X}_k, \quad (6.42)$$

где  $\Delta \mathbf{X}_k$  — вектор, определяющий направление движения из точек  $\mathbf{X}_{k-1}$ ;  $\alpha_k$  — числовой множитель, значение которого определяет длину шага в направлении  $\Delta \mathbf{X}_k$ . Для большинства методов

$$\Delta \mathbf{X}_k = \mathbf{P}_k / \|\mathbf{P}_k\|, \quad (6.43)$$

где  $\mathbf{P}$  — вектор, указывающий направление поиска;  $\|\mathbf{P}\|$  — норма вектора  $\mathbf{P}$ ;  $k$  — индекс, обозначающий номер шага поиска.

Процесс (6.42) будет определен, если указаны способы построения вектора  $\Delta \mathbf{X}_k$  и вычисления величины  $\alpha_k$  на каждой итерации. От того, каким образом строится вектор  $\Delta \mathbf{X}_k$  и определяется множитель  $\alpha_k$ , непосредственно зависят свойства процесса: поведение функции  $F(\mathbf{X})$  на элементах последовательности  $\{\mathbf{X}^{(k)}\}$ , сходимость последовательности к решению, скорость сходимости и др. В то же время различные способы построения вектора  $\Delta \mathbf{X}_k$  и множителя  $\alpha_k$  требуют различных затрат машинного времени и различной емкости оперативной памяти ЭВМ.

Чтобы приблизиться к точке  $\mathbf{X}^*$ , естественно двигаться от точки  $\mathbf{X}_{k-1}$  в одном из направлений убывания функции  $F(\mathbf{X})$  (в направлении спуска). Если точка  $\mathbf{X}_{k-1}$  не является точкой минимума или стационарной точкой, то существует бесконечно много векторов  $\Delta \mathbf{X}_k$ , определяющих направления спуска из точки  $\mathbf{X}_{k-1}$ , причем каждый из них определяется условием

$$(F'(\mathbf{X}_{k-1}), \Delta \mathbf{X}) < 0,$$

вытекающим из (6.40).

Поскольку местонахождение точки  $X^*$  неизвестно, процесс поиска экстремума может быть прекращен в точке  $X_k$ , при этом число шагов  $r$ , разделяющих точки  $X_k$  и  $X_{k-r}$ , определяют из условия  $|X_k - X_{k-r}| < \epsilon$ . Следовательно, поиск прекращается, если расстояние, на которое продвинулась отображающая точка в пространстве переменных проектирования за последние  $r$  шагов, оказывается меньше заданного числа  $\epsilon$ .

Прямые методы оценки направлений. Наиболее простым является *метод покоординатного спуска* (метод Гаусса — Зейделя). Направление поиска выбирают поочередно вдоль всех координатных осей, т. е. вектор  $P_k$  в (6.43) состоит из нулевых элементов за исключением одного, равного единице.

Пусть  $X_0 = (x_1^{(0)}, \dots, x_n^{(0)})$  — начальная точка поиска (рис. 6.4, а). Результат первой итерации  $X_1$  получается из  $X_0$  при выполнении однопараметрической оптимизации по параметру  $x_1$ , т. е.  $X_1 = (x_1^{(1)}, x_2^{(0)}, \dots, x_n^{(0)})$ , причем

$$F(x_1^{(1)}, x_2^{(0)}, \dots, x_n^{(0)}) = \min_{x_1} F(x_1^{(1)}, x_2^{(0)}, \dots, x_n^{(0)}).$$

Результат второй итерации  $X_2$  получается из  $X_1$  путем оптимизации  $F(X)$  по параметру  $x_2$  и т. д. Далее процесс продолжается из точки  $X_n$  снова путем варьирования параметра  $x_1$  и т. д. Величину шага  $\alpha_k$  выбирают по способу оптимального шага.

Величина  $r$  за один цикл поиска равна размерности пространства, т. е.  $r = n$ .

Модификацией алгоритма покоординатного спуска является *метод ортогональных направлений* (*метод Розенброка*), который основан на вращении системы координат в соответствии с изменением скорости убывания критерия оптимальности. При этом направление одной оси соответствует наиболее вероятному направлению скорейшего убывания на данной итерации критерия оптимальности, а остальные находятся из условия ортогональности.

Если в задаче оптимального проектирования поверхность отклика ограничена концентрическими эллипсоидами, то точное местоположение оптимума не более чем за  $(2n-1)$  одномерных итераций позволяет получить *метод параллельных касательных*. Идея этого метода для  $n=2$  иллюстрируется на рис. 6.4, б. Метод заключается в поиске центра системы концентрических эллипсов. Первоначально определяют направление касательной  $\pi_0$  из точки

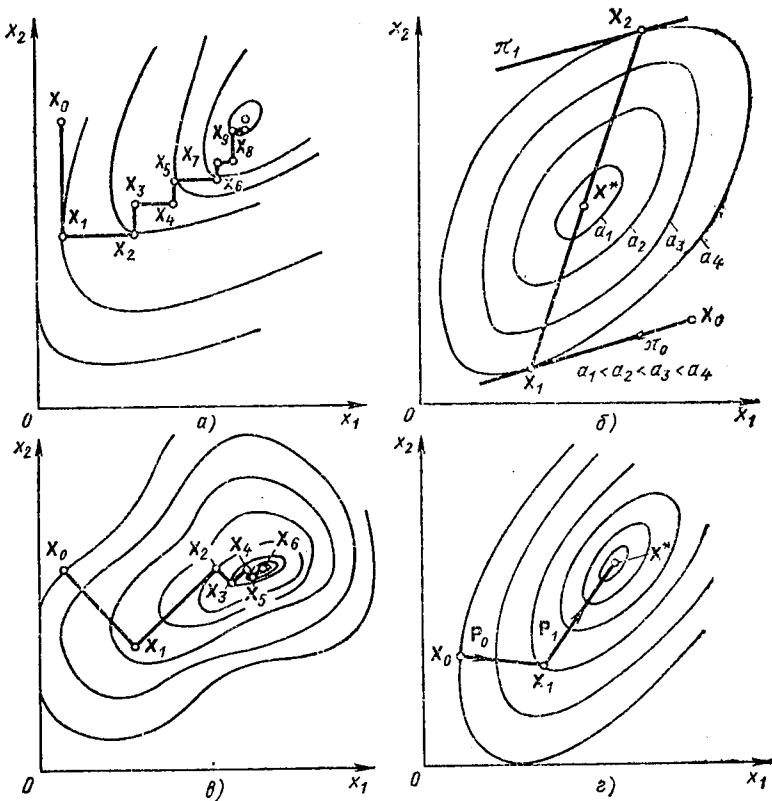


Рис. 6.4. Поиск экстремума квадратичной функции методом покоординатного спуска (а), методом параллельных касательных (б), методом наискорейшего спуска (в) и методом сопряженных градиентов (г)

$X_0$  к линии уровня (точка  $X_1$ ), затем строят произвольную прямую  $\pi_1$ , параллельную  $\pi_0$ , и на ней точку  $X_2$ , соответствующую точке касания  $\pi_1$  и линии уровня. Местоположение центра  $X^*$  определяется путем поиска вдоль линии  $X_1X_2$ .

Метод параллельных касательных пригоден для поиска оптимума на любой унимодальной функции, требует небольшого количества информации, но сопряжен с громоздкими вычислениями.

Градиентные методы. Методы поиска, в которых направление движения от итерации  $X_{k-1}$  к  $X_k$  опреде-

ляется градиентом (антиградиентом), вычисленным в точке  $X_{k-1}$ , носят название градиентных методов.

Наиболее применяемым в настоящее время из методов минимизации является *метод наискорейшего спуска*. В большой степени широкому распространению метода способствуют его сравнительная простота и возможность применения для минимизации весьма широкого класса функций. При определении направления поиска выбирают наибо́льшее убывание целевой функции  $F(X)$ , т. е.

$$P_k = - \text{grad } F(X_{k-1}).$$

Работа метода заключается в следующем. После определения градиента критерия оптимальности в точке  $X_k$  движутся вдоль направления антиградиента до точки, в которой достигается минимальное значение функции. Затем в этой точке снова определяют градиент и движутся по прямой согласно направлению нового антиградиента и т. д., пока не достигнут точки, имеющей наименьшее значение функции  $F(X)$ . На рис. 6.4, в приведен пример движения при поиске методом наискорейшего спуска оптимума для критерия оптимальности, зависящего от двух переменных. Направление  $\text{grad } F(X_{k-1})$  является касательным к поверхности уровня в точке  $X_k$ , и, следовательно,  $\text{grad } F(X_k)$  в точке  $X_{k+1}$  ортогонален  $\text{grad } F(X_{k-1})$ .

Простейшим среди градиентных методов является метод, при котором движение из точки  $X_{k-1}$  осуществляется *по антиградиенту с постоянным шагом*  $\alpha_k = \text{const}$ .

Если окажется, что  $F(X_k) > F(X_{k-1})$ , то для определения экстремума используют один из возможных алгоритмов поиска с возвратом. Так, в частности, поиск может производиться путем возврата в точку  $X_{k-1}$  и дальнейшего перемещения из этой точки вдоль антиградиента с шагом  $\alpha/2$ . Процедура дробления шага производится до тех пор, пока его значение не станет меньше некоторого малого положительного числа  $\epsilon$ .

Описанные варианты реализации градиентного метода отличаются друг от друга способом выбора длины шага. Скорость сходимости этих методов примерно одинакова, а трудоемкость каждой итерации вариантов процесса (6.42) различна только в способах определения параметра  $\alpha_k$ . Как правило, вычисления градиента в меньшем числе точек требует метод наискорейшего спуска.

Градиентные методы эффективны для решения задач минимизации гладких и выпуклых функций. В практике

автоматизированного проектирования для ускорения сходимости решения задач оптимизации градиентные методы используют в комбинации с другими, более эффективными на начальной стадии решения задачи.

**Метод сопряженных градиентов.** В градиентных методах для поиска экстремума использовались свойства ортогональности векторов. В методе сопряженных градиентов оптимум целевой функции ищется на основе свойств ортогональности приращений вектора градиентов. Для этой цели наряду с градиентом используют матрицу Гессе  $\Gamma$  критерия оптимальности. С помощью матрицы  $\Gamma$  удастся выбрать направление поиска, наиболее полно учитывающее особенности критерия оптимальности. Напомним, что векторы  $\mathbf{A}$  и  $\mathbf{B}$  называют сопряженными относительно симметричной и положительно определенной матрицы  $\Gamma$ , если скалярное произведение векторов  $\mathbf{A}$  и  $\Gamma\mathbf{B}$  равно нулю, т. е.  $\langle \mathbf{A}, \Gamma\mathbf{B} \rangle = 0$ . Направление поиска  $\mathbf{P}_{k+1}$  на  $k+1$ -м шаге определяется как

$$\mathbf{P}_{k+1} = -\text{grad } F(\mathbf{X}_k) + \beta_k \mathbf{P}_k,$$

где  $\beta_k = \frac{\langle \text{grad } F(\mathbf{X}_k), \text{grad } F(\mathbf{X}_k) \rangle}{\langle \text{grad } F(\mathbf{X}_{k-1}), \text{grad } F(\mathbf{X}_{k-1}) \rangle}$ .

Если критерий оптимальности представлен квадратичной функцией, то минимум функции достигается ровно за  $n$  шагов (рис. 6.4,  $z$ ). В случае критерия оптимальности произвольного вида метод позволяет для заданной погрешности получить приближенное решение быстрее, чем это позволяют сделать методы наискорейшего спуска и параллельных касательных.

Методы Ньютона и переменной метрики. Ускорение поиска экстремума связано с улучшением выбора сопряженных направлений. Довольно эффективным является поиск сопряженных направлений с одновременным накоплением информации о матрице Гессе критерия оптимальности. Используя соотношение

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \alpha_k \Gamma_k^{-1} \text{grad } F(\mathbf{X}_k), \alpha_k > 0, k = 0, 1, \dots,$$

которое является формулой *метода Ньютона с регулировкой шага*. Обычный метод Ньютона соответствует случаю  $\alpha_k = 1$ .

Метод Ньютона с регулировкой шага сходится к решению независимо от выбора начальной точки  $\mathbf{X}_0$  и обладает

либо линейной, либо квадратичной скоростью сходимости в зависимости от вида функции  $F(\mathbf{X})$ . В обычном методе Ньютона сходимость гарантируется лишь при наличии достаточно хорошего начального приближения.

При решении задач минимизации выпуклых функций метод Ньютона обеспечивает более высокую скорость сходимости последовательных приближений к решению по сравнению с градиентными методами, однако количество вычислений на итерации метода Ньютона высоко за счет необходимости вычисления и обращения матрицы вторых производных. Минимизация квадратичных функций происходит за один шаг.

На основе метода Ньютона разработан эффективный метод, получивший название *метода переменной метрики*. Идея метода заключается в использовании информации о градиенте критерия оптимальности для приближенного вычисления матрицы Гессе. Этот метод — итерационный. Поиск в нем ведется по формуле

$$\mathbf{X}_{k+1} = \mathbf{X}_k - h_k \mathbf{H}_k \text{grad } F(\mathbf{X}_k),$$

где  $\mathbf{H}_k$  — приближенно вычисляемая обратная матрица Гессе;  $h_k$  — величина шага, определяемая одномерной минимизацией целевой функции на луче —  $\mathbf{H}_k \text{grad } F(\mathbf{X}_k)$ .

Главное преимущество метода переменной метрики перед методом Ньютона — отказ от вычислений матрицы Гессе на каждой итерации. Положительно определенная матрица

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \frac{\Delta \mathbf{X}_{k-1} (\Delta \mathbf{X}_{k-1})^T}{\langle \Delta \mathbf{X}_{k-1}, \mathbf{R}_{k-1} \rangle} + \frac{\mathbf{H}_{k-1} \mathbf{R}_{k-1} (\mathbf{R}_{k-1})^T \mathbf{H}_{k-1}}{(\mathbf{R}_{k-1})^T \mathbf{H}_{k-1} \mathbf{R}_{k-1}},$$

где  $\Delta \mathbf{X}_{k-1}$  — приращение вектора переменных проектирования на предыдущем  $(k-1)$ -м шаге, имеет форму вектор-столбца;  $(\Delta \mathbf{X}_{k-1})^T$  — транспонированный вектор  $\Delta \mathbf{X}_{k-1}$ , т. е. вектор-строка;  $\mathbf{R}_{k-1} = \text{grad } F(\mathbf{X}_k) - \text{grad } F(\mathbf{X}_{k-1})$  — вектор-строка.

В начале вычислений нужно задаться произвольной положительно определенной матрицей  $\mathbf{H}_0$ , в частности  $\mathbf{H}_0$  может быть единичной матрицей. Шаг  $h_k$  выбирают по методу одномерной оптимизации.

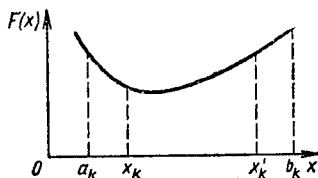
Ввиду того что в методе переменной метрики достаточно полно учитывается локальная информация, его целесообразно применять в окрестности оптимального решения.

**Методы одномерной оптимизации.** Эти методы позволяют найти оптимум для функций одной переменной. Они

не требуют условия дифференцируемости критерия оптимальности, а требуют только его непрерывности. Среди этих методов наиболее распространены методы чисел Фибоначчи, золотого сечения и полиномиальной аппроксимации. Для их применения необходимо знать интервал  $[a, b]$  переменной  $x$ , на котором функция  $F(x)$  имеет единственный минимум.

При применении метода чисел Фибоначчи должно быть зафиксировано число точек  $N$ , в которых производится вычисление критерия оптимальности.

Рис. 6.5. Расположение интервалов поиска экстремума методом чисел Фибоначчи



Пусть на  $k$ -й итерации интервал изменения переменной  $x$  сократился до  $[a_k, b_k]$ ,  $a \leq a_k \leq b_k \leq b$ . Тогда для вычисления следующего интервала  $[a_{k+1}, b_{k+1}]$  выбирают точки  $x_k$  и  $x'_k$  (рис. 6.5) по формулам

$$x_k = \frac{\Phi_{N-1-k}}{\Phi_{N+1-k}} (b_k - a_k) + a_k; \quad x'_k = \frac{\Phi_{N-k}}{\Phi_{N+1-k}} (b_k - a_k) + a_k,$$

где  $\Phi_k$  — числа Фибоначчи, определяемые с помощью рекуррентных соотношений

$$\Phi_k = \Phi_{k-1} + \Phi_{k-2}; \quad \Phi_0 = \Phi_1 = 1.$$

Если  $F(x_k) < F(x'_k)$ , то в качестве следующего интервала выбирают  $[a_{k+1}, b_{k+1}] = [a_k, x'_k]$ , если  $F(x_k) > F(x'_k)$ , то выбирают  $[a_{k+1}, b_{k+1}] = [x_k, b_k]$ , если  $F(x_k) = F(x'_k)$ , то может быть выбран любой из интервалов.

Последние точки задаются формулами

$$x'_{N-1} = (0,5 + \varepsilon)(b_{N-1} - a_{N-1}) + a_{N-1};$$

$$x_{N-1} = 0,5(b_{N-1} - a_{N-1}) + a_{N-1},$$

где  $\varepsilon$  — произвольно малое число, вводимое на последней итерации. Следует отметить, что длина последнего интервала неопределенности определяется как

$$b_N - a_N = (b_0 - a_0) / (2\Phi_N). \quad (6.44)$$



Выражение (6.44) позволяет определить количество вычислений критерия оптимальности исходя из требуемой точности поиска.

В методе золотого сечения сохраняется постоянным отношение длин двух последовательных интервалов неопределенности:

$$\tau = L_k/L_{k+1} = 1,61803\dots$$

По результатам двух экспериментов устанавливают, какую область неопределенности оставить для дальнейших исследований. Процесс поиска оптимума можно продолжать сколь угодно долго. После  $N$  испытаний длина интервала неопределенности составляет  $L_N = 1/\tau^{N-1}$ .

Отметим, что метод золотого сечения требует сравнительно небольшого объема памяти ЭВМ и прост в реализации.

Метод полиномиальной аппроксимации заключается в определении полинома, аппроксимирующего функцию  $F(\mathbf{X})$  (чаще всего — квадратичного полинома), и поиске его минимума.

В конкретных задачах оптимального проектирования довольно часто зависимость критерия оптимальности  $F$  от параметров проектирования  $\mathbf{X}$  получается слишком сложной. В этих случаях вместо вышеизложенных регулярных методов оптимизации используют *методы случайного поиска*. В этих методах направление поиска  $\mathbf{P}_k$  выбирают случайно, например, равновероятно в пределах гиперсферы с центром в точке  $\mathbf{X}_{k-1}$ . Существует огромное число алгоритмов случайного поиска. Следует отметить, что регулярные алгоритмы поиска являются частным (а точнее, вырожденным) случаем стохастических алгоритмов.

**Методы условной оптимизации.** Задачи условной оптимизации, заключающиеся в минимизации некоторого критерия оптимальности с ограничениями на область существования переменных проектирования, относятся к классу задач математического программирования.

Одним из наиболее простых и широко известных методов решения задачи математического программирования является *метод штрафных функций*. Основная идея метода состоит в приближенном сведении задачи минимизации функции  $F(\mathbf{X})$  при ограничениях  $Q_i(\mathbf{X}) \leq 0$ ,  $i = \overline{1, n}$ , к задаче минимизации функции

$$\Phi(\mathbf{X}, t) = F(\mathbf{X}) + tR(\mathbf{X}) \quad (6.45)$$

без ограничений. При этом вспомогательную функцию  $\Phi(\mathbf{X}, t)$  подбирают так, чтобы она совпадала с функцией  $F(\mathbf{X})$  внутри допустимой области  $S$  и быстро возрастала вне ее.

В выражении (6.45)  $R(\mathbf{X})$  — дифференцируемая функция штрафа, удовлетворяет следующим условиям:  $R(\mathbf{X}) = 0$ , если  $\mathbf{X} \in S$ , и  $R(\mathbf{X}) > 0$ , если хотя бы для одного  $k$  будет  $Q_k(\mathbf{X}) > 0$ , ( $k = \overline{1, n}$ ),  $t$  — некоторое положительное число — коэффициент штрафа.

Примерами функций  $R(\mathbf{X})$  могут служить выражения

$$R_1(\mathbf{X}) = \sum_{i=1}^n \max R_i \{(\mathbf{X}), 0\}^\alpha, \quad \alpha \geq 1;$$

или

$$R_2(\mathbf{X}) = \sum_{i=1}^n \left[ \frac{R_i(\mathbf{X}) + |R_i(\mathbf{X})|}{2} \right]^2.$$

На рис. 6.6 иллюстрируется метод штрафных функций в одномерном случае. Допустимая область  $S$  определяется ограничением  $R(\mathbf{X}) \geq 0$ , в этой области  $F(\mathbf{X})$  и  $\Phi(\mathbf{X}, t)$  совпадают. В области, где  $R(\mathbf{X}) < 0$ , функция  $\Phi(\mathbf{X}, t)$  резко возрастает. На рисунке  $X_1$  и  $X^*$  — точки безусловного и условного минимумов.

В задачах условной оптимизации, в которых ограничения заданы только в виде неравенств, возможно построение обобщенного критерия оптимальности с помощью барьерных функций. Значения, принимаемые барьерной функцией, неограниченно возрастают при приближении к границе допустимой области.

Примером барьерной функции является  $R(\mathbf{X}) = -\sum_{i=1}^n \ln [-R_i(\mathbf{X})]$ . Эта функция существует только внутри допустимой области  $S$ . Вне области  $S$  и на ее границе функция  $R(\mathbf{X})$  не определена, а при приближении к границе области  $S$  она неограниченно возрастает.

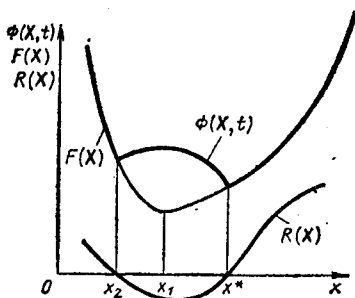


Рис. 6.6. Метод штрафных функций

Исходя из организации поиска условного оптимума иногда метод штрафных функций называют *методом внешней точки*, а метод барьерных функций — *методом внутренней точки*.

При использовании штрафных и барьерных функций необходимо принять меры против переполнения разрядной сетки ЭВМ: осторожно подходить к выбору вида штрафной функции и, если это необходимо, ввести ограничивающую функцию в (6.45), которая замедлила бы рост второго слагаемого, начиная с некоторого  $X$ .

Таким образом, задачу нелинейного программирования удастся свести к задаче или последовательности задач безусловной минимизации.

### 6.5. ЗАДАЧА ОПТИМИЗАЦИИ ДОПУСКОВ И ТЕХНИЧЕСКИХ ТРЕБОВАНИЙ

Качество проектируемых объектов в значительной мере определяется характером постановки задачи параметрического синтеза, реализуемой при проектировании, т. е. тем, насколько сформулированные целевая функция и ограничения отражают объективно существующие требования к свойствам объекта. При формализации ТЗ такие требования выражаются в виде условий работоспособности. Условие работоспособности — это требуемое соотношение между выходным параметром  $y_j$ , значения которого зависят от принимаемых проектных решений, и предельно допустимым значением — нормой  $y_j^{(0)}$ . Величину  $y_j^{(0)}$  часто называют также техническим требованием на параметр  $y_j$ . Условия работоспособности могут иметь одну из следующих форм:

$$y_j < y_j^{(0)}, \quad (6.46a)$$

$$y_j > y_j^{(0)}, \quad (6.46б)$$

$$y_{j\min}^{(0)} < y_j < y_{j\max}^{(0)}. \quad (6.46в)$$

Формы (6.46б) и (6.46в) могут быть сведены к форме (6.46а), поэтому в дальнейшем будем считать, что все условия работоспособности в ТЗ имеют вид (6.46а).

Область в пространстве  $X\Pi$  управляемых параметров, в которой все условия работоспособности, а заданные прямые ограничения на управляемые параметры  $x_i$  вида

$$x_{i\min} \leq x_i \leq x_{i\max} \quad (6.46г)$$

удовлетворяются, называют *областью работоспособности*  $Z_0$ .

Задача оптимизации допусков сводится к определению размеров допусксовой области  $Z_d$  и ее расположения в пространстве  $X_P$ . Цель оптимизации допусков — максимизация размеров области  $Z_d$  при выполнении ограничений на степень несовпадения областей  $Z_0$  и  $Z_d$ .

Решение задачи оптимизации допусков выполняется в два этапа: на первом ищут какую-либо точку  $X_s \in Z_0$ , называемую опорной; на втором определяют оптимальную допусксовую область.

На первом этапе для определения опорной точки целесообразно использовать постановку задачи оптимизации параметров, известную под названием максиминной постановки. Последняя приводит к получению опорной точки внутри области  $Z_0$  на достаточном удалении от границ, что удобно для реализации алгоритмов второго этапа.

При максиминной постановке вводится количественная оценка  $s_j$  степени выполнения  $j$ -го условия работоспособности. Каждая из оценок  $s_j$  может носить детерминированный или статистический характер. При детерминированном подходе используют формулу

$$s_j = (y_j^{(0)} - y_j) / y_j^{(0)},$$

при статистическом — формулу

$$s_j = (y_j^{(0)} - M_j) / \delta_j,$$

где  $M_j$  — оценка математического ожидания;  $\delta_j$  — оценка рассеяния параметра  $y_j$ .

В частности, величина  $\delta_j$  может рассматриваться как некоторый весовой коэффициент, указание его физического смысла упрощает правильное задание его численного значения без трудоемких статистических расчетов.

Так как вероятность надежного функционирования объекта определяется главным образом наименьшей из вероятностей выполнения отдельных условий работоспособности, то в первую очередь нужно увеличивать наименьший из запасов  $s_j$ . Поэтому в качестве целевой функции  $F(X)$  следует выбрать наименьший из запасов, и задача оптимизации параметров проектируемого объекта формулируется как максиминная задача нелинейного программирования:

$$\max_{X \in Z_0} \min_j s_j(X).$$

Ограничениями задачи при этом будут прямые ограничения (6.46г).

Максиминный критерий запаса работоспособности применим при наличии у проектируемого объекта параметров с условиями работоспособности любого вида. Этот критерий в зависимости от конкретной ситуации может рассматриваться либо как детерминированный, либо как статистический.

На втором этапе формулируется задача вписывания гиперпараллелепипеда допусков  $Z_d$  в область работоспособности  $Z_0$ . В этой задаче к исходным данным относятся:

1. Область работоспособности  $Z_0$ , задаваемая прямыми ограничениями (6.46г) и условиями работоспособности, приведенными к виду  $y_j(X) \leq y_j^{(0)}$ .

2. Точка  $X_3 \in Z_0$ .

3. Соотношения  $\alpha_i$  между допусками  $g_i$ , равноценными с позиций затрат на их получение при производстве.

При решении задачи сначала нормируют управляемые параметры в соответствии с заданными соотношениями между допусками

$$u_i = \alpha_i (x_i - x_{3i}) / x_{3i} + 1,$$

где  $u_i$  — нормированный параметр  $x_i$ ;  $\alpha_i = g_1 / g_i$ .

При таком нормировании оптимальной формой допусковой области является гиперкуб с ребрами, параллельными координатным осям. Областям  $Z_0$  и  $Z_d$  в нормированном пространстве управляемых параметров соответствуют области  $U_0$  и  $U_1$ .

Задача вписывания формулируется как задача определения центра гиперкуба  $U^* \in U_0$ , имеющего максимальную длину ребра при условии, что оценка рассогласования положений областей работоспособности  $U_0$  и допусковой области  $U_1$  не превышает заданной величины.

Рассмотрим один из алгоритмов вписывания, основанный на линеаризации зависимостей  $y_j(U)$  и ориентированный на случай, когда  $U_1 \subset U_0$ . Этот случай часто называют оптимизацией параметров и допусков в условиях 100 %-ного выхода годных. Принимается также допущение о постоянстве знаков коэффициентов влияния  $a_{ji} = \partial y_j / \partial x_i$ , т. е.

$$\text{sgn}(a_{ji}) = \text{const}, \quad (6.47)$$

в пределах  $U_0$ ,  $i=1, 2, \dots, n$ ;  $n$  — количество управляемых параметров.

Для  $j$ -го участка границы области  $U_0$ , выражаемого уравнением

$$y_j^{(0)} - y_j(\mathbf{U}) = 0, \quad (6.48)$$

известно положение нормали, проходящей через центр  $\mathbf{U}^*$  гиперкуба. Система уравнений этой нормали в параметрическом виде

$$u_i - u_i^* = a_{ji} h_j, \quad (6.49)$$

где  $h_j$  — параметр;  $a_{ji}$  определяется в точке пересечения нормали с поверхностью (6.48).

Запишем систему уравнений для диагонали искомого гиперкуба, имеющей ту же ориентацию, что и  $j$ -я нормаль (6.49):

$$u_i - u_i^* = \operatorname{sgn}(a_{ji}) \gamma_j, \quad (6.50)$$

где  $\gamma_j$  — параметр.

В соответствии с (6.47) вычисление  $a_{ji}$  может быть выполнено в точке  $\mathbf{U}_3$ . Проведем линейризацию границ (6.48) области  $U_0$ , используя разложение  $y_j(\mathbf{U})$  в ряд Тейлора в окрестности опорной точки  $\mathbf{U}_3$ :

$$y_j^{(0)} - y_j(\mathbf{U}_3) - \sum_{i=1}^n a_{ji} (u_i - u_i^*) - \sum_{i=1}^n a_{ji} (u_i^* - u_{3i}) = 0. \quad (6.51)$$

Обозначим  $c_j(\mathbf{U}^*) = y_j^{(0)} - y_j(\mathbf{U}_3) - \sum_{i=1}^n a_{ji} (u_i^* - u_{3i})$ . Тогда (6.51) перепишем в виде

$$c_j(\mathbf{U}^*) - \sum_{i=1}^n a_{ji} (u_i - u_i^*) = 0. \quad (6.52)$$

Точка пересечения диагонали (6.50) с границей (6.52) определяется из совместного решения уравнений (6.50) и (6.52):  $c_j(\mathbf{U}^*) - \gamma_j \sum_{i=1}^n \operatorname{sgn}(a_{ji}) a_{ji} = 0$ , т. е.

$$\gamma_j = c_j(\mathbf{U}^*) / \sum_{i=1}^n |a_{ji}|. \quad (6.53)$$

Здесь  $\gamma_j$  имеет смысл половины длины ребра гиперкуба, имеющего центр в точке  $\mathbf{U}^*$  и вершину на гиперповерхности (6.52). Всего имеем  $m$  условий работоспособности и, следовательно, не более чем  $m$  вершин на границах  $U_0$ ,

Чтобы выполнить условие  $U_1 \subset U_0$ , нужно контролировать принадлежность области  $U_0$  вершины, которой соответствует минимальная величина  $\gamma_q$  среди величин  $\gamma_i$ . Чтобы область  $U_1$  имела максимально возможные размеры, нужно максимизировать  $\gamma_q$ , варьируя положение центра  $U^*$ .

Таким образом, задача вписывания допусковой области в область работоспособности формулируется как задача линейного программирования

$$\max_{U^* \in U_0} \gamma_q(U^*)$$

при ограничениях

$$\gamma_p(U^*) - \gamma_q(U^*) \geq 0; \quad u_i^* \geq 0; \quad \|U_s - U^*\| < \gamma_q.$$

Последнее ограничение ставится в том случае, если точка  $U_s$  должна принадлежать допусковой области.

Отметим, что основные затраты машинного времени на реализацию алгоритма связаны с анализом чувствительности. Анализ чувствительности методом приращений требует  $n+1$  раз обращаться к математической модели объекта. Первое обращение производится при значении вектора управляемых параметров  $U_s$  и позволяет вычислить  $y_j(U_s)$ , фигурирующие в (6.51). Каждое последующее обращение позволяет вычислить очередную строку матрицы чувствительности и в итоге дает значения  $a_{ji}$ . Теперь полностью определена линеаризованная модель объекта (6.53). Манипулирование ею при решении задач линейного программирования не требует заметных затрат машинного времени.

При большом числе  $n$  управляемых параметров (несколько сотен и более) применение алгоритма, включающего анализ чувствительности методом приращений, становится нерациональным. С ростом  $n$  более предпочтительными оказываются алгоритмы, основанные на статистических испытаниях.

Одним из таких алгоритмов является алгоритм центрирования по методу статистического градиента. На каждом шаге алгоритма выполняются  $N$  статистических испытаний с выбором случайных точек в пределах некоторой области  $U_{ст}$ . По результатам испытаний выделяются те точки  $U_p \in U_{ст}$ , которые оказались в области работоспособности  $U_0$ . Для следующего шага в качестве координат  $u_i^*$  центра  $U^*$  допусковой области  $U_{ст}$  принимаются средние арифметические значения координат  $u_{ip}$  выделенных точек  $U_p$ .

Если в процессе испытаний область  $U_{ст}$  выбирается в соответствии с некоторыми условиями (например,  $U_0 \subset U_{ст}$  и  $U_{ст}$  имеет форму гиперболы) и если выполнено достаточное количество шагов, то точка  $U^*$  может быть принята в качестве центра области  $U_0$ , а окончательная допусковая область  $U_d$  устанавливается в соответствии с характером распределения точек  $U_p$  на последнем шаге центрирования.

Задача оптимизации допусков обычно решается на том иерархическом уровне проектирования, на котором в качестве управляемых параметров фигурируют параметры базовых элементов. Рассчитанные допуски используются для выбора унифицированных деталей и узлов по справочникам и каталогам либо служат непосредственными исходными данными для последующего технологического проектирования.

На промежуточных иерархических уровнях нисходящего функционального или конструкторского проектирования также возникают задачи, подобные задаче оптимизации допусков. Предположим, что на  $k$ -м иерархическом уровне управляемыми параметрами системы являются параметры  $y_j$ . На следующем,  $(k+1)$ -м иерархическом уровне эти же параметры рассматриваются уже как выходные параметры подсистем, а управляемыми параметрами здесь оказываются другие параметры  $x_i$ . Для выполнения проектирования на  $(k+1)$ -м иерархическом уровне на выходные параметры  $y_j$  нужно задать условия работоспособности. Очевидно, что эти условия должны быть результатом проектирования на  $k$ -м уровне, т. е. должны быть определены не только некоторая оптимальная точка  $Y^*$  в пространстве параметров  $y_j$ , но и технические требования  $y_j^{(0)}$  на эти параметры.

Задача оптимального расчета технических требований  $y_j^{(0)}$  по своей постановке и методам решения близка к рассмотренной выше задаче оптимизации допусков. Исходными данными при ее решении являются условия работоспособности, задаваемые на параметры системы, а результатом должны быть условия работоспособности для подсистем. При оптимизации технических требований на последующем  $(k+1)$ -м уровне исходными данными будут условия работоспособности на параметры подсистем, а результатом — условия работоспособности для элементов подсистем и т. д.

Таким образом, блочно-иерархический подход приво-



дит к формулировке основных оптимизационных задач нисходящего проектирования как задач оптимального преобразования технического задания на объект  $k$ -го иерархического уровня в технические задания на объекты  $(k+1)$ -го иерархического уровня. Эти задачи решаются с помощью статистических или детерминированных алгоритмов вписывания гиперфигур в заданную область  $n$ -мерного пространства параметров.

## 6.6. МЕТОДЫ ОПТИМИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

Комплексные САПР охватывают все этапы разработки технических объектов, в том числе и этап технологической подготовки производства. Автоматизация проектирования технологических процессов включает в себя разработку принципиальных схем технологических процессов, маршрутной технологии, операционной технологии и получение управляющей информации на машинных носителях для программно-управляемого технологического оборудования.

Любой технологический процесс независимо от его физической природы всегда можно представить в виде некоторой системы, а следовательно, для его организации сле-

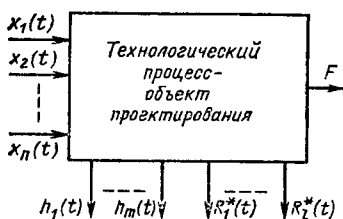


Рис. 6.7. Постановка задачи оптимизации технологического процесса

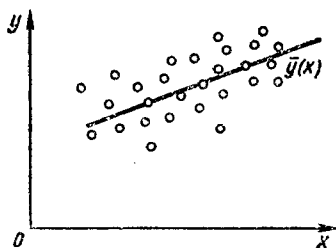


Рис. 6.8. Линия регрессии

дует применять системный подход, сущность которого заключается в комплексном, едином рассмотрении всех частей систем технологического проектирования и в гармоническом их сочетании.

Постановку задачи оптимизации технологического процесса можно представить следующим образом (рис. 6.7). Технологический процесс рассматривается как объект

проектирования, на вход которого поступает вектор  $\mathbf{X}(t) = (x_1(t), \dots, x_n(t))$  входных переменных, а скалярный выход  $F$  характеризует качество технологического процесса. Требуется сформировать такой вектор управляющих воздействий  $\mathbf{R}^*(t) = (R_1^*(t), \dots, R_m^*(t))$ , который минимизировал бы значение показателя  $F$ , т. е.

$$F(\mathbf{X}(t), \mathbf{R}^*(t)) = \min_{\mathbf{R}(t)} \{F_k(\mathbf{X}(t), \mathbf{R}(t))\}.$$

Выходы  $h_1(t), \dots, h_m(t)$  характеризуют состояние технологического процесса и индицируют нежелательные режимы работы оборудования или выход контролируемых параметров технологического процесса за установленные пределы:

$$h_j(t) \leq b_j, \quad j = \overline{1, m},$$

где  $b_j$  — требуемые или допустимые значения соответствующих параметров.

Таким образом, оптимизация технологического процесса при проектировании рассматривается как задача определения оптимального вектора управления  $\mathbf{R}^*(t)$ , минимизирующего целевую функцию  $F(\mathbf{X}(t), \mathbf{R}(t))$  при условии выполнения заданных ограничений.

Подобные задачи оптимизации решают в два этапа. На первом этапе определяют идеальный вектор управления  $\mathbf{R}_{ид}^*(t)$ , обеспечивающий оптимизацию технологического процесса. Практически реализовать это не представляется возможным, и вектор  $\mathbf{R}_{ид}^*(t)$  является эталоном, к которому надо стремиться. Зная  $\mathbf{R}_{ид}^*(t)$ , на втором этапе выбирают реализуемый квазиоптимальный вектор управления, с помощью которого стараются получить решение, наименее отличающееся от идеального и в то же время реализуемое наиболее просто.

Следует отметить, что в большинстве практических случаев оптимизированные технологические процессы дополнительно подвергаются наладке и корректировке, поскольку при построении математических моделей процессов невозможно учесть все влияющие на процесс факторы.

На любой технологический процесс оказывает влияние множество случайных факторов (неточность оборудования, неточности режущего инструмента и приспособлений, внутренние напряжения обрабатываемой детали, разброс параметров у электрорадиоэлементов при монтаже электрон-

ных схем и т. д.). Поэтому параметры изготавливаемых изделий являются случайными величинами, причем влияние действующих случайных факторов на изменение параметров изделий можно, как правило, определить исходя из статистического анализа. В подобных случаях очень важно учитывать характер взаимосвязи между случайными величинами. Для количественного выражения этой взаимосвязи служат регрессия и корреляция. Остановимся более подробно на этих понятиях.

Пусть  $x$  и  $y$  — случайные величины, характеризующие параметры некоторого изделия, причем упорядоченная пара  $(x, y)$  характеризует параметры одного варианта изделия и может быть изображена точкой на плоскости. Полная совокупность вариантов изображается множеством точек, показанных на рис. 6.8. Математические ожидания случайных величин  $x$  и  $y$  равны соответственно  $M_{(x)}$  и  $M_{(y)}$ , и среднеквадратичные отклонения  $\sigma_x$  и  $\sigma_y$  характеризуют рассеивание величин  $x$  и  $y$  относительно их математических ожиданий.

Рассмотрим зависимость  $\bar{y}(x)$ , являющуюся условным математическим ожиданием  $M_{(y|x)}$ . Используя выражение для условного математического ожидания и обозначая через  $p(x, y)$  совместную вероятность данных значений  $x$  и  $y$ , находим

$$\bar{y}(x) = M_{(y|x)} = \frac{\sum_y yp(x, y)}{\sum_y p(x, y)}. \quad (6.54)$$

Определяя  $\bar{y}(x)$  при различных  $x$ , можно построить линию, графически выражающую эту зависимость и называемую линией регрессии  $y$  по  $x$  (рис. 6.8). Аналогично может быть получена зависимость  $\bar{x}(y)$ , называемая регрессией  $x$  по  $y$ .

На практике наиболее часто встречается случай линейной регрессии, уравнение которой записывается в виде

$$\bar{y}(x) = a + b(x - M_{(x)}). \quad (6.55)$$

Коэффициенты  $a$  и  $b$  выбирают такими, чтобы получить наибольшую концентрацию точек  $(x, y)$  вблизи прямой  $\bar{y}(x)$ , что выражается условием

$$\varphi(a, b) = M \{ [y - \bar{y}(x)]^2 \} = \min. \quad (6.56)$$

Выражение (6.56) с учетом (6.55) дает следующую си-

стему уравнений для определения коэффициентов  $a$  и  $b$ :

$$M_{(y)} - a = 0; \quad M[y(x - M_{(x)})] - b\sigma_x^2 = 0. \quad (6.57)$$

Величину  $\mu_{xy} = M[y(x - M_{(x)})]$  называют ковариацией между  $x$  и  $y$ . Она служит мерой взаимной связи между случайными величинами  $a$  и  $b$ .

Из (6.57) находят значения  $a = M_{(y)}$  и  $b = \mu_{xy}/\sigma_x^2$ , определяющие линию регрессии.

Ковариация  $\mu_{xy}$  зависит от дисперсий самих случайных величин, поэтому для оценки взаимосвязи между случайными величинами более удобен коэффициент корреляции  $r_{xy} = \mu_{xy}/(\sigma_x\sigma_y)$ , который может меняться от нуля для независимых случайных величин до единицы, если случайные величины связаны линейной функциональной зависимостью.

При технологическом проектировании в качестве критериев оптимальности могут рассматриваться такие показатели эффективности, как себестоимость производства изделий, производительность технологических процессов, основное технологическое время и т. д.

Методы оптимизации технологических процессов рассмотрены на примерах управления технологическим процессом производства магнитоуправляемых контактов и управления технологической установкой.

**Пример 6.6.** Управление технологическим процессом на основе текущего регрессионного анализа. Рассмотрим технологический процесс производства магнитоуправляемых контактов (МК). Основная задача производства МК — получение изделий с заданными величинами напряженности магнитного поля  $\Theta$  и зазора  $S$  между контактами.

Одной из задач автоматизации проектирования технологического процесса производства МК является определение функциональной связи между величинами  $\Theta$  и  $S$  последующей реализацией математической модели процесса управления заварки лепестков МК на управляющей мини- или микро-ЭВМ.

Схема управления заваркой лепестков МК приведена на рис. 6.9.

Технологический процесс установки зазора по заданной напряженности магнитного поля состоит в следующем: заваривается верхний пружинный контакт 6; нижний пружинный контакт 7 отводится исполнительным механизмом 8 на максимальное расстояние; в катушку 4 подается ток, значение которого обеспечивает заданную величину напряженности магнитного поля для срабатывания МК; начинается движение контакта 7 к контакту 6.

Движение контактов происходит до момента их замыкания под действием поля катушки 4. Значение поля пропорционально силе тока  $I$ ,

протекающего в катушке. При замыкании регистрирующее устройство 9 срабатывает и останавливает исполнительный механизм 8; нижний пружинный контакт 7 заваривается, зазор становится равным величине  $S$ .

Исследования показывают, что величины  $\Theta$  и  $S$  после заварки не связаны жесткой функциональной зависимостью. Характер геометриче-

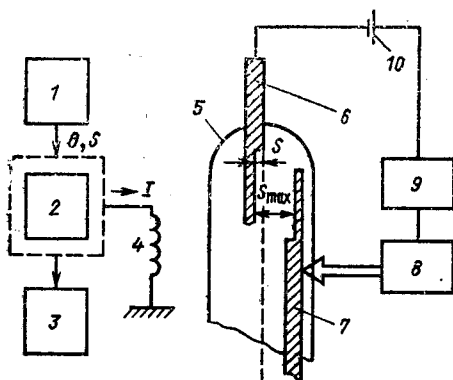


Рис. 6.9. Схема управления заваркой лепестков МК:

1 — пульты статистического контроля  $\Theta$  и  $S$ ; 2 — управляющая мини-ЭВМ; 3 — блок выдачи рекомендаций технологу; 4 — катушка МК; 5 — стеклянный баллон; 6, 7 — верхний и нижний пружинные контакты; 8 — исполнительный механизм; 9 — регистрирующее устройство; 10 — источник питания

ского места точек  $\Theta(S)$  зависит от многих факторов — качества исходного материала, режимов операции заварки и т. д.

С пультов статистического контроля 1 данные экспериментов поступают в управляющую мини-ЭВМ 2, в которой непрерывно строятся регрессионные модели и вырабатываются управляющие воздействия на технологический процесс (корректируется сила тока в катушке, выдаются различные рекомендации).

Математически процесс управления заваркой МК можно свести к следующей модели. Пусть  $\Theta_T$  и  $S_T$  — требуемые значения напряженности магнитного поля и зазора между контактами. Для решения задачи оптимизации сформулируем целевую функцию вида

$$F(\Theta, S) = \left(\frac{S_T - S}{S_T}\right)^2 + \left(\frac{\Theta_T - \Theta}{\Theta_T}\right)^2,$$

которую следует минимизировать с учетом ограничения  $\Theta = kS + b$ , определяемого уравнением регрессии.

Сформулированная задача является задачей квадратичного программирования, которую можно решить с использованием неопределенных множителей Лагранжа.

Функция Лагранжа имеет вид

$$F(\Theta, S, \lambda) = \left(\frac{S_T - S}{S_T}\right)^2 + \left(\frac{\Theta_T - \Theta}{\Theta_T}\right)^2 + \lambda(\Theta - kS - b).$$

Беря частные производные, получаем

$$\begin{aligned} \frac{\partial F}{\partial S} &= -2 \left( \frac{S_T - S}{S_T} \right) - k\lambda = 0; \quad \frac{\partial F}{\partial \Theta} = \\ &= -2 \left( \frac{\Theta_T - \Theta}{\Theta_T} \right) + \lambda = 0; \quad \frac{\partial F}{\partial \lambda} = \Theta - kS - b = 0. \end{aligned} \quad (6.58)$$

Решением системы будут значения  $S^*$  и  $\Theta^*$ , при этом  $\Theta^*$  косвенно определяет силу тока  $I$  в катушке.

Пусть  $\Theta_T = 40 \pm 2$  А/м;  $S_T = 10 \pm 2$  мкм;  $\Theta = 3S + 6$ ; число витков в катушке  $\omega = 50$ .

Решая систему уравнений (6.58), получаем  $\Theta^* = 40,44$  А/м,  $S^* = 10,48$  мкм.

Оптимальные значения  $\Theta$  и  $S$  находятся в поле допуска. Необходимый ток в катушке  $I = 0,8$  А. После установки  $I = 0,8$  А строят новую регрессионную зависимость и рассчитывают новое значение  $I$ . В результате многократной коррекции силы тока  $I$  находят такое значение, при котором процент выхода МК, отвечающих требованиям технических условий, будет максимальным.

**Пример 6.7.** Оптимизация управления технологической установкой. При конструкторском проектировании и при технологической подготовке производства узлов ЭВА для формирования графической информации используют приборы последовательного действия (ППД), в частности координатографы, сверлильные станки с числовым программным управлением и др.

Управление ППД производится подачей на их входы совокупности команд  $A$ , которые формируют изображение  $M$ . Требуется сформировать такую совокупность команд, которая обеспечивает функционирование ППД (получение изображения  $M$ ) с минимальным числом переходов от фрагмента к фрагменту.

Допустим, выработана последовательность команд  $A_s = (\alpha_1, \dots, \alpha_n)$  с индексом  $s$ . Тогда время работы ППД при формировании полного изображения  $M$  определяется соотношением

$$T^{(s)} = T_n^{(s)} + T_m = \sum_{i=1}^{n-1} t_{\alpha_i, \alpha_{i+1}} + \sum_{i=1}^n \Delta t_i,$$

где  $T_n^{(s)}$  — общее время переходов между фрагментами изображения;  $T_m$  — время выполнения команд ( $T_m = \text{const}$ );  $t_{\alpha_i, \alpha_{i+1}}$  — время перехода от команды  $\alpha_i$  к команде  $\alpha_{i+1}$ ;  $\Delta t_i$  — время формирования  $i$ -го фрагмента изображения  $M$ .

Стоимость формирования полного изображения  $M$  последовательностью команд  $A_s$  с учетом затрат на оптимизацию последовательности будет

$$C^{(s)} = C_{\text{опт}}^{(s)} + C_{\text{ППД}}^{(s)} = C_{\text{ЭВМ}} T_{\text{ЭВМ}}^{(s)} + C_{\text{ППД}} T^{(s)} q,$$

где  $C_{ЭВМ}$ ,  $C_{ППД}$  — соответственно удельная стоимость работы ЭВМ, производящей оптимизацию, и ППД;  $T_{ЭВМ}^{(s)}$  и  $T^{(s)}$  — соответственно времени работы ЭВМ и ППД;  $q$  — необходимое число копий изображения  $M$ .

Для оптимизации последовательностей  $A$  обычно выбирают такие методы, для которых

$$C_{\text{опт}}^{(s)} \ll C_{\text{ППД}}^{(s)}.$$

Таким образом, оптимизация последовательных команд  $A$  заключается в минимизации переходов  $T_{\pi}$ .

Для получения решения сформируем матрицу  $\|t_{ij}\|_{n \times n}$ , в которой элемент  $t_{ij}$  равен времени перехода от выполнения команды  $\alpha_i$  к команде  $\alpha_j$ .

Введем псевдодобулевы переменные

$$x_{ij} = \begin{cases} 1, & \text{если } i = \alpha_k, j = \alpha_{k+1}; \\ 0, & \text{если } i = \alpha_k, j \neq \alpha_{k+1} \end{cases}$$

и предположим, что последовательность команд образует замкнутый контур (начинается и заканчивается командой  $\alpha_i$ ). Тогда наша задача совпадает с классической задачей о коммивояжере

$$F(x) = \sum_{1 \leq i+j < n} t_{ij} x_{ij} \rightarrow \min$$

при ограничениях

$$\sum_{\substack{i=1 \\ i+j}}^n x_{ij} = 1, j = \overline{1, n}; \quad \sum_{\substack{j=1 \\ j+i}}^n x_{ij} = 1, i = \overline{1, n}, \quad (6.59)$$

$$u_i - u_j + n x_{ij} \leq n - 1, i, j = \overline{1, n}, \quad (6.60)$$

где  $u_i, u_j$  — произвольные действительные числа.

Условия (6.59) гарантируют непрерывность маршрута, а условие (6.60) — получение последовательности команд, не распадающейся на ряд замкнутых циклов.

Задача может быть решена методом ветвей и границ.

## 6.7. ОСОБЕННОСТИ РЕШЕНИЯ ЗАДАЧ СТРУКТУРНОГО СИНТЕЗА

Задача структурного синтеза заключается в выборе принципа действия технического объекта и в определении оптимальной структуры объекта для реализации заданных функций. Структура объектов определяется природой входящих в них элементов и физической реализацией связей между ними в составе объекта. Структуры наглядно изображают, как устроен объект — из каких частей он состоит и как эти части связаны друг с другом. Одной из форм

математического описания структуры является граф. Вершины графа отождествляются с элементами объекта (на принятом уровне детализации), а дуги и ребра графа — со связями между соответствующими элементами. Инженерная форма отображения структуры — схема. Схема и граф тождественны по своему содержанию, но различаются по форме. В схемах для обозначения элементов используют различные геометрические фигуры, разнообразие форм которых облегчает чтение схем.

Принцип, по которому объединение элементов приводит к появлению новых свойств, отличных от свойств элементов, называют *принципом организации*. Организация — понятие более высокого ранга, чем функция и структура. Различные принципы организации могут приводить к построению объектов, различающихся своими структурами и конструкциями, но тождественным по своему функциональному назначению.

*Сложность* — свойство объектов, заключающееся в том, что функция, реализуемая объектом, не может быть представлена в виде композиции функций, реализуемых элементами объекта. Например, при структурном синтезе ЭВМ рассматривается как система, состоящая из взаимосвязанных функциональных блоков и узлов, организованных таким образом, чтобы их функционирование приводило к реализации заданных функций — вычислениям на основе алгоритмов. Одни и те же функции могут быть реализованы различными структурами, обеспечивающими производительность решения задач при различных затратах оборудования. Закон функционирования ЭВМ невозможно рассмотреть только с точки зрения электрических процессов, происходящих в цепях ЭВМ. Функции ЭВМ выявляются лишь при рассмотрении процессов в ЭВМ в информационном и алгоритмическом аспектах. Это объясняется эффектом организации, порождающим в совокупностях элементов новые свойства.

Процесс проектирования структуры сложного технического объекта представляет собой многоэтапную процедуру, осуществляемую по блочно-иерархическому принципу:

Этап 1. Построение допустимого множества (каталога) принципиально возможных типов, входящих в объект элементов. Примерами могут служить: каталоги типовых конфигураций интегральных транзисторов в САПР больших интегральных схем; каталог допустимых микропроцессорных модулей, секций памяти и интерфейсного обрам-



ления в САПР микропроцессорных систем; каталог типовых проектов в САПР плитно-балочных разрезных мостов и др. Каталоги входят в состав информационного обеспечения САПР.

**Этап 2.** Выбор множества допустимых систем элементов, используемых при структурном синтезе объекта. Поскольку различные элементы, входящие в каталог применяемых элементов, могут быть реализованы на основе разнообразных физических принципов и с помощью различных технических решений, не все элементы могут стыковаться друг с другом. Поэтому основной задачей данного этапа является определение требований к параметрам и принципам функционирования отдельных элементов, входящих в состав проектируемого объекта.

**Этап 3.** Построение вариантов структур объекта с учетом их технической реализации. Здесь должны быть известны состав элементов, правила их соединения между собой и способ определения по структуре синтезируемого объекта функции, которую он реализует.

При структурном синтезе объектов небольшой сложности возможно построение полного множества допустимых структур объекта (для реализации полного перебора вариантов). При этом в ЭВМ должны быть заложены правила генерации всех вариантов структур проектируемого объекта.

При синтезе сложных объектов прямой перебор уже невозможен и необходима разработка процедур и алгоритмов направленного поиска оптимальной структуры синтезируемого объекта. Эти процедуры обычно базируются на использовании методов математического программирования (в основном — дискретного программирования), последовательных и итерационных алгоритмов синтеза, сетевых и графовых моделей проектирования, а также методов теории эвристических решений и методов решений изобретательских задач.

**Этап 4.** Оценка вариантов и выбор компромиссной структуры синтезируемого объекта. Как правило, оценка варианта структуры требует формирования и анализа математической модели синтезированной структуры объекта и выполнения параметрической оптимизации, так как для объективной оценки сравнивать варианты структуры имеет смысл при оптимальных значениях параметров. Эти процедуры сложны и громоздки, в связи с чем полный перебор вариантов при таком подходе практически неосуществим.

Для уменьшения сложности этого этапа целесообразно либо использовать косвенные критерии предпочтения вариантов, либо искать оценки варианта структуры без исследования громоздких математических моделей. При таком подходе вводят параметр, характеризующий качество объекта. Это может быть число элементов в объекте, его стоимость, занимаемый объем, максимальное число элементов, находящихся в активном состоянии (мощность), вероятность выхода из строя, максимальная длина проводников (в задачах размещения и трассировки) и т. д.

Другим путем уменьшения сложности решения задачи структурного синтеза является организация диалогового режима разработчика с ЭВМ на 3-м и 4-м этапах синтеза. При этом разработчик сам решает, какие программы анализа и оптимизации будет использовать для оценки вариантов. Сокращение времени на получение решения в диалоговом режиме происходит за счет эвристических способностей человека, за счет возможности прерывания построения заведомо бесперспективного варианта структуры и за счет поиска не оптимального, а допустимого варианта синтезируемого объекта.

Этап 5. Коррекция технического задания. Под коррекцией ТЗ понимают изменения заданных ограничений в тех случаях, когда не существует вариантов проектируемого объекта, обеспечивающих требуемое ограничение. Если коррекция произведена, то соответствующие этапы проектирования повторяют при новых значениях ограничений.

Объем решаемых на каждом этапе задач настолько велик, что проведение исследований в полном объеме невозможно без средств автоматизации проектирования. При этом на различных этапах к разработкам привлекаются различные специализированные коллективы научных работников и инженеров. Следует отметить, что при проектировании отдельных элементов, устройств и подсистем, входящих в состав проектируемого сложного объекта, следует проводить их анализ и синтез на различных уровнях. В процессе проектирования приходится учитывать существующие многократные перекрестные связи между элементами, что существенно усложняет задачу структурного синтеза.

Исходя из вышесказанного следует, что особенность задач структурного синтеза заключается в том, что для получения оптимального варианта структуры проектируемого объекта необходимо наличие его математической модели,

представляющей собой формальное описание множества структур объекта на принятом уровне детализации. В этом случае задача структурного синтеза сводится к выбору компромиссного варианта в счетном множестве.

#### 6.8. МОДЕЛИ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ В СТРУКТУРНОМ СИНТЕЗЕ ТЕХНИЧЕСКИХ ОБЪЕКТОВ

Из множества формализуемых задач структурного синтеза технических объектов значительная их часть может быть сведена к определению экстремального значения целевой функции

$$F(X) = \sum_{j=1}^m c_j x_j \rightarrow \max (\min) \quad (6.61)$$

при ограничениях

$$\sum_{j=1}^m a_{ij} x_j \leq b_i \left( \sum_{j=1}^m a_{ij} x_j \geq b_i \right), \quad i = \overline{1, n}, \quad (6.62)$$

$$x_j \geq 0, \quad j = \overline{1, m}; \quad (6.63)$$

$$x_j \text{ — целые числа, } j = \overline{1, p} \quad (p \leq m). \quad (6.64)$$

Если в сформулированной задаче ограничения (6.64) отсутствуют, то имеет место классическая задача линейного программирования, если ограничения (6.64) имеются и  $p = m$ , то данная задача является полностью целочисленной, при  $p < m$  задача является частично целочисленной.

**Задача линейного программирования.** В настоящее время теория линейного программирования хорошо разработана и имеется целый арсенал методов решения задач линейного программирования — это, например симплекс-метод, реализующий последовательную процедуру направленного поиска оптимального значения целевой функции (6.61) при существующих ограничениях вида (6.62) и (6.63).

**Симплекс-метод.** Первоначально система неравенств (6.62) путем введения дополнительных переменных  $x_{m+i} \geq 0$  преобразуется в систему уравнений таким образом, чтобы имело место одно из двух выражений

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{im} x_m + x_{m+i} = b_i;$$

$$a_{i1} x_1 + a_{i2} x_2 + \dots + a_{im} x_m - x_{m+i} = b_i, \quad i = \overline{1, n}.$$

Такое изменение приводит просто к увеличению числа переменных, что не меняет существа задачи. Введем ряд

понятий, широко применяемых в задачах линейного программирования.

*Базисом* называют любой набор из  $n$  таких переменных, что определитель, составленный из коэффициентов при этих переменных, не равен нулю. Отметим, что остальные  $m-n$  переменных называют *свободными*.

Не уменьшая общности, рассмотрим сущность симплекс-метода на примере задачи максимизации целевой функции (6.61) при наличии ограничений (6.62) и (6.63). Для определения первоначального базисного решения какие-либо  $m-n$  переменных принимают за свободные, т. е. приравняют нулю, при этом все базисные переменные выражают через свободные, после чего решают систему полученных уравнений. Если некоторые из базисных переменных окажутся отрицательными, то полученное базисное решение является *недопустимым* и производится переход к новому базису путем выбора новой совокупности свободных переменных. Базисное решение, в котором отсутствуют отрицательные переменные, называют *допустимым*.

После того как найдено допустимое базисное решение, проверяют, не достигнут ли максимум целевой функции  $F(X)$ . Если нет, то ищут новое допустимое базисное решение, но не любое, а такое, которое увеличивает значение целевой функции  $F(X)$ . Затем процедуру повторяют. Данный метод довольно быстро приводит к цели, так как позволяет исключить из рассмотрения большое число базисных решений, заведомо не обращающих в максимум целевую функцию  $F(X)$ .

Проверку того, не достигнут ли при найденном решении максимум целевой функции, можно сделать путем поиска нового базисного решения, при котором значение целевой функции  $F(X)$  будет больше предыдущего. Для прихода к новому допустимому базисному решению одну из свободных переменных следует сделать базисной, при этом она будет отличной от нуля, т. е. возрастет. Следовательно, если какая-либо из свободных переменных входит в выражение для целевой функции со знаком «+», а значит, при ее увеличении целевая функция увеличивается, то максимум целевой функции не достигнут и данную свободную переменную следует перевести в базисную.

Однако при возрастании свободной переменной некоторые из базисных переменных начнут уменьшаться. Так как отрицательные значения переменных недопустимы, в

качестве новой свободной переменной следует взять ту из базисных, которая раньше других обратится в нуль.

**Задачи целочисленного программирования.** В общем случае условие целочисленности накладывает дополнительные ограничения, вследствие которых максимальное значение целевой функции (в задачах максимизации) оказывается, как правило, меньше максимального значения целевой функции соответствующей задачи линейного программирования; в последней отсутствуют условия целочисленности переменных.

Рассмотрим два основных подхода к отысканию точного оптимального решения задач целочисленного программирования, базирующихся на методах отсекающих плоскостей и методах возврата.

Методы отсекающих плоскостей (методы отсечения). Исходным моментом решения задачи целочисленного программирования является оптимальное решение соответствующей задачи линейного программирования, полученной после отбрасывания условий целочисленности. На каждой итерации добавляется линейное ограничение, удовлетворяющее целочисленному решению исходной задачи, но исключающее текущее нецелочисленное решение. Вычислительный процесс прекращается, как только будет достигнуто любое целочисленное решение. Сходимость обеспечивается за конечное, но иногда очень большое число итераций.

Сущность алгоритмов, основанных на методе отсечения, легко уяснить, обратившись к геометрическим представлениям в пространстве решений (см. § 6.1). Определим выпуклую оболочку множества допустимых целочисленных точек (решений) как минимальное выпуклое множество, содержащее все эти точки. Допустимыми решениями будет не вся область допустимых решений, находящаяся внутри и на границе выпуклой оболочки, а лишь отдельные дискретные точки этой области, имеющие все целочисленные координаты. Целевая функция достигает оптимального значения в одной из вершин этой выпуклой оболочки, которая представляет собой одно из допустимых целочисленных решений.

Выпуклую оболочку можно представить конечным множеством линейных ограничений (6.62), как изображено на рис. 6.10. Можно, не считаясь с условиями целочисленности, найти решение, определяемое точкой 1, а затем, округлив это решение до ближайших целых чисел, получить цело-

численное решение в точке 2. Однако при этом может получиться решение, далекое от оптимального. Оптимальным целочисленным решением будет точка 3.

Для определения оптимального решения в алгоритмах отсечения вначале рассмотрим выпуклую оболочку, определенную линейными ограничениями (6.62) и условиями неотрицательности переменных исходной задачи, и отыщем экстремальную точку этой оболочки (точка 1 на рис. 6.10). Если такое решение оказывается нецелочисленным, то добавляют ограничение, отсекающее текущую экстремальную точку и уменьшающее «объем» выпуклой оболочки (прямая  $A-A$ ). Однако новое ограничение не отсекает ни одной экстремальной точки выпуклой оболочки, принадлежащей допустимым целочисленным решениям. В конечном итоге вводится такое число дополнительных ограничений, что экстремальная точка усеченной выпуклой оболочки представляет собой целочисленное решение исходной задачи.

Покажем, как формально построить дополнительные линейные ограничения, которым должно удовлетворить любое решение задачи (6.61) — (6.64). Рассмотрим ограничение

$$\sum_{j=1}^m a_{ij} x_j = b_i. \quad (6.65)$$

Возможно, одна или несколько величин  $a_{ij}$  и  $b_i$  являются дробными. Обозначим символом  $[a_{ij}]$  целую часть  $a_{ij}$ , т. е. наибольшее целое число, меньшее действительного числа  $a_{ij}$  или равное ему. Поскольку на величины  $x_j$  наложено ограничение (6.63), любые значения  $x_j$ , соответствующие условию (6.65), должны удовлетворять более слабому ограничению

$$\sum_{j=1}^m [a_{ij}] x_j \leq b_i. \quad (6.66)$$

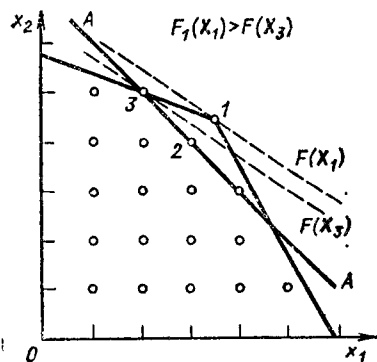


Рис. 6.10. Геометрическая иллюстрация принципа отсечения

Поскольку сумма в левой части неравенства (6.66) должна быть целочисленной, (6.66) можно усилить следующим образом:

$$\sum_{j=1}^m [a_{ij}] x_j \leq [b_i]. \quad (6.67)$$

Наконец, неравенство (6.67) можно преобразовать в равенство, добавив дополнительную переменную  $x_{m+i}$ :

$$\sum_{j=1}^m [a_{ij}] x_j + x_{m+i} = [b_i], \quad (6.68)$$

где  $x_{m+i}$  — целое, положительное.

Таким образом, если в исходную задачу добавить линейное ограничение (6.68), задача все равно останется полностью целочисленной.

Уравнение отсекающей плоскости получается в результате вычитания из (6.68) уравнения (6.65):

$$\sum_{j=1}^m -f(a_{ij}) x_j + x_{m+i} = -f(b_i), \quad (6.69)$$

где  $f(a_{ij})$ ,  $f(b_i)$  — дробная часть чисел  $a_{ij}$  и  $b_i$ .

Заметим, что текущее решение задачи линейного программирования не удовлетворяет ограничению (6.69), поскольку значение  $x_{m+i} = -f(b_i)$  строго отрицательно.

Алгоритм отсечения (алгоритм Гомори) состоит из следующих шагов.

1. Найти оптимальное решение задачи линейного программирования (6.61) — (6.63) без условия целочисленности (6.64).

2. Прекратить вычисления, если текущее решение задачи является целочисленным. В противном случае выбрать какую-либо дробную базисную переменную. Составить ограничение (6.69) из уравнения, содержащего эту базисную переменную в текущем оптимальном решении задачи линейного программирования.

3. Добавить к исходной задаче линейного программирования новое ограничение (6.69), найти оптимальное решение задачи с дополнительным ограничением и вернуться к шагу 2.

Введение на шаге 3 отсекающего ограничения (6.69) наряду с условием  $x_{m+i} > 0$  делает текущее решение задачи линейного программирования недопустимым. Отсечение

текущего оптимального решения означает, что на шаге 3 значение целевой функции в задаче максимизации не может возрасти, а может лишь уменьшиться.

Отметим, что алгоритмы отсечения не гарантируют получения допустимого целочисленного решения до самой последней итерации. Вследствие этого при преждевременном прекращении вычислений можно допустимое решение и не получить.

**Методы возврата.** В этой группе методов имеются различные модификации. Наиболее распространенным среди них является *метод ветвей и границ*, который предназначен для решения частично целочисленных задач. Как и в методе отсечения, решение задачи начинается с отыскания оптимального решения задачи линейного программирования без учета условия целочисленности. Затем формируется семейство связанных, но различных задач линейного программирования. Термин «возврат» определяет специфический способ формирования и решения последовательности задач.

Рассмотрим задачу максимизации (6.61) — (6.64). Допустим, что для каждой целочисленной переменной можно задать верхнюю и нижнюю границы, в пределах которых безусловно содержатся ее оптимальные значения

$$L_j \leq x_j \leq U_j, \quad j = \overline{1, p}. \quad (6.70)$$

Идея метода ветвей и границ основана на следующем элементарном факте. Рассмотрим любую переменную  $x_j$  и примем, что  $R$  есть некоторое целое число, где  $L_j \leq R \leq U_{j-1}$ . Тогда оптимальное решение задачи будет удовлетворять одному из ограничений

$$x_j \geq R + 1 \text{ или } x_j \leq R. \quad (6.71)$$

К примеру, если при решении задачи линейного программирования получено  $x_1 = 2,6$ , то можно поставить и решить две задачи линейного программирования, причем в одну из них вводится согласно (6.71) условие  $3 \leq x_1 \leq U_1$ , а в другую — условие  $L_1 \leq x_1 \leq 2$ . Предположим, что каждая из этих задач имеет оптимальное решение, удовлетворяющее условию целочисленности (6.64). Тогда решение, доставляющее большее значение целевой функции, является оптимальным решением исходной целочисленной задачи.

Метод ветвей и границ основан на решении некоторого множества задач линейного программирования. Границы



(6.70) на каждую переменную  $x_j$  служат для конкретизации диапазона изменений переменных в задаче целочисленного программирования, что в свою очередь определяет мощность множества задач линейного программирования, используемых в процедуре реализации метода ветвей и границ. Поэтому трудоемкость вычислений определяется числом целочисленных переменных, содержащихся в задаче.

Алгоритм решения задачи целочисленного программирования методом ветвей и границ заключается в следующем. На каждой итерации (обозначим номер итерации через  $t$ ) имеются нижняя оценка  $F_t^*(X)$  оптимального значения целевой функции и список задач линейного программирования, подлежащих решению. Процедура решения состоит в последовательном улучшении оценки  $F_t^*(X)$  и приближении ее к оптимальному значению  $F_{\text{opt}}(X)$ .

На итерации 1 список задач содержит одну задачу (6.61) — (6.63). На итерации  $t$  из списка выбирают и решают задачу линейного программирования. Если она не имеет допустимого решения или если полученное оптимальное значение целевой функции  $F_{t \text{ opt}}(X) \leq F_t^*(X)$ , то нижняя оценка остается прежней и из списка выбирают очередную задачу для решения. Если полученное решение удовлетворяет условию целочисленности (6.64) и  $F_{t \text{ opt}}(X) > F_t^*(X)$ , то полученное оптимальное решение  $F_{t \text{ opt}}(X)$  на итерации  $t$  принимают в качестве нижней оценки для последующих итераций. Если полученное оптимальное решение задачи линейного программирования не удовлетворяет условиям целочисленности (6.64), то выбирают нецелочисленную переменную  $x_j$  и решаемую задачу разбивают на две новые задачи линейного программирования путем введения в каждую из них по одному ограничению (6.71).

При остановке алгоритма в случае, если допустимому решению соответствует значение целевой функции  $F_{t \text{ opt}}(X) = F_t^*(X)$ , полученное решение оптимально, в противном случае допустимого решения не существует. Метод ветвей и границ особенно эффективен для решения комбинаторных задач, в частности задачи коммивояжера.

В общем виде процедура реализации этого метода заключается в следующем (рассмотрим ее применительно к задаче минимизации). Предположим, что имеется возможность получить нижнюю оценку качества решения  $F_v^*(X)$

как для всего множества  $V$  возможных решений, так и для его различных подмножеств. Разобьем множество на два непересекающихся подмножества  $A$  и  $B$  с точными нижними границами критерия качества  $F_{Aopt}(X)$  и  $F_{Bopt}(X)$ , связанных с  $F^*(X)$  очевидным соотношением

$$F_{opt}(X) = \min \{F_{Aopt}(X), F_{Bopt}(X)\}.$$

Поскольку множества  $A$  и  $B$  имеют меньшее число элементов, чем множество  $V$ , т. е. возможность получить нижние оценки качества  $F_A^*(X)$  и  $F_B^*(X)$  более близкие к  $F_{Aopt}(X)$  и  $F_{Bopt}(X)$ , чем в первоначальном множестве  $V$ , это означает, что оценка  $\min\{F_A^*(X), F_B^*(X)\}$  будет более близка к  $F_{opt}(X)$ , чем оценка  $F_V^*(X)$ .

Можно предположить, что оптимальное решение будет с большой вероятностью принадлежать тому из подмножеств  $A$  и  $B$ , которое имеет меньшую нижнюю оценку. Если оказалось, что  $F_B^*(X) < F_A^*(X)$ , то есть все основания для детального исследования подмножества  $B$ . Последнее разбивают на два подмножества  $C$  и  $D$  с соответствующими нижними оценками  $F_C^*(X)$  и  $F_D^*(X)$ , и если оказалось, что  $F_D^*(X) < F_C^*(X)$ , то подобному разбиению подвергается подмножество  $D$ . Эту процедуру продолжают до тех пор, пока не придут к подмножеству, состоящему из одного элемента с оценкой  $F^*(X) = F_0(X)$ . Процедура разбиения обычно представляется в виде дерева решений (рис. 6.11).

Найденное решение необходимо проверить на оптимальность. Для этого проверяют, нет ли среди нерассмотренных множеств  $A$ ,  $C$  и т. д. элемента со значением нижней оценки  $F^*(X) < F_0(X)$ . При этом множества, у которых  $F^*(X) \geq F_0(X)$ , дальше не рассматриваются.

Разбиение перспективных множеств либо приведет к

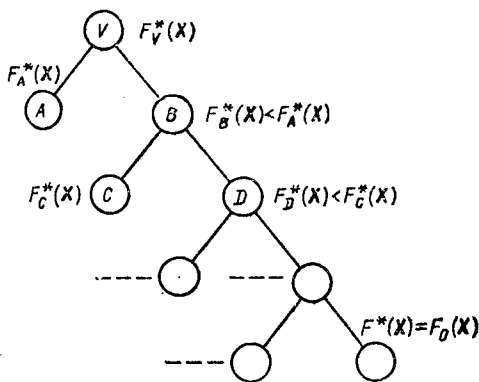


Рис. 6.11. Геометрическая иллюстрация метода ветвей и границ

решению с оценкой  $F^*(X) < F_0(X)$ , которое должно быть принято за оптимальное, либо позволит убедиться в оптимальности полученного ранее решения с оценкой  $F_0(X)$ . Из процедуры реализации метода ветвей и границ становится ясным, почему этот метод получил название метода возврата.

Метод ветвей и границ наряду с методами отсечения обладает существенными достоинствами с вычислительной точки зрения. Алгоритмы, построенные на этих методах, сравнительно легко программируются на ЭВМ и реализуются на любой итерации без вмешательства человека, однако их эффективность резко снижается при увеличении размерности решаемой задачи.

Применение моделей и методов математического программирования при конструировании технических объектов было рассмотрено в примерах § 6.2. Ниже приводятся примеры постановки типовых задач структурного синтеза в терминах математического программирования.

**Пример 6.8.** Оптимизация структуры сети электросвязи. Процесс проектирования региональных сетей электросвязи состоит из ряда взаимосвязанных этапов:

построения структуры первичной сети связи, обеспечивающей связь между заданными пунктами региона по критерию минимальной стоимости;

корректировки полученной структуры сети путем введения дополнительных связей между пунктами с целью получения живучей сети электросвязи;

построения вторичной сети связи, заключающейся в определении конфигурации сетки пучков каналов электросвязи для обеспечения необходимой пропускной способности каналов между различными пунктами региона.

Формально задачу синтеза структуры первичной сети связи можно представить в виде следующей задачи математического программирования. Задана матрица расстояний  $D = |d_{ij}|$  размерности  $n \times n$  между всеми  $n$  пунктами данного региона. Необходимо определить такую структуру сети, которая обеспечивала бы связь между всеми пунктами региона по критерию минимальной стоимости. При этом будем считать, что стоимость канала связи между пунктами  $i$  и  $j$  пропорциональна расстоянию  $d_{ij}$  между ними.

Введем псевдобулевы переменные:

$$x_{ij} = \begin{cases} 1, & \text{если имеется прямой канал связи между пунктами } i \text{ и } j; \\ 0 & \text{в противном случае.} \end{cases}$$

Задача синтеза оптимальной структуры сети связей по критерию,

минимальной стоимости заключается в определении таких переменных  $x_{ij} \in \{0, 1\}$ , которые обращали бы в минимум целевую функцию

$$F(X) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}$$

при ограничениях

$$\sum_{i=1}^n x_{ij} \geq 1, \sum_{j=1}^n x_{ij} \geq 1, \sum_{i=1}^n \sum_{j=1}^n x_{ij} \geq n-1.$$

Решение сформулированной задачи математического программирования удобно определять методом ветвей и границ.

**Пример 6.9.** Задача синтеза структуры памяти специализированной ЭВМ. Требуется синтезировать структуру внутренней памяти (СОЗУ—ОЗУ—ПЗУ) специализированной ЭВМ, работающей в режиме реального времени, совместно с определением необходимого пакета объектных программ и их размещением в различных блоках системы памяти.

Пусть ЭВМ ориентирована на решение множества задач (заявок)  $C = \{c_1, \dots, c_n\}$ , причем каждая задача  $c_i \in C$  встречается с относительной частотой  $g_i$ , и пусть на множестве  $C$  задано его разбиение на подмножества  $C_l, l = \overline{1, u}, (u \leq n)$ , так что в подмножество  $C_l \subset C$  входят только те задачи  $c \in C_l$ , которые имеют  $l$ -й приоритет, причем чем меньше  $l$ , тем выше приоритет.

Для решения поступающих в ЭВМ задач имеется множество алгоритмов  $\tilde{L} = \{L_1, \dots, L_p\}, p \leq n$ , по отношению к которым множество  $C$  разбито на подмножества  $C(L_i) \subseteq C$  таким образом, что если  $c \in C(L_i)$ , то задача может быть решена с помощью алгоритма  $L_i \in \tilde{L}$ . Каждый из алгоритмов  $L_i \in \tilde{L}$  может быть реализован  $m_i$  методами, т. е. для каждой  $c_i$  можно записать множество упорядоченных пар  $L_i = \{(i, j)\}, i = \overline{1, p}, j = \overline{1, m_i}$ , где любая пара  $(i, j)$  описывает метод решения задачи  $c_i$ . Каждый из методов  $(i, j) \in L_i$  характеризуется: способом хранения программы (в ОЗУ или ПЗУ); числом требующихся ячеек  $a_{ij}$  ОЗУ или ПЗУ для записи программ; числом  $b_{ij}$  ячеек СОЗУ для хранения часто встречающихся данных и промежуточных результатов и числом  $h_{ij}$  ячеек ОЗУ для хранения исходных данных при обработке заявки  $c_i$  в реальном времени; количеством  $r_{ij}^{(k)}$  ресурса  $k$ , где под ресурсами будем понимать габариты, массу, стоимость, число используемых ячеек или панелей, потребляемую мощность и т. д. Поскольку программы могут храниться как в ПЗУ, так и в ОЗУ, это влияет на время реализации программы, надежность и оперативность в смене программ. Часть параметров, например, время решения задачи, стоимость, габариты, потребляемая мощность и др., подчиняется условиям аддитивности, в то время как такие параметры, как надежность, диагностическая возможность, ремонтпригодность и др., являются неаддитивными.

Задача заключается в определении структуры внутренней памяти и такого набора программ  $\Pi$  решения поступающих задач, чтобы удовлетворялись все ограничения на параметры ЭВМ, а выбранный критерий оптимальности достигал своего экстремального значения.

В частности, в качестве критерия оптимальности можно выбрать суммарный объем ОЗУ и ПЗУ, который требуется для хранения программ и данных и который следует минимизировать.

Дадим математическое описание задачи. Введем псевдодобулевы переменные:

$$x_{ij} = \begin{cases} 1, & \text{если алгоритм } L_i \in \tilde{L} \text{ обработки заявки } c_i \in C \text{ реализуется} \\ & \text{j-й программой;} \\ 0 & \text{в противном случае.} \end{cases}$$

Для ЭВМ, работающих в реальном времени (время реализации алгоритмов строго ограничено и задаче соответствует свой алгоритм решения  $L_i \in \tilde{L}$ , ( $p=n$ ), задачу выбора набора программ для реализации множества алгоритмов  $\tilde{L}$  можно сформулировать следующим образом (при этом заменим соответствующие обозначения  $L_i$  на индекс  $i$ ): минимизировать целевую функцию

$$\Phi(X) = \sum_{i=1}^n \sum_{j=1}^{m_i} (a_{ij} + h_{ij}) x_{ij} \quad (6.72)$$

при временных ограничениях на реализацию алгоритмов

$$\begin{aligned} \sum_{i \in C_1} \sum_{j=1}^{m_i} \bar{t}_{ij} g_i x_{ij} &< T_1; \\ \sum_{i \in C_1} \sum_{j=1}^{m_i} \bar{t}_{ij} g_i x_{ij} + \sum_{i \in C_2} \sum_{j=1}^{m_i} \bar{t}_{ij} g_i x_{ij} &< T_2; \\ &\dots \dots \dots \\ \sum_{i=1}^u \sum_{i \in C_i} \sum_{j=1}^{m_i} \bar{t}_{ij} g_i x_{ij} &< T_i; \end{aligned} \quad (6.73)$$

$$\begin{aligned} \max_{L_i \in L} \left\{ \sum_{j=1}^{m_i} b_{ij} x_{ij} \right\} &< B; \quad \sum_{i=1}^n \sum_{j=1}^{m_i} h_{ij} x_{ij} < H; \end{aligned} \quad (6.74)$$

при ограничениях на аддитивные параметры

$$\sum_{i=1}^n \sum_{j=1}^{m_i} r_{ij}^{(k)} x_{ij} < R_k, \quad k = \overline{1, q}; \quad (6.75)$$

при логических условиях существования единственной программы реализации каждого алгоритма

$$\sum_{j=1}^{m_i} x_{ij} = 1, \quad i = \overline{1, n}. \quad (6.76)$$

Здесь  $\overline{t}_{ij}$  — математическое ожидание времени реализации алгоритма  $L_i$   $j$ -м методом;  $g_i$  — относительная частота появления алгоритма в общей программе работы ЭВМ;  $R_k, B, H$  — ограничения на  $k$ -й ресурс, объем СОЗУ и ОЗУ соответственно;  $T_l$  — максимально допустимое время реализации задач, имеющих  $l$ -й приоритет.

Задача (6.72)—(6.76) также является задачей дискретного программирования с псевдодулевыми переменными. Подставляя в целевую функцию задачи оптимизации другие параметры, в частности стоимость, время решения задач, энергетические и другие параметры системы памяти, можно оптимизировать структуру системы памяти ЭВМ по соответствующим критериям.

Рассмотренная задача может быть использована для синтеза технических и программных средств САПР.

## 6.9. ПОСЛЕДОВАТЕЛЬНЫЕ МЕТОДЫ В ЗАДАЧАХ ПРОЕКТИРОВАНИЯ СЛОЖНЫХ ОБЪЕКТОВ

Выбор оптимального варианта структуры проектируемого объекта методами, базирующимися на полном переборе вариантов, является дорогостоящей, трудоемкой и, как правило, неосуществимой процедурой. Использование методов математического программирования для принятия решений в задачах структурного синтеза технических объектов требует большой предварительной подготовки для исследования пространства решений и не всегда оправдано из-за больших трудностей учета многочисленных факторов, влияющих на корректность постановки задачи оптимального проектирования, и из-за существенных вычислительных трудностей решения задач математического программирования большой размерности.

При проектировании сложных объектов довольно эффективными оказываются последовательные методы анализа и синтеза.

Последовательные методы анализа основаны на направленной генерации множества вариантов проектных решений и осуществлении процедуры анализа вариантов с целью выбора наилучшего путем последовательного отсеивания неперспективных вариантов. Наибольшее распространение в задачах проектирования получил метод последовательного анализа вариантов, развитый в работах академика В. С. Михалевича и основанный на обобщении идей

теории последовательных статистических решений  
А. Вальда.

Последовательные алгоритмы синтеза основаны на наращивании структуры путем добавления по определенным правилам элементов к некоторому начальному элементу.

**Метод последовательного конструирования, анализа и отсеивания вариантов.** В основе этого метода лежит идея процесса принятия решения в виде многоступенчатой структуры. Каждая ступень связана с проверкой наличия определенных свойств у подмножества вариантов и либо ведет к непосредственному сокращению исходного множества вариантов, либо подготавливает возможность такого сокращения в будущем. Для решения задачи необходимо определить отличительные свойства, которыми должен обладать искомый вариант. Первоначально из множества признаков выбирают наиболее легко проверяемые и присущие одновременно возможно большему числу вариантов. После этого выбор численной схемы решения состоит в выборе рационального порядка проверки признаков, позволяющего провести отсев неконкурентоспособных вариантов и найти оптимальный.

Алгоритмы последовательного анализа вариантов основаны на принципе оптимальности, который представляет собой естественное обобщение принципа оптимальности динамического программирования для решения многошаговых задач оптимизации.

Напомним, что в методе динамического программирования выбор решения (управления) на отдельном шаге производится не с точки зрения интересов данного шага, выражающихся в минимизации потерь на данном шаге, а с точки зрения всего многошагового процесса принятия решений в целом, выражающихся в минимизации суммарных потерь на всех последующих шагах. Отсюда следует основное свойство оптимального процесса принятия решений, заключающееся в том, что каковы бы ни были начальное состояние и начальное решение, последующие решения на каждом шаге должны быть оптимальными относительно состояния, являющегося результатом применения первого решения. Из этого свойства следует, что оптимизация выбора решения для многошагового процесса принятия решений заключается в выборе решений только на последующих шагах процесса.

Основное правило отсева бесперспективных вариантов — монотонная рекурсивность, идейно родственная

критерию оптимальности динамического программирования.

Пусть имеются множество  $W = \{\omega\}$  вариантов решения и множество опытов  $\Pi = \{\pi_\alpha\}$ . Каждый вариант  $\omega \in W$  описывается некоторым множеством признаков. Задача состоит в определении подмножества  $W^* \subseteq W$ , инвариантного относительно любого  $\pi_\alpha$  и содержащего оптимальное решение  $\omega_0 \in W^*$ . Для определения подмножества  $W^*$  необходимо поставить опыты по анализу и оценке свойств элементов  $\omega \in W$ . Исходы опытов позволяют отбросить неперспективные варианты  $\omega$ , которые не имеют общих частей с элементами подмножества  $W^*$ , и сделать заключение о целесообразности постановки последующих опытов с целью определения элементов, входящих в подмножество  $W^*$ .

Дадим описание правила отсева. Пусть задано некоторое базовое множество  $X$ . Обозначим множество конечных последовательностей вида

$$p = (x_1, \dots, x_i, \dots, x_{k_p}), \quad x_i \in X, \quad 1 \leq i \leq k_p,$$

через  $P(X)$ . В этом множестве выделено некоторое подмножество допустимых последовательностей  $W(X) \subseteq P(X)$ , в свою очередь во множестве  $W(X)$  выделено подмножество  $W_0(X) \subseteq W(X)$  полных допустимых последовательностей. Пусть задана последовательность  $p$ . Последовательность

$$p_l = (x_1, \dots, x_l), \quad 1 \leq l \leq k_p,$$

называют  $l$ -м начальным отрезком последовательности  $p$ , а последовательность

$$p^{(q)} = (x_q, x_{q+1}, \dots, x_{k_p}), \quad 1 \leq q \leq k_p,$$

называют  $q$ -м конечным отрезком. Если  $q = l + 1$ , то соответствующие части последовательности  $p$  называют сопряженными. Рассмотрим две допустимые последовательности  $p_1$  и  $p_2$ . В  $p_1$  выделены  $l_1$ -й начальный отрезок  $p_{1l_1}$  и  $(l_1 + 1)$ -й конечный отрезок  $p_{1l_1+1}$ , в  $p_2$  выделены  $l_2$ -й начальный отрезок  $p_{2l_2}$  и  $(l_2 + 1)$ -й конечный отрезок  $p_{2l_2+1}$ . Если функционал  $\Phi$ , определенный на множестве  $W(X)$ , обладает тем свойством, что из

$$p_{1l_1} \in W(X); \quad p_{2l_2} \in W(X); \quad p_{1l_1+1} \equiv p_{2l_2+1};$$

$$\Phi(p_{1l_1}) < \Phi(p_{2l_2})$$



следует  $\Phi(p_1) < \Phi(p_2)$ , то его называют монотонно-рекурсивным.

Вариант  $w_0 \in W^*$ , характеризующийся максимальным значением функционала  $\Phi$ , определяется согласно обобщенному принципу оптимальности: если заданы монотонно-рекурсивный функционал  $\Phi$  и две допустимые последовательности  $p_1$  и  $p_2$ , причем

$$\Phi(p_1) < \Phi(p_2), \quad P(p_1) \subseteq P(p_2),$$

то последовательности, у которых начальным является отрезок  $p_1$ , будут неперспективными и не входят в подмножество  $W^*$ . Здесь  $P(p_i)$  — множество конечных отрезков, сопряженных с последовательностью  $p_i$ .

Таким образом, схема последовательного выбора оптимального варианта сводится к следующим повторяющимся процедурам:

1. Множество вариантов решений  $W$  разбивают на несколько подмножеств, каждое из которых обладает специфическими свойствами.

2. На основании обобщенного принципа оптимальности отсеивают те варианты  $w \in W$ , значение функционала  $\Phi$  для которых заведомо не может быть максимальным.

3. Последовательно конструируются и анализируются не отсеянные ранее варианты решения.

4. Для множества, состоящего из вновь образованных в п. 3 допустимых последовательностей, и последовательностей не исключенных и не продолженных ранее, выполняются операции п. 2.

Далее операции п. 2, 3 и 4 циклически повторяются. Если на каком-то этапе процесса решения не останется ни одной последовательности, требующей своего развития до получения полной последовательности, то процесс завершен и в качестве решения берется одна из рассмотренных полных допустимых последовательностей с наибольшим значением функционала  $\Phi(p)$ .

Основным достоинством метода последовательного анализа вариантов является значительная экономия в вычислительной процедуре за счет отсеивания бесперспективных начальных частей вариантов до их полного построения. Экономия является тем существенней, чем больше определяющих свойств задачи использовано для построения процедур анализа и отсева.

**Последовательные алгоритмы структурного синтеза.** Последовательные алгоритмы синтеза относятся к классу

эвристических. Их достоинством является сравнительно высокая экономичность по затратам машинного времени и требуемому объему оперативной памяти за счет отсутствия процедуры многоразового анализа вариантов структуры. Однако, как правило, последовательные алгоритмы дают не оптимальные, а близкие к оптимальным решения.

Развитие диалоговых средств общения разработчика с ЭВМ инициировало широкое применение последовательных методов и алгоритмов в структурном синтезе технических объектов разнообразного назначения. В качестве иллюстрации рассмотрим идею формирования последовательных алгоритмов для решения задач конструкторского проектирования ЭВА — задач компоновки, размещения и трассировки.

При иерархической организации конструкции ЭВА под *компоновкой* понимают определение состава типовых конструкций каждого уровня. Задача компоновки обычно решается «снизу — вверх», т. е. известные схемы  $i-1$ -го уровня необходимо распределить по конструкциям  $i$ -го уровня. Так, например, на самом низшем уровне элементами могут выступать корпуса микросхем, а конструкциями (блоками) — типовые элементы замены, связанные друг с другом путем разъемных соединений.

В качестве критериев оптимальности при решении задач компоновки наиболее часто используют критерии либо минимума суммарного числа  $N_i$  типов модулей

$$N_i = \sum_j x_{ij},$$

где  $x_{ij}$  — число модулей  $j$ -го типа  $i$ -го уровня схемы, либо минимума межблочных соединений

$$R_i = 0,5 \sum_{k=1}^{N_i} R_{ik},$$

где  $R_{ik}$  — число внешних связей каждого модуля  $i$ -го уровня.

Первый критерий непосредственно связан с конструктивными характеристиками аппаратуры и показателем технологичности стоимости, второй критерий ведет к повышению надежности конструктивной реализации схемы за счет сокращения числа разъемных соединений, уменьшению помех и задержек сигналов благодаря снижению числа межблочных соединений.

Для иллюстрации рассмотрим принцип построения последовательных алгоритмов компоновки по критерию минимума межблочной связности. Этот критерий широко используется при компоновке оборудования и различных технических приложениях. Идея алгоритмов заключается в следующем. Первоначально выбирают исходный элемент (модуль) схемы. Выбор начального элемента основывается на схмотехнических соображениях.

В первый компонуемый узел включены все элементы, смежные с начальным, и сам начальный элемент. Если полученное число элементов равно максимально допустимому числу элементов в первом узле, то компоновка этого узла заканчивается. Если это число больше или меньше максимально допустимого, то выполняются соответствующие операции по устранению лишних или добавлению недостающих элементов, причем из нескомпонованных элементов выбирают такой, который имеет наибольшее число связей с элементами, уже вошедшими в состав компонуемого узла. Далее сформированный узел удаляют из схемы и компонуют новые узлы. Процесс повторяется до тех пор, пока схема не будет разбита на требуемое число частей или не будет выяснена невозможность этого.

Сформулируем описанный алгоритм в терминах теории графов. Пусть задан граф схемы  $G = (X, U)$ , который необходимо разбить на  $l$  частей  $G_1, G_2, \dots, G_l$  с числом вершин в каждом соответственно  $n_1, n_2, \dots, n_l$  ( $\sum_{i=1}^l n_i = n, |X| = n$ ).

Первоначально в графе  $G$  определяют вершину  $x_i \in X$  с наибольшей локальной степенью  $\rho(x_i)$  [напомним, что локальной степенью  $\rho(x_i)$  вершины  $x_i \in X$  называют число ребер, инцидентных этой вершине графа]. Если таких вершин несколько, то предпочтение отдается той, которая имеет большее число кратных ребер. Вершина  $x_i$  и все смежные с ней вершины включаются в граф  $G_1$ . Обозначим это множество вершин через  $\Gamma x_i$ . Если  $|\Gamma x_i| = n_1$ , то  $G_1$  образован, если же  $|\Gamma x_i| > n_1$ , то из графа  $G_1$  удаляют вершины, связанные с остающимися вершинами графа  $G$  меньшим числом ребер. Когда  $|\Gamma x_i| < n$ , то выбирают вершину  $x_j \in \Gamma x_i$ , удовлетворяющую условию

$$\sigma(x_j) = \max_{x_k \in \Gamma x_i} \{\sigma(x_k)\} = \max_{x_k \in \Gamma x_i} \{\rho(x_k) - a_k\},$$

где  $a_k$  — число ребер, соединяющих вершину  $x_k$  со всеми невыбранными вершинами графа  $G$ .

Строят множество вершин  $Gx_j$ , смежных  $x_j$ , и процесс выборки вершин  $G_1$  повторяют. Образованный подграф  $G_1$  исключают из исходного и получают граф  $G^* = (X^*, U^*)$ , где  $X^* = X \setminus X_1$ ,  $U^* = U \setminus U_1$ . Далее в графе  $G^*$  выбирают вершину с наибольшей локальной степенью, включают ее в  $G_2$  и процесс повторяют до тех пор, пока граф  $G$  не будет разрезан на  $l$  частей.

Первоначальную компоновку можно улучшить с помощью итерационных алгоритмов, основанных на реализации методов парных или групповых перестановок элементов из одной части схемы в другую таким образом, чтобы улучшилось значение целевой функции с учетом заданных ограничений.

Задача размещения заключается в определении оптимального (с точки зрения выбранного критерия оптимальности) положения элементов и связей между ними в монтажном пространстве типовой конструкции с учетом заданных конструктивно-технологических ограничений. Исходными данными в задаче решения являются принципиальная электрическая схема узла или устройства, метрические параметры и топологические свойства монтажного пространства.

Главная цель размещения — создание наилучших условий для трассировки с учетом обеспечения тепловых режимов и электромагнитной совместимости электрорадиоэлементов. Несмотря на обилие существующих критериев размещения (минимума пересечений, минимума суммарной длины соединений и т.д.) истинной целью размещения компонентов является максимальное упрощение процесса трассировки соединений, т.е. достижение минимального числа непроведенных трасс. При размещении  $n$  электрорадиоэлементов в регулярном монтажном пространстве с числом позиций  $m$  общее число размещений  $N(n, m)$  определяется как

$$N(n, m) = n! C_m^n = m! / (m - n)!$$

В связи с этим поиск оптимального размещения с помощью перебора нецелесообразен уже при  $n > 15$ .

Имеется много разновидностей последовательных алгоритмов размещения. Основной идеей этих алгоритмов является идея упорядочения электрорадиоэлементов по определенным признакам. Сначала устанавливают очередность электрорадиоэлементов, а затем для каждого из них определяют наилучшую позицию по выбранному критерию,

например по суммарной длине связей с уже размещенными компонентами. Затем процесс повторяют для оставшихся компонентов и свободных позиций. Связность размещаемых элементов задается матрицей смежности  $R$  графа  $G = (X, U)$ . Для выбора размещаемого элемента используют различные оценки степени связности.

Пусть на  $k$ -м шаге алгоритма размещено  $I_k \subset I$  элементов, тогда  $I'_k = I \setminus I_k$  — множество еще не размещенных элементов. Основными правилами для выбора элемента на  $(k+1)$ -м шаге алгоритма являются:

а) максимум суммарной связности  $h_i$  со всеми размещенными элементами

$$h_i = \max_{i \in I'_k} \left\{ \sum_{j \in I_k} r_{ij} \right\}, \quad i \neq j;$$

б) максимум разности связей  $f_i$  между размещенными и неразмещенными элементами

$$f_i = \max_{i \in I'_k} \left\{ \sum_{j \in I_k} r_{ij} - \sum_{j \in I'_k} r_{ij} \right\}, \quad i \neq j.$$

Выбранный для размещения элемент устанавливают в такую позицию среди оставшихся незаполненных, при которой будет иметь наименьшее значение некоторая целевая функция. Для многих задач размещения в качестве такой функции может быть выбрана суммарная длина связей с уже размещенными элементами.

Последовательные алгоритмы размещения требуют небольших затрат машинного времени, относят их к классу полиномиальных алгоритмов со сложностью  $O(n)$ , приводящих к неоптимальным решениям. Улучшить решение можно путем применения итерационных алгоритмов компоновки, основанных на изменении позиций одиночных элементов или групп элементов. Итерационные алгоритмы также относятся к классу полиномиальных со сложностью порядка  $O(n^2) - O(n^4)$ .

Задача *трассировки* заключается в определении конкретной геометрии печатного или проводного монтажа, реализующего соединения между элементами схемы. Исходными данными для трассировки являются список цепей, метрические параметры и топологические свойства типовой конструкции и ее элементов, а также результаты решения задачи размещения, по которым находят координаты выводов элементов.

При решении задачи трассировки строят множество трасс, соединяющих выводы элементов соответствующих цепей схемы. Разработка отдельной трассы представляет собой построение на фиксированных вершинах минимального покрывающего или связывающего дерева, а разработка множества трасс сводится к построению леса непесекающихся минимально покрывающих или связывающих деревьев. Известно, что на  $n$  вершинах можно построить  $n^{n-2}$  различных деревьев, поэтому точное решение задачи трассировки методом полного перебора практически нереализуемо.

В последовательных алгоритмах трассировки трассы цепей проводятся в определенном порядке одна за другой, при этом каждая проложенная трасса становится препятствием для всех последующих цепей. В последовательных алгоритмах производят локальную оптимизацию качества трассировки каждой отдельной трассы без учета влияния размещения данной трассы на возможность проведения последующих. Это приводит к тому, что некоторые участки платы могут оказаться заблокированными.

Большинство известных алгоритмов трассировки основывается на волновом алгоритме (алгоритм Ли). Основные принципы волнового алгоритма Ли заключаются в следующем. Плоскость трассировки разбивают на прямоугольные площадки — дискретные заданного размера. Размер дискретной площадки определяется допустимыми размерами проводников и расстояниями между ними. Задача проведения трасс сводится к получению последовательности дискретов, соединяющих элементы  $a$  и  $b$ , соответствующие началу и концу проводимой трассы.

Вводим целевую функцию  $F = F(f_1, \dots, f_r)$  как критерий качества пути. Начиная с элемента  $a$  дискретам, соседним с ранее просмотренным, присваивают определенное значение целевой функции  $F_{ij} = m(i, j)$ . Этот этап проводится итерационно до элемента  $b$ , которому присваивают некоторое значение веса  $m(i_b, j_b)$ . Затем, начиная от элемента  $b$ , перемещаются к элементу  $a$  по пройденным дискретам таким образом, чтобы значения целевых функций дискретов монотонно убывали. В результате получается трасса, соединяющая элементы  $a$  и  $b$ .

Обычно работа алгоритма Ли реализуется следующим образом. На трассируемой плоскости из источника  $a$  моделируется распространение волны до тех пор, пока не будет достигнута точка  $b$  или пока на некотором шаге фронт

волны не сможет включить ни одного незанятого дискрета. Эту часть алгоритма называют распространением волны. После этого проводят трассу, начиная от конечной точки  $b$ , по дискретам с последовательно уменьшающимися весами. На рис. 6.12 цифры в квадратах соответствуют весам

8	9	10	11	10	11	$b$	13
7	///	9	8	9	///	///	12
6	///	8	7	8	9	10	11
5	///	5	6	7	///	///	///
4	3	4	5	6	7		
3	2	3	///		6		
2	1	///	///	///	5	6	7
1	$a$	1	2	3	4	5	6

Рис. 6.12. Пример соединения элементов  $a$  и  $b$  с помощью волнового алгоритма

дискретов, занятые дискреты заштрихованы, а построенная трасса показана штриховой линией. Существует несколько вариантов проведения пути, из которых конструктор (или ЭВМ) выбирает один, наиболее удовлетворяющий заданным требованиям. Имеется многообразие модификаций волновых алгоритмов, направленных на повышение быстродействия трассировки, уменьшение объема требуемой оперативной памяти ЭВМ и т. д.

Волновые алгоритмы широко применяют в различных областях техники, в частности при разработке сетей связи и сетей ЭВМ.

### Контрольные вопросы и задания

1. Сформулируйте задачу синтеза технического объекта.
2. В каких случаях многокритериальные задачи оптимизации могут быть сведены к однокритериальным?
3. Что такое «модель проектируемого объекта»? Какие математические задачи можно использовать в качестве моделей проектируемых объектов?
4. Объясните сущность диалектического единства и противоположности структурного и параметрического синтеза технических объектов.
5. Разработайте математическую модель распределения оборудования по производственным цехам по критерию минимума межоперационных транспортных расходов.
6. Разработайте математическую модель проектирования линий электропередачи.
7. Сформулируйте задачу оптимизации размещения цехов по территории завода по критерию минимума транспортных межцеховых расходов.

8. Почему в задачах параметрического синтеза необходимо свертывание векторного критерия?

9. Приведите примеры параметрической оптимизации при синтезе усилителей низкой частоты, при синтезе четырехполюсников, при проектировании строительных конструкций.

10. Сформулируйте необходимые и достаточные условия экстремума непрерывных функций.

11. Объясните различие между экстремальными и седловыми точками в задачах оптимизации.

12. Дайте классификацию методов поиска экстремума.

13. Что понимается под эффективностью методов поиска оптимума?

14. Сформулируйте принципы прямых методов оценки направлений в задачах безусловной оптимизации.

15. В чем заключаются достоинства и недостатки градиентных методов поиска экстремума?

16. Оцените эффективность методов одномерной оптимизации.

17. Объясните общность и различие методов штрафных и барьерных функций в задачах условной оптимизации.

18. На любом техническом примере объясните физическую сущность области работоспособности в пространстве параметров и в оптимальной допусковой области.

19. В чем сущность задачи вписывания гиперпараллелепипеда допусков в область работоспособности?

20. Назовите особенности задачи оптимизации технологического процесса.

21. Почему при оптимизации технологических процессов для оценки параметров используются регрессия и корреляция?

22. В задаче оптимизации структуры сети электросвязи (пример 6.7) матрица

$$D = \begin{bmatrix} 0 & 3 & 6 & 2 & 1 \\ 2 & 0 & 1 & 3 & 7 \\ 6 & 2 & 0 & 4 & 3 \\ 4 & 2 & 4 & 0 & 5 \\ 2 & 6 & 1 & 6 & 0 \end{bmatrix}.$$

Решите задачу оптимизации методом ветвей и границ.



## ВОПРОСЫ РАЗРАБОТКИ САПР

## 7.1. РАЗРАБОТКА ТЕХНИЧЕСКОГО ОБЕСПЕЧЕНИЯ САПР

**Задача разработки технического обеспечения САПР.** Разработка САПР представляет собой комплекс взаимосвязанных работ по созданию математического, программного, технического, информационного и других видов обеспечения систем, ориентированных на автоматизированное проектирование определенного класса объектов (САПР машиностроения, самолетостроения, БИС, ЭВМ и др.).

В разработке и внедрении САПР принимают участие большие коллективы проектных и конструкторско-технологических организаций, усилия которых координируются группой системных исследователей.

Принципы организации и стадии разработки САПР регламентированы руководящими и методическими материалами, а также государственными стандартами и подробно описаны в гл. 1.

В настоящей главе рассмотрим некоторые специфичные аспекты разработки технического обеспечения САПР (ТО САПР). К ТО САПР предъявляются требования возможности организации оперативного взаимодействия проектировщиков с ЭВМ, достаточной производительности вычислительных средств и необходимого объема оперативной памяти для решения задач автоматизированного проектирования за приемлемое время, возможности одновременной работы нескольких пользователей с ресурсами ТО, высокой надежности, приемлемой стоимости и т. п.

Удовлетворение перечисленных требований возможно только путем организации ТО САПР в виде специализированной иерархической вычислительной системы (ВС) или вычислительной сети с развитым периферийным оборудованием, ориентированным на ввод, обработку и выдачу текстовой и графической информации.

Задача разработки ТО САПР заключается в обосновании, расчете и выборе структуры многоуровневого комплекса технических средств (КТС) САПР, ориентированного на решение задач автоматизированного проектирования определенного класса объектов. Построение КТС может осуществляться путем комплексирования как стандартного оборудования (ЭВМ, каналы, дисплеи, устройства внешней

памяти и т. д.), так и специально разработанного для КТС САПР (АРМ, графопостроители, кодировщики и т. д.).

Создание многоуровневых КТС предполагает наличие на высшем уровне одной или нескольких ЭВМ большой производительности (типа ЕС ЭВМ старших моделей или «Эльбрус»). Эти ЭВМ предназначены для решения сложных задач проектирования, требующих больших затрат машинного времени и памяти. На низших уровнях иерархии могут находиться ЭВМ средней производительности, а также мини- и микро-ЭВМ, входящие в состав АРМ (терминальные ЭВМ). Эти ЭВМ предназначены для решения сравнительно несложных задач проектирования, для управления работой комплекта периферийного оборудования и для организации обменов информации между различными уровнями КТС.

Для определения структуры КТС и параметров входящих в него компонентов могут служить ограничения: снизу — на число программ  $N$ , входящих в состав программного обеспечения САПР; сверху — на среднее время  $T$  реакции КТС на поступившую задачу проектирования; снизу — на объем оперативной памяти  $E_{\text{п}}$  для хранения программ проектирования; сверху — на время  $T_{\text{п}}$ , необходимое процессору для решения усредненной задачи в однопрограммном режиме, а также номенклатура периферийного оборудования КТС САПР.

Комплексы технических средств САПР создаются на базе средств вычислительной техники общего назначения — Единой системы ЭВМ, «Эльбрус», БЭСМ-6, мини- и микро-ЭВМ различных типов.

**Единая система ЭВМ.** Представляет собой совокупность технических средств и программного обеспечения, на основе которых можно создавать вычислительные системы различной конфигурации.

Концепции, заложенные в ЕС ЭВМ (программная совместимость, универсальность, модульный принцип построения технических средств и программного обеспечения), позволяют совершенствовать все компоненты системы. С помощью набора команд ЕС ЭВМ производят операции с фиксированной и плавающей запятыми, десятичные операции и операции с полями переменной длины.

Система программного обеспечения ЕС ЭВМ состоит из операционных систем, пакетов прикладных программ и программ технического обслуживания. Единая система ЭВМ-2 (ЕС-1035, ЕС-1045 и др.) в отличие от ЕС ЭВМ-1

(ЕС-1020, ЕС-1022, ЕС-1030, ЕС-1050, ЕС-1052) имеет более развитый набор команд для обеспечения новых функций процессоров и каналов, обеспечения многопроцессорных средств и др. В старших моделях ЕС ЭВМ-2 используется принцип конвейерной обработки. В ЭВМ второй очереди имеется три типа каналов: селекторный, байт-мультиплексный и блок-мультиплексный. Последний обладает свойствами как селекторного, так и мультиплексного канала и предназначен для организации параллельной работы нескольких высокоскоростных внешних устройств по одной информационной линии. Все модели ЕС ЭВМ-2 поставляются с операционной системой версии 6.1 объемом 3,5—4 млн. команд.

На основе ЕС ЭВМ создаются многомашинные и многопроцессорные комплексы. В многомашинной системе каждая ЭВМ управляется собственной ОС в отличие от многопроцессорной, где операционная система общая.

Основные данные о ЕС ЭВМ-2 и ЕС ЭВМ-3 приведены в табл. 7.1. В табл. 7.2 указаны основные параметры двухмашинных вычислительных комплексов на базе ЕС ЭВМ и многопроцессорного вычислительного комплекса (МВК) «Эльбрус».

В двухмашинных вычислительных комплексах на базе ЕС ЭВМ объединение технических средств организовано на трех структурных уровнях:

Таблица 7.1

Основные данные	ЕС ЭВМ-2				ЕС ЭВМ-3		
	ЕС-1035	ЕС-1045	ЕС-1061	ЕС-1065	ЕС-1036	ЕС-1046	ЕС-1066
Производительность, тыс. оп/с	140	860	2000	6000—7000	400	1300	5000
Емкость оперативной памяти, Мбайт	0,5	1—4	8	16	2—4	4—8	8—16
Суммарная пропускная способность каналов, Мбайт/с	1,2	5	10,5	15	4,5	8,5	18
Потребляемая мощность, кВт·А	43	30	60	150	40	50	100
Занимаемая площадь, м <sup>2</sup>	100—110	120	200	410	100	100	190

таблица 7.2.

Основные данные	БК-2Р-35А	БК-2М-45	БК-2Р-60	БК-2П-45	«Эльбрус-1»
Производительность, тыс. оп/с	190—380	850—1700	1000—2000	1700	$1,2 \cdot 10^3$ — $12,1 \cdot 10^3$
Емкость оперативной памяти, Мбайт	2	2—8	8	2—8	1—8
Число ЭВМ (процессоров)	2	2	2	2	1—10 (спецпроцес- сор—1)
Потребляемая мощность, кВт·А	140	60	225	50	120—786
Занимаемая площадь, м <sup>2</sup>	240—260	250	470	135	298—1267

процессоров с помощью средств прямого управления; каналов ввода/вывода с помощью адаптеров канал — канал;

общего поля внешней памяти с помощью накопителей на магнитных дисках и магнитных лентах.

Приведенные в табл. 7.1 машины ЕС ЭВМ-3 полностью совместимы с ЕС ЭВМ-2, поставляются с ОС ЕС версии 6.1. В них сохранена архитектура ЕС ЭВМ-2 с некоторыми усовершенствованиями: широкое применение микропрограммного управления, дальнейшее развитие принципов виртуальных систем, возможность использования проблемно-ориентированных процессоров и др.

**Многопроцессорные вычислительные комплексы «Эльбрус».** Предназначены для решения различных научно-технических задач, в том числе и задач автоматизированного проектирования, с высокими требованиями к производительности и живучести вычислительной системы.

Одна из главных особенностей МК «Эльбрус» — наличие в его составе специализированного процессора, работающего в режиме аппаратной эмуляции ЭВМ БЭСМ-6 и позволяющего обеспечивать реализацию обширной библиотеки прикладных программ, написанных для ЭВМ БЭСМ-6.

Комплексы «Эльбрус» выпускают нескольких типов, структурные и логические решения которых одинаковы, а отличие заключается в числе используемых центральных и специализированных процессоров, в емкости ОЗУ и в коли-

чественном составе внешних запоминающих устройств. Основными особенностями структуры центрального процессора являются безадресная система команд, динамическое распределение сверхоперативных регистров, работа с полями переменной длины, виртуальная память емкостью  $2^{32}$  слов, распределение оперативной памяти сегментами переменной длины, организация параллельных процессов и т. п.

Создаваемые на базе МК «Эльбрус» вычислительные комплексы имеют высокие показатели надежности и достоверности обработки информации за счет модульного принципа построения и наличия системы реконфигурации, которая при возникновении сигнала неисправности от системы аппаратного контроля модуля автоматически исключает его из состава комплекса и восстанавливает прерванный вычислительный процесс.

**Мини- и микро-ЭВМ.** Мини- и микро-ЭВМ, выпускаемые отечественной промышленностью, являются универсальными ЭВМ с широкими функциональными и техническими возможностями. Будучи оснащенными проблемно-ориентированным программным обеспечением и специальным периферийным оборудованием, они становятся важнейшими компонентами ТО САПР. Основные характеристики ряда мини- и микро-ЭВМ приведены в табл. 7.3.

Т а б л и ц а 7.3

Основные данные	СМ-4	СМ-1420 (СМ-51/20)	СМ-1430 (СМ-51/30)	СМ-1800
Максимальная емкость ОЗУ, Кбайт	256	256	4000	64
Быстродействие процессора, оп/с	400	1000	2000	400
Графопостроитель	—	—	—	—
Устройство ввода графической информации	—	—	—	—
Интерфейс для связи с другими ЭВМ	ОШ	ОШ	ОШ	ИРПС С2 ИРПР

Из множества мини- и микро-ЭВМ для целей создания КТС САПР можно выделить следующие.

Микро-ЭВМ «Искра-226» ориентирована на обработку научной информации, выполнение инженерных расчетов и автоматизацию проектных работ. Для этих целей в состав внешних устройств включены графический дисплей на  $256 \times 512$  графических точек, графопостроитель, устройство ввода графической информации, а также накопители на магнитных лентах и магнитных дисках. Благодаря наличию интерфейса для связи с другими ЭВМ «Искра-226» может использоваться в качестве интеллектуального терминала в распределенных КТС САПР.

Микро-ЭВМ СМ-1300 предназначена для использования в качестве центрального вычислителя в составе локальных вычислительных сетей (в том числе и сетей для реализации САПР) и систем машинной графики. Микро-ЭВМ СМ-1300 снабжена графопостроителем, графическим дисплеем на  $1024 \times 1024$  графических точек, устройством ввода графической информации.

Широкое применение мини- и микро-ЭВМ находят в организации автоматизированных рабочих мест (СМ-4, СМ-1420, СМ-1300 и др.). Автоматизированные рабочие места являются основным техническим средством связи оператора с КТС. Для достижения наибольшей эффектив-

СМ-1300	СМ-1300.01	СМ-1634	«Искра-226»	«Электроника-НЦ 80-20/3»
64	256	128	64	64
500	500	200	500	500
+	+	-	+	+
+	+	-	+	+
ИРПС С2 ИРПР С1-ФЛ	ИРПС С2 ИРПР	2К ИРПР	ИРПС С2	МПИ ИРПС

ности связи в состав АРМ обычно включают дисплей с высокой разрешающей способностью и со световым пером, а также большое количество цифровых преобразователей для конструирования графических проектов.

Для разработки изделий электронной техники отечественной промышленностью создано семейство унифицированных интерактивных вычислительных комплексов на базе микро-ЭВМ «Электроника». Средствами диалога в комплексах служат символьные и графические дисплеи. Состав устройств, входящих в эти комплексы, приведен в табл. 7.4.

Таблица 7.4

Наименование	Тип применяемой ЭВМ	Устройства внешней памяти	Печатающие устройства	Графопостроитель
15УТ-4-017	«Электроника 100-25»	НМД СМ-5400 НМЛ ИЗОТ 5003	DZM-180	ЭМ-7022
15УТ-1-037	«Электроника-60М»	НГМД «Электроника ГМД-70»		
15УТ-8-060	«Электроника-79»	НМД СМ-5404 НМЛ СМ-5309	DZM-180	
15УТ-1-061	«Электроника-60М»	НМД ИЗОТ 1370 НГМД «Электроника ГМД-70»		ЭМ-739А
Интерактивная графическая система	«Электроника-82»	НМД СМ-5404 НМЛ СМ-5306	DZM-180	ЭМ-739А

**Режимы работы аппаратуры в КТС САПР.** Комплексы технических средств САПР и автоматизированные рабочие места являются системами коллективного пользования. Поэтому ЭВМ, входящие в состав ТО САПР, и их операционные системы должны допускать одновременное решение нескольких задач от различных пользователей, т.е. должны работать в режиме мультипрограммирования.

По характеру обмена информацией между пользователем и ЭВМ различают пакетный и диалоговый режимы работы. В пакетном режиме решаются задачи, для которых возможна и целесообразна полная формализация и для решения которых не требуется оперативное вмешательство человека.

В диалоговом (интерактивном) режиме решаются задачи, для которых отсутствуют или являются неэффективными формальные правила принятия решений в точках ветвления алгоритмов и выполняются условия предпочтительности диалогового режима (небольшое время реакции системы на запрос пользователя и малый объем информации, вводимой пользователем при обращении к ЭВМ). Характерная черта КТС САПР — наличие диалогового режима.

Мультипрограммный диалоговый режим работы КТС называется режимом разделения времени. Этот режим необходим в САПР для организации одновременной работы нескольких пользователей САПР на общих технических средствах.

Отдельные устройства ввода/вывода могут работать в режиме on-line (на линии), где осуществляется непосредственная электрическая связь внешнего устройства с терминальной ЭВМ, и в режиме off-line (вне линии), где устройство работает автономно.

**Подходы к выбору структуры КТС САПР.** При проектировании КТС САПР широкое применение находят методы оптимизации и моделирование.

Методы синтеза КТС САПР предназначены для выбора оптимальной структуры системы и ее компонентов и базируются в основном на методах исследования операций — методах математического программирования, теории графов и сетей, теории массового обслуживания и др.

Методы анализа КТС предназначены для оценки вычислительной мощности комплекса и необходимой емкости оперативной и внешней памяти вычислительных средств и базируются на применении методов имитационного и аналитического моделирования. Методы имитационного моделирования позволяют учесть большое число параметров и достигнуть большой степени адекватности при соответствующем усложнении модели проектируемого объекта. Однако процесс построения имитационных моделей является довольно трудоемким и требует в качестве первоначальных методов оценки структур КТС САПР использования аналитических методов, которые применяют для построения моделей синтеза оптимальных структур.

Проектирование КТС САПР начинается с анализа технических требований к системе проектирования, итогов предпроектного обследования объектов проектирования и исходных данных. Как правило, предварительное комплек-



сирование структуры технического обеспечения САПР производится с помощью справочных материалов.

В соответствии с номенклатурой выпускаемого оборудования определяют минимальный состав технических средств, необходимый для реализации заданного класса задач автоматизированного проектирования, что дает возможность принять в расчетах конкретные значения параметров применяемых технических средств.

При решении задачи комплексирования технических средств необходимо определить типы применяемых ЭВМ, их число и иерархию для обеспечения решения заданного набора задач в приемлемое для разработчиков время. При этом следует учитывать, что часть задач может быть решена в пакетном режиме, где на сроки решения каждой задачи накладываются плановые или директивные ограничения (например, конец смены, через две недели и т. д.), а основная масса задач автоматизированного проектирования и конструирования должна решаться в интерактивном режиме и результат решения каждой задачи должен получаться в определенном временном интервале после подачи запроса на ее решение.

В КТС САПР, содержащих большое число терминалов, время реакции системы

$$T = T_{в.с} + T_{п.д},$$

где  $T_{в.с}$  — время ответа вычислительной системы;  $T_{п.д}$  — время передачи данных.

В КТС с развитой иерархией  $T_{п.д} > T_{в.с}$  и пропускная способность системы передачи данных существенно влияет на общую производительность КТС.

Необходимая емкость оперативной памяти  $E$  каждой ЭВМ, входящей в состав КТС САПР:

$$E = E_0 + A(E_n + E_p),$$

где  $E_0$  — емкость резидентной части программ ОС;  $A$  — уровень мультипрограммирования, определяемый возможностями ОС (для ДОС СМ-3 и СМ-4  $A \leq 24$ );  $E_n$  — емкость программной области ОЗУ в расчете на одну программу проектирования;  $E_p$  — емкость рабочей области оперативной памяти (массивы данных, справочные таблицы) в расчете на одну задачу.

Структуру и состав КТС САПР выбирают с учетом развития функциональных возможностей САПР в рамках интегрированных автоматизированных систем управления (ИАСУ), представляющих собой симбиоз САПР—ГАП—

АСУТП — АСУП. С развитием интеграции САПР — ГАП иерархическая структура КТС САПР должна трансформироваться в локальную вычислительную сеть — систему распределенной обработки информации. В состав таких систем входят интеллектуальные станции автоматизированного или автоматического проектирования, построенные на базе сетей из микро-ЭВМ и связанные между собой на основе обмена информационными носителями, например, гибкими дисками или с помощью интерфейсов, образующих сеть проектирования.

Создание и широкое внедрение 32-разрядных микро-ЭВМ и персональных компьютеров, увеличение емкости их оперативной памяти, расширение номенклатуры и емкости накопителей на магнитных дисках, появление более эффективных специализированных операционных систем способствуют широкому их использованию в составе КТС САПР и ИАСУ. Однако при проектировании ТО САПР необходимы детальные расчеты системных характеристик применения микро-ЭВМ в составе КТС, с тем чтобы обеспечить достаточно эффективное их использование. Создаваемые КТС САПР и ИАСУ должны обеспечивать эффективное использование устройств сбора и передачи информации, гарантированные характеристики надежности КТС, возможность гибкого изменения структуры, номенклатуры и количества технических средств, обеспечивающих поэтапный ввод в действие компонентов КТС и его модернизацию.

## 7.2. ОПРЕДЕЛЕНИЕ ХАРАКТЕРИСТИК И ОЦЕНКА КАЧЕСТВА СОЗДАВАЕМОЙ САПР

Разработка систем автоматизированного проектирования является чрезвычайно трудоемким процессом, в котором задействованы большие коллективы специалистов. Стоимость современных САПР довольно высока, поэтому техническое, программное и информационное обеспечение САПР в настоящее время составляет значительную долю основных производственных фондов предприятий и организаций, использующих САПР. Фондоотдача этой составляющей зависит от того, имеет ли пользователь все компоненты САПР, необходимые для организации автоматизированного проектирования заданной номенклатуры изделий, удовлетворяют ли пользователя состав и качество КТС, программного и информационного обеспечения САПР.

Исходя из особенностей вычислительного процесса к САПР предъявляются следующие требования.

1. Наличие диалогового режима, вытекающего из сущности проектирования как процесса принятия сложных решений в условиях неоднозначности планов вычислений. Конфликтная часть процесса проектирования отображается интерактивным терминалом, а процесс принятия решений осуществляется проектировщиком.

2. Коллективный доступ пользователей к ресурсам системы. Доступ к ресурсам САПР означает возможность вызова любого программного модуля, обращения к базе данных, отображения результатов проектирования на экране дисплея, на АЦПУ или на графопостроителях в виде конструкторской документации.

3. Работа программ с базами данных различной логической структуры и сложности путем обеспечения независимости программ от способа организации данных.

4. Возможность автономной отладки программного обеспечения САПР.

5. Возможность дальнейшего развития системы путем развития программного, информационного и технического обеспечения.

При наличии необходимого набора КТС, ПО и ИО определяющим становится качество САПР. Под *качеством* понимается совокупность свойств изделия, обуславливающих его пригодность удовлетворять определенные потребности в соответствии с его назначением. Количественные характеристики одного или нескольких свойств называют *показателями качества*. Улучшение показателей качества проектируемых САПР способствует проведению бездефектного проектирования технических объектов и как следствие — улучшение их качества.

В соответствии с государственными стандартами оценки качества изделий рассмотрим следующие группы показателей для оценки САПР: назначения, функциональные, эксплуатационные, технологические, экономические и стандартизации.

**Показатели назначения.** К этим показателям относятся область применения САПР, размерность решаемых задач проектирования и требуемые технические средства для реализации САПР.

**Функциональные показатели.** Показатели этой группы характеризуют свойства САПР, проявляющиеся непосредственно в процессе функционирования; точность; надеж-

ность; устойчивость к искажающим воздействиям; время реакции системы.

Точность проектирования зависит от многих факторов, в частности от степени адекватности математической модели проектирования и проектируемого объекта; от погрешности математических методов, используемых при решении задач оптимизации; от погрешности округления величин и т. д.

Под надежностью САПР будем понимать свойство системы не терять работоспособность при наличии отказов и сбоев в КТС САПР (аппаратурная надежность) и при преобразовании определенного набора исходных данных из-за использования при этом программ, содержащих ошибки (программная надежность).

Под устойчивостью САПР к искажающим воздействиям будем понимать способность системы выполнять свое функциональное назначение при наличии внешних и внутренних искажающих воздействий. Источниками внешних воздействий могут быть сети электропитания, непригодные для эксплуатации вычислительной техники помещения, ошибки в подготовке данных и т. п. Источниками внутренних воздействий являются сбои и отказы в самом КТС САПР, которые могут приводить к искажению кодов программы, результатов проектирования и т. п. Количественной оценкой показателя устойчивости может служить такой критерий, как область устойчивого функционирования. Задача определения области устойчивого функционирования САПР аналогична задаче определения допусков и технических требований, рассмотренной в § 6.5.

Время реакции системы  $T_c$  характеризуется временным интервалом между моментом поступления в КТС задания на проектирование и моментом выдачи соответствующей документации. Величина  $T_c$  является случайной и зависит от характеристик используемых вычислительных средств, периферийного оборудования и трансляторов, структуры программного и информационного обеспечения, а также от структуры алгоритмов проектирования и размерности решаемых задач.

**Эксплуатационные показатели.** Эти показатели характеризуют приспособленность САПР к эксплуатации и включают в себя простоту освоения и подготовки к работе, диагностируемость, количество и требуемую квалификацию обслуживающего персонала, возможность внесения исправ-

лений и доработок при изменении параметров проектируемых объектов и требований к проектированию.

**Технологические показатели.** К технологическим относятся показатели распределения затрат ресурсов и времени на ввод САПР в эксплуатацию, поддержание ее работоспособности и сопровождение в специфических условиях конкретного пользователя.

**Экономические показатели.** Эти показатели характеризуют затраты на разработки или приобретение всех видов обеспечения САПР, создание и испытание опытного образца системы, тиражирование, освоение, обслуживание и сопровождение САПР.

**Уровень стандартизации.** Проверка этого уровня осуществляется путем проведения нормоконтроля всей документации на соответствие требованиям ГОСТ и ЕСПД.

Важной характеристикой САПР является степень ее информационной связи с окружающей средой. Систему называют *статической*, если в процессе проектирования не требуется информации о состоянии внешней среды в данный момент времени, и *динамической*, если при своем функционировании система непрерывно потребляет информацию о состоянии внешней среды из источников, находящихся вне системы проектирования.

Анализируя множество приведенных показателей, нетрудно определить, что одна группа показателей является измеряемой, а другая — неизмеряемой. С точки зрения полезности обе эти группы могут иметь одинаковое значение, так как в разной степени содержат информацию о свойствах САПР. К измеряемым характеристикам относятся все функциональные, экономические и большинство эксплуатационных показателей, а к неизмеряемым — область применения, требуемая квалификация обслуживающего персонала и т. д. При оценке неизмеряемых показателей целесообразно использовать экспертные методы.

Системы автоматизированного проектирования представляют собой совокупность технических, программных, информационных и других средств, находящихся между собой в тесной взаимосвязи, поэтому на показатели качества САПР существенное влияние оказывают показатели качества входящих в состав системы компонентов.

Рассмотрим некоторые показатели качества технического, программного и лингвистического обеспечения САПР.

**Оценка качества технического обеспечения САПР.** Показатели качества можно расчленить на две группы:

системные показатели, которые отражают наиболее общие свойства технических средств и характеризуют требования пользователей аппаратуры к эффективности ее эксплуатации (стоимость, производительность вычислительных средств, емкость памяти, число пользователей, число алфавитно-цифровых и графических терминалов, загруженность отдельных устройств и др.);

пользовательские показатели, которые отражают специфические требования конкретных типов систем автоматизированного проектирования (режимы работы аппаратуры, применяемые языки, скорость и точность переработки информации, организация диалогового режима и др.).

В связи с расширением сферы применения средств вычислительной техники и увеличением объемов вычислительных работ эффективная организация использования ЭВМ приобрела государственное значение. Поэтому система показателей оценки качества КТС САПР должна учитывать степень загрузки ЭВМ и других технических средств, входящих в состав КТС.

Характеристикой фактической производительности ЭВМ, которая учитывает ее реальную загруженность, является комплексная производительность ЭВМ

$$P = P_{\text{ном}} \sum_{i=1}^{\Pi} T_i / 1440,$$

где  $P_{\text{ном}}$  — номинальная производительность ЭВМ, тыс. оп./с;  $\Pi$  — пропускная способность ЭВМ, заданий/сут;  $T_i$  — время использования центрального процессора  $i$ -м заданием, мин.

Процессорное время на усредненную задачу

$$T_{\Pi} = \sum_{i=1}^{\Pi} g_i T_i = \sum_{i=1}^{\Pi} \sum_{j=1}^{s_i} g_i \alpha_j n_{ij} \tau,$$

где  $n_{ij}$  — число процессорных операций  $j$ -го типа при выполнении  $i$ -го задания;  $\alpha_j \geq 1$  — коэффициент сложности  $j$ -й операции;  $\tau$  — время выполнения элементарной операции;  $s_i$  — число типов процессорных операций в программе выполнения  $i$ -го задания;  $g_i$  — относительная частота появления  $i$ -го задания  $\left( \sum_{i=1}^{\Pi} g_i = 1 \right)$ .

Показатели «стоимость обработки информации» и «обобщенная степень загрузки оборудования» соответствен-

но будут:

$$C = S/P; \quad Q = \sum_{q=1}^k S_j \sum_{i=1}^{\Pi} \tau_{iq}/(ST),$$

где  $S$  — стоимость КТС САПР;  $\tau_{iq}$  — время использования  $q$ -го ресурса КТС  $i$ -м заданием, мин;  $S_j$  — стоимость  $j$ -го ресурса КТС;  $k$  — количество различных ресурсов КТС;  $T$  — интервал времени (принимается равным 1440 мин).

Данный показатель аналогичен показателю «резерв фондоотдачи», используемому в промышленности, и позволяет объективно осуществлять расчеты за использование машинного времени и планировать загрузку КТС.

**Эффективность операционной системы.** Эффективность функционирования вычислительных средств ТО САПР существенно зависит от эффективности работы операционной системы. В вычислительных системах с наличием средств генерации ОС у пользователей имеется возможность выбора конкретного варианта структуры ОС с учетом имеющейся конфигурации технических средств, классов решаемых задач и требуемых режимов использования ВС (пакетная обработка, режим телеобработки и т. д.).

Работа ОС требует значительного собственного потребления ресурсов ВС, что приводит к искажению реального вычислительного процесса. Качество вычислительного процесса, организуемого ОС, оценивается прежде всего степенью полезного использования ресурсов ВС и расходом ресурсов на организацию вычислительного процесса. Пусть имеется множество структур ОС  $R$  и в ВС используется ОС, имеющая структуру  $r \in R$ . Эффективность структуры  $r$  на некотором интервале  $T$  можно определить с помощью коэффициента использования основного ресурса (центрального процессора — ЦП)

$$K_{\text{ЦП}}(r) = \frac{T_{\text{сч}}(r)}{T} = \frac{T_{\text{сч}}(r)}{T_{\text{сч}}(r) + T_0(r)},$$

где  $T_{\text{сч}}(r)$  и  $T_0(r)$  — соответственно время нахождения ЦП в состояниях СЧЕТ и ОЖИДАНИЕ.

Поскольку в состоянии СЧЕТ выполняются как пользовательские программы, так и программы ОС, обычно вводят коэффициенты загрузки ЦП полезной работой  $K_{\Pi}(r)$  и программами ОС  $K_{\text{ОС}}(r)$ :

$$K_{\Pi}(r) = \frac{T_{\text{сч}}(r) - T_{\text{ОС}}(r)}{T_{\text{сч}}(r) + T_0(r)}; \quad K_{\text{ОС}}(r) = \frac{T_{\text{ОС}}(r)}{T_{\text{сч}}(r) + T_0(r)},$$

где  $T_{\text{ОС}}(r)$  — время выполнения программ ОС.

Эффективность использования ЦП  $K_{\Pi}(r)$  может быть улучшена либо путем сокращения времени пребывания ЦП в состоянии ОЖИДАНИЕ (за счет перераспределения библиотек между НМД и файлов на отдельных НМД), либо путем сокращения затрат на работу программ ОС (переформированием ядра ОС, коррекцией размещения модулей ОС, коррекцией использования системных управляющих программ).

Качество генерации ОС оценивают измерительными мониторами, фиксирующими смену текущих состояний ЦП и соответствующих им параметров.

**Оценка качества программного обеспечения САПР.** Разработка программного обеспечения (ПО) САПР является одним из основных и ответственных этапов создания САПР. При постоянном расширении ПО САПР расходы на ПО непрерывно растут и, как правило, значительно превышают расходы на ТО САПР. Разработка ПО в настоящее время поставлена на индустриальную основу. Конечным продуктом разработки ПО являются *программное изделие* (ПИ), под которым понимаются программы, записанные на носителях данных и прошедшие стадии промышленной разработки, и комплект документации, сопровождающий ПИ.

Эффективность функционирования САПР во многом определяется качеством программных изделий. Низкое качество ПИ наносит большой материальный ущерб, вызывает недоверие к ПИ, отвлекает специалистов пользователей от основной работы. Стоимость локализации и устранения ошибок в эксплуатируемом ПИ обходится в десятки раз дороже, чем устранение ошибок на стадии проектирования.

Оценка качества ПИ — задача многофакторная и трудно формализуемая. Определяющим фактором при оценке качества программы является многообразие интересов пользователя. По этой причине невозможно предложить какую-либо единственную универсальную меру качества ПИ, здесь требуется множество характеристик, охватывающих целый спектр желаемых свойств. В литературе встречаются различные наборы показателей качества ПИ. Рассмотрим основные из них.

1. Документированность — характеристика программного обеспечения, объединяющая свойства понятности, осмысленности и завершенности. Программное обеспечение обладает свойством понятности, если оно позволяет пользователю понять назначение программных средств.



Программное изделие обладает свойством осмысленности, если его документация не содержит избыточной информации и не допускает многозначной интерпретации терминов и символов, и завершенности, если в нем присутствуют все необходимые компоненты, каждый из которых разработан всесторонне.

2. Эффективность — свойство ПИ выполнять требуемые функции без излишних затрат ресурсов. В качестве оценки эффективности можно принять характеристику программы, значение которой прямо пропорционально быстродействию и обратно пропорционально объему используемых ресурсов. Термин «ресурсы» понимается в самом широком смысле: это могут быть оперативная и внешняя память, устройства графического ввода и графопостроители, пропускная способность канала и т. д.

3. Надежность — свойство ПИ устойчиво выполнять необходимые функции. Надежность прежде всего подразумевает отсутствие ошибки в программе, но так как они неизбежны, то программа должна быть построена так, чтобы все ошибки могли быть просто исправлены, а это возможно при наличии качественной программной документации.

4. Простота использования — свойство ПИ нормально функционировать в четко определенной области применения данного ПИ.

5. Удобство эксплуатации — свойство ПИ адаптироваться в соответствии с вновь появившимися требованиями. Удобство эксплуатации предполагает понятность, оцениваемость и простоту внесения изменений.

6. Мобильность — свойство ПИ, заключающееся в его приспособленности к переносу на ЭВМ иного типа, чем тот, для которого ПИ разрабатывалось, а также к смене операционной системы.

7. Совместимость — свойство ПИ, заключающееся в том, что ПИ или входящие в него программы могут выполняться в технической, информационной и программной среде иного типа, чем та, для которой оно непосредственно предназначено.

8. Испытуемость — свойство ПИ, заключающееся в наличии возможности достаточно просто оценивать правильность функционирования программы в условиях конкретной среды.

9. Стоимость программы является функцией всех ее характеристик. Улучшение любой из характеристик влияет на стоимость ПИ, поэтому всегда должен быть найден ра-

зумный компромисс между степенью улучшения интересующей нас характеристики ПИ и увеличением его стоимости.

Непосредственная оценка выделенных характеристик крайне затруднительна. Для получения качественного программного обеспечения необходимо принимать специальные меры, направленные на гарантирование заданных характеристик качества. Связь качества ПИ с процессом его разработки и создания аналогична зависимости качества технического объекта от процесса его проектирования и изготовления. Анализ качественных характеристик должен проводиться на каждом этапе процесса разработки и создания ПИ.

**Оценка качества лингвистического обеспечения САПР.** Современные ЭВМ в своем математическом обеспечении имеют компиляторы с различных языков программирования. Выбор языка программирования, наиболее подходящего для решаемых задач, в значительной мере влияет и на качество ПО САПР. Поэтому одним из основных факторов при выборе языка программирования является соответствие языка программирования типу решаемой задачи. Несоответствие языка решаемой задаче создает трудности в написании и отладке программы.

Язык программирования должен обладать эффективным средством для установления соглашений и регламентов, обеспечивающих быструю и надежную стыковку отдельно создаваемых частей общей программы.

Критериями качества языка программирования и транслятора являются простота, надежность, быстрота трансляции, эффективность объектного кода, удобочитаемость и блочность структуры. Рассмотрим эти качества подробнее.

**Простота.** Средства языка должны быть простыми для понимания и выражены в легко запоминающейся и интуитивно понятной для программирования форме, которая является естественной основой для общения программиста с машиной. Простоте языка способствуют единообразие в символике и организации, целостность основных концепций.

**Надежность.** Под надежностью языка понимается его свойство, которое характеризуется вероятностью безошибочного написания и трансляции программы заданного размера.

**Быстрота трансляции.** Скорость трансляции — существенный показатель, потому что языки программиро-

вания не исключают этап отладки в процессе разработки программы и довольно часто программа много раз транслируется в процессе отладки, а выполняется только один раз. Одним из путей увеличения скорости трансляции является использование независимой трансляции, заключающейся в том, что компилятор может транслировать программу по частям.

**Эффективность объектного кода.** Получение эффективных машинных кодов возможно в процессе трансляции только с языков, при разработке которых предусматривалась оптимизация программ компилятором (языки ФОРТРАН, АДА). Наличие хорошо проработанных компиляторов позволяет получать программы приемлемых размеров и эффективности. Машинно-независимые оптимизаторы должны преобразовать неэффективные программы в более эффективные, полностью эквивалентные и записанные в том же исходном языке.

**Удобочитаемость.** От того, насколько удачны комментарии и сколь удобно они размещаются в программе, существенно зависит скорость понимания программы. Программы пишутся программистами, читаются ими же или их коллегами еще до ввода в ЭВМ, поэтому качеству комментариев (удобочитаемости) при разработке языка должно уделяться большое внимание.

**Блочность структуры.** Блочную структуру имеют программы, написанные на языках АЛГОЛ, ПЛ/1 и АДА. В этих языках каждая программа или подпрограмма организуется в виде последовательности вложенных друг в друга блоков, ограниченных специальными указателями (например, в АЛГОЛ — словами *begin* и *end*).

Блочная структура обеспечивает гибкую систему управления данными и позволяет получить высокоэффективно выполняемые объектные коды.

Ясность структуры программы дает программисту и пользователю большие преимущества. Программу, структура которой отражает структуру реализуемого ею алгоритма, легко проектировать, кодировать, отлаживать, понимать и сопровождать.

### 7.3. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ В АВТОМАТИЗИРОВАННОМ ПРОЕКТИРОВАНИИ

Сложность и многообразие процессов функционирования проектируемых технических систем не всегда позволяют получать для них адекватные математические модели,

сформулированные в виде различных аналитических соотношений. Автоматизированное проектирование в этих случаях базируется на использовании методологии имитационного моделирования.

Имитационное моделирование — метод исследования, основанный на том, что изучаемая динамическая система заменяется ее имитатором и с ним проводятся эксперименты с целью получения информации об изучаемой системе. Напомним, что динамическими являются такие системы, которые изменяются во времени. Имитаторы могут быть реализованы на ЭВМ, а также на гидродинамических, механических или электронных системах.

Важное значение имеет эксперимент, который может проводиться человеком или выполняться автоматически как в соответствии с заранее составленным планом, так и последовательно, когда цели нового эксперимента устанавливаются на основе анализа результатов проведенных экспериментов. Это имеет особое значение в случае реализации имитатора на ЭВМ, в частности при автоматизированном проектировании, когда составляется программа имитатора, являющаяся в свою очередь программой функционирования проектируемого объекта.

При имитационном моделировании процессов не требуется преобразовывать аналитические выражения в специальную систему уравнений относительно искомых величин. Для имитационного моделирования характерно воспроизведение на ЭВМ явлений, описываемых математической моделью, с сохранением их логической структуры и последовательности чередования во времени.

Имитационное моделирование реализуется моделирующим алгоритмом, в соответствии с которым в ЭВМ имитируется функционирование исследуемой системы с учетом выбранного уровня детализации для получения нужных характеристик. Эти характеристики выводятся на печать и используются в качестве прямых или косвенных результатов проектирования. Таким образом, в процессе имитационного моделирования конструируется модель проектируемого объекта. На ней проводятся эксперименты с целью изучения закона функционирования и поведения проектируемого объекта с учетом заданных ограничений и целевой функции.

Широко распространены два подхода к организации имитационных моделей. Первый подход заключается в том, что каждый элемент сложной системы представляется в не-

которой стандартной форме, когда уже заготовлены программные модели — модули. С помощью специальных схем описываются взаимосвязи между модулями и конструируется модель сложной системы.

Второй подход предусматривает создание универсальной имитационной модели, которая может настраиваться на любой объект заданного класса. Для этого необходимо, чтобы структурные и функциональные характеристики, отличающие один объект от другого, не входили в структуру модели и ее описание, а являлись легко заменяемыми исходными данными. В этом случае при подготовке к моделированию конкретного объекта из заданного класса программирование оказывается ненужным.

Исходя из принципа построения имитационных моделей все их компоненты действуют последовательно. Чтобы произвести в модели одновременность нескольких событий, происходящих в различных частях реальной системы, необходимо построить определенный механизм задания времени в моделях. Существуют два основных метода: фиксированного шага и шага до следующего события. В частности, при моделировании средств вычислительной техники, как правило, используются оба метода.

До построения моделирующего алгоритма должны быть решены все принципиальные вопросы выбора математического аппарата исследования. Для имитации процессов функционирования отдельных элементов объекта и всего объекта в целом должны быть выбраны основные операторы, которые увязываются между собой в соответствии с формализованной схемой исследуемого процесса. К основным операторам относятся вычислительные (арифметические) и логические операторы, операторы формирования реализаций случайных процессов и неслучайных величин, а также операторы счета.

*Вычислительные операторы* предназначены для реализации любых вычислительных функций с помощью системы арифметических операций, присущей системе команд моделирующей ЭВМ. Для выполнения вычислительных операторов необходимо четко определить, какие величины должны быть вычислены в результате реализации сформированного оператора, и обеспечить наличие к моменту начала работы оператора всех необходимых данных, получаемых от других операторов алгоритма.

Для имитации действия различных случайных факторов на моделируемый процесс необходимо наличие опера-

торов *формирования реализаций случайных процессов*. Исходными данными для получения этих операторов служат таблицы и генераторы случайных чисел.

Формирование различных *констант* и *неслучайных функций времени* принципиально не отличается по сущности от формирования обычных вычислительных операций.

Для подсчета числа различных моделируемых объектов, обладающих заданными свойствами, формируются *операторы счета (счетчики)*. Результаты, выдаваемые счетчиками, часто являются исходными данными для логических служебных операторов, обеспечивающих синхронизацию моделирующего алгоритма.

Логические операторы предназначены для проверки справедливости заданных условий и выработки признаков, обозначающих результат проверки.

Процесс имитации включает в себя большое число операций, связанных с формированием, преобразованием и использованием реализации случайных событий, величин и процессов, поэтому результаты моделирования также носят случайный характер. Они отражают случайные сочетания действующих факторов, складывающихся в процессе моделирования. Искомые величины при имитационном моделировании определяют в результате статистической обработки совокупностей данных некоторого числа реализаций процесса моделирования. Совокупность реализаций выступает в роли статистического материала при машинном эксперименте, а оценка параметров — в роли экспериментальных данных, поэтому имитационное моделирование иногда называют методом статистического моделирования.

При автоматизированном проектировании имитационные модели предназначены для изучения особенностей функционирования проектируемых структур, состоящих из разнообразных элементов (дискретных и непрерывных, детерминированных и стохастических и т. д.). Имитационные программы строят по модульному принципу, при котором все элементы системы описываются единообразно в виде некоторой стандартной математической схемы — модуля. Схемы и операторы сопряжения модулей друг с другом позволяют строить универсальные программы имитации, которые должны осуществлять ввод и формирование массива исходных данных для моделирования, преобразования элементов системы и схем сопряжения к стандартному виду, имитацию модуля и взаимодействия элементов системы, обработку и анализ результатов моделирования,

а также управление моделью. Модель проектируемого объекта должна быть не столь упрощена, чтобы стать тривиальной, однако не настолько детализирована, чтобы стать громоздкой в обращении.

Большое внимание при моделировании уделяется выбору языковых средств. В настоящее время имеется большое число специализированных языков моделирования, поэтому для определения лучшего языка при конкретных приложениях возникают серьезные трудности. Выбор языка программирования для описания имитационных моделей в первую очередь определяется постановкой задачи, когда учитываются характеристики объекта моделирования, тип разрабатываемой модели, условия проведения эксперимента.

Языки моделирования сконструированы так, что позволяют описывать не только сам имитатор, но и вспомогательные действия, которые выполняются в течение эксперимента, прежде всего поступление входных данных и обработку результатов эксперимента.

К языкам моделирования предъявляют следующие требования.

1. Семантика языка моделирования должна иметь также возможности, которые позволяют легко составлять модель имитируемого процесса. Это означает, что хороший язык моделирования должен предоставлять пользователю возможности описания моделируемого процесса в соответствии с предметным мышлением.

2. Язык моделирования должен иметь удобные синтаксические средства для быстрого и простого описания моделируемой системы. Формализованное описание моделируемой системы на языке моделирования позволяет выявить ряд свойств системы, которые трудно обнаружить без формализованного описания.

3. Язык моделирования является удобным средством имитации только тогда, когда все работы на реализованной модели (включая создание имитатора) автоматизированы настолько, что от пользователя не требуется специальных знаний.

Некоторые языки моделирования имеют средства, позволяющие выполнять сбор и обработку информации автоматически, независимо от целей пользователя. Например, для очередей автоматически рассчитываются их максимальная и средняя длины или максимальный и средний периоды пребывания в них элементов. Такие средства облегчают пользователям проведение анализа объектов, исследуемых

в рамках их профессий, при минимальном опыте программирования.

Следует отметить, что в настоящее время существует большое число языков моделирования (более 500), и тем не менее появляются новые. Так, при имитационном моделировании применение универсальных языков высокого уровня имеет ряд недостатков, чем обусловлено появление многих специализированных языков моделирования, в частности языка GPSS. Однако благодаря универсальности языков высокого уровня исследователь может их использовать при решении многих задач автоматизированного проектирования без дополнительных затрат на ознакомление с новыми языками в процессе перехода к решению другого класса задач.

Для решения задач моделирования хорош универсальный язык ПЛ/1, на котором можно решать научно-технические задачи более разнообразные, чем, например, на ФОРТРАНе. Кроме того, ПЛ/1 дает системным программистам средства для решения задач в реальном времени. Элементарные средства языка ПЛ/1 позволяют, например, описывать элементы цифровой вычислительной техники в виде программ имитационных моделей. Язык ПЛ/1 имеет простые операторы для проверки условий выполнения определенных действий, различные варианты реализации операции присваивания, операторы преобразования форм представления данных, несложные правила присваивания имен структурным элементам; позволяет ограничивать учет времени и происходящих действий, простыми операторами реализовать булевы функции, легко реализовать статистические испытания модели при различных данных, изменять структуру модели и т. д.

Схема организации процесса имитационного моделирования при автоматизированном проектировании приведена на рис. 7.1. На первом этапе формируется цель проектирования. Анализируя требования ТЗ на проектирование, оценивают сложность проектируемого объекта и определяют наиболее рациональный путь нахождения математической модели объекта проектирования и ее реализации для целей проектирования — путем имитационного моделирования, путем решения задач математического программирования и т. д. На этапе формирования имитационной модели осуществляется переход от представлений о реальной системе к абстрагированию, к некоторой логической схеме. Подготовка данных состоит в выборе данных, необходимых



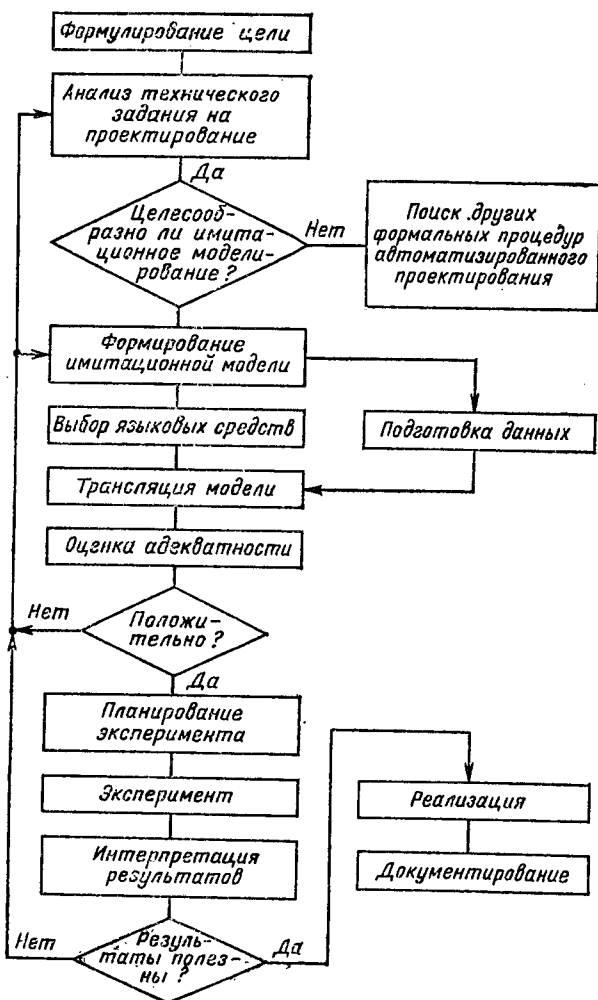


Рис. 7.1. Схема организации процесса имитационного моделирования

для построения модели, и представления их в соответствующей форме. Трансляция заключается в переводе описания модели, представленного на языке высокого уровня или на языке моделирования, на машинный язык (компиляция, редактирование объектных модулей, получение загрузочных модулей). Оценка адекватности имитационной модели

по объекту проектирования проводится на основе сравнения результатов моделирования с информацией, имеющей о проектируемой системе.

При планировании эксперимента составляют план последовательности выполнения процедур в имитационном моделировании и получают оценки результатов моделирования. Экспериментирование представляет собой процесс имитации с получением необходимых статистических данных, а также прямых и косвенных результатов проектирования. Построение выводов по данным, полученным путем имитации, осуществляется на этапе интерпретации.

Документирование включает в себя регистрацию результатов моделирования и проектирования. Реализация подразумевает практическое использование модели и результатов моделирования для целей автоматизированного проектирования.

В качестве примера рассмотрим организацию имитационного моделирования процессора, входящего в состав ЭВМ.

Исходными данными для моделирования являются структурная схема процессора и ограничения ТЗ на ряд параметров (быстродействие, точность и т. д.). Структурная схема дает представление о входящих в его состав блоках и связях между ними. Имитационная модель позволяет представить работу процессора путем абстрагирования способа реализации логических зависимостей (определяемых микропрограммами реализации операций) в виде последовательности выполнения логических операторов. Схема алгоритма моделирования должна быть эквивалентной структурной схеме процессора. По схеме алгоритма производится компоновка отдельных программных модулей, описывающих функционирование реальных блоков процессора, в единую программу. Поскольку обработка элементов программы происходит последовательно, порядок их расположения соответствует распространению исходной информации по всем блокам по мере ее прохождения от входа к выходу. За исходную информацию принимается содержимое всех регистров процессора в начальный момент времени.

Несмотря на достоинства метода имитационного моделирования, позволяющего в принципе исследовать структуры и функционирование объектов любой сложности и на любом уровне детализации, метод имеет ряд недостатков, а именно: невозможность получения аналитических зави-

симостей при одиночном испытании модели, необходимость набора статистики для формальных и аналитических зависимостей, сложность и трудоемкость разработки моделей.

Следует отметить, что если задача проектирования может быть сведена к какой-либо аналитической модели, то необходимость в имитационном моделировании отпадает.

#### 7.4. ПРОЕКТИРОВАНИЕ САПР

Поскольку САПР относятся к сложным системам, их проектирование должно быть блочно-иерархическим.

На верхнем иерархическом уровне — системном — решаются задачи создания описаний комплекса технических средств (КТС) и программно-методических комплексов (ПМК) САПР. На этом уровне в качестве составных частей (элементов) КТС рассматриваются ЭВМ или отдельные устройства: процессоры, запоминающие, ввода/вывода, передачи данных и т. п. Элементами ПМК являются подсистемы программного обеспечения, такие как СУБД или мониторная, и отдельные программы, реализующие различные проектные процедуры. Необходимо определить функции элементов ПМК, типы и число элементов КТС. При отсутствии готовых элементов требуется сформулировать ТЗ на разработку оригинальных составных частей комплексов.

На последующих иерархических уровнях нисходящего проектирования САПР осуществляется разработка оригинальных составных частей КТС и ПМК.

**Синтез структуры САПР.** Как и при проектировании других сложных систем, проектирование КТС САПР начинается с синтеза структуры комплекса. Поскольку синтез не удастся представить как совокупность полностью формализованных процедур, выполняемых по критериям, отражающим всевозможные требования ТЗ, при синтезе могут быть допущены те или иные ошибки, приняты неудачные решения. Для их выявления после синтеза идет процедура верификации. В отличие от синтеза верификация может быть автоматизирована, ее на ЭВМ реализуют с помощью имитационного моделирования КТС САПР как системы массового обслуживания.

Рассмотрим типичную последовательность процедур синтеза структуры САПР.

1. Построение обобщенных маршрутов проектирования, подлежащих автоматизации. Каждый обобщенный маршрут относится к одному определенному классу проекти-

руемых изделий и получается путем объединения индивидуальных маршрутов проектирования отдельных представителей этого класса. Примерами классов изделий могут служить большие интегральные схемы, панели и стойки РЭА, технологическая оснастка и приспособления для изготовления деталей из пластмасс и т. п. Обобщенные маршруты разрабатывают с той или иной степенью подробности. Например, маршрут проектирования БИС при наименьшей степени детализации может быть представлен в виде последовательности следующих этапов проектирования: функционально-логического, схемотехнического, топологического, контроля, проектирования фотошаблонов.

При большой степени детализации маршруты представляются состоящими из проектных процедур, например для БИС имеем: разработку алгоритма функционирования, абстрактный синтез конечного автомата, структурный синтез функциональной схемы, верификацию проектных решений функционально-логического проектирования, разбиение функциональной схемы, ее покрытие функциональными ячейками заданного базиса, размещение, трассировку, контроль соблюдения проектных норм и соответствия электрической и топологической схем, расслоение общего вида топологии, получение управляющей информации для фотонаборных установок. Возможна еще большая детализация маршрута с представлением проектных процедур совокупностями проектных операций, например структурный синтез функциональной схемы БИС можно разложить на следующие операции: поиск эквивалентных состояний конечного автомата, реализацию памяти, кодирование состояний, определение функций выхода и возбуждения элементов памяти, синтез комбинационной части схемы.

Структуру обобщенного маршрута, получающуюся при объединении индивидуальных маршрутов, удобно представить в виде орграфа, в котором вершины отображают проектные процедуры, а дуги — последовательность их выполнения. Возможные разветвления соответствуют альтернативным вариантам построения маршрутов. Например, схемотехническое проектирование выполняется по различающимся маршрутам в зависимости от того, является ли проектируемая БИС заказной или полузаказной, цифровой, цифроаналоговой или аналоговой и т. п. Схемотехническое проектирование может сводиться к покрытию функциональной схемы ячейками из заданного набора, но можно включать ряд операций от структурного синтеза при-

ципиальных схем функциональных узлов до их статистического анализа и верификации с учетом топологических параметров.

2. Выбор способа выполнения проектных операций и процедур в выявленных маршрутах. Возможны следующие способы: ручной (неавтоматизированный), автоматический (на ЭВМ в пакетном режиме) и автоматизированный (на основе взаимодействия ЭВМ и проектировщика).

3. Выбор структуры КТС, сводящийся к выбору числа уровней в структуре и базовых семейств ЭВМ.

Как правило, выбор семейства ЭВМ определяется сложившимися на предприятии или в отрасли традициями и накопленным опытом эксплуатации ЭВМ. Однако при соответствующем обосновании возможна переориентация на ЭВМ других типов. Одновременно принимается решение относительно использования тех или иных операционных систем для выбранного семейства или семейств ЭВМ.

4. Выбор ПМК из числа доступных для приобретения и предназначенных для выполнения процедур, выделенных в п. 2 в качестве объектов автоматизации. Очевидно, что выбор должен быть согласован с решениями, принятыми в п. 3, относительно типов ЭВМ и операционных систем.

Типична ситуация, когда существующие ПМК можно непосредственно применить для выполнения лишь части выделенных процедур, поэтому проводятся исследования, направленные на разработку недостающих компонентов математического обеспечения. Выясняются возможности и целесообразность модернизации и адаптации существующих ПМК к решению новых задач. В случае необходимости принимают решение о разработке новых компонентов программного и лингвистического обеспечения, в частности новых пакетов прикладных программ или ПМК.

Среди выбираемых или планируемых для разработки ПМК должны быть предусмотрены не только проектирующие, но и обслуживающие подсистемы программного обеспечения, например СУБД.

5. Распределение проектных процедур, фигурирующих в маршрутах проектирования, по уровням выбранной структуры КТС, дальнейшая конкретизация автоматизированного способа выполнения процедур, имеющая целью выяснение состава используемых периферийных устройств для ведения диалога и документирования результатов.

6. Назначение типов ЭВМ и АРМ в рамках выбранных семейств и определение их числа на основе ориентировоч-

ных оценок требуемой производительности технических средств. Этим заканчивается первая итерация синтеза структуры КТС.

**Моделирование САПР.** Исходные данные для моделирования можно разделить на три группы. К первой группе относятся сведения об объектах проектирования: интенсивности поступления заявок на проектирование  $\mu_i$  для всех  $q$  предполагаемых классов объектов  $i=1, 2, \dots, q$ ; распределения вероятностей параметров  $\alpha_i$ , где  $\alpha_i$  — оценка сложности проектируемого объекта в  $i$ -м классе. В качестве  $\alpha_i$  при моделировании нужно использовать размерность модели  $i$ -го проектируемого объекта. Вторую группу составляют данные об эффективности используемого программного обеспечения. Прежде всего к ним относятся зависимости требуемых объемов вычислений  $N_k$  и оперативной памяти  $P_k$  от размерности моделей  $\alpha_i$  для всех  $m$  основных проектных процедур,  $k=1, 2, \dots, m$ . Третья группа — характеристики выбранного вычислительного оборудования, — это данные о производительности устройств программной обработки данных, документирования, передачи данных о емкости внешних и оперативных запоминающих устройств, а также данные о структуре КТС, т. е. об объединении технических средств в программно-технические комплексы (ПТК) и о связях между ПТК.

Цель моделирования — проверка соответствия (верификация) результатов синтеза САПР ее назначению, выявление «узких мест» в синтезируемом варианте САПР и, следовательно, получение информации для обоснованных корректировок в проекте САПР. Эти корректировки касаются состава КТС и программного обеспечения проектирующих подсистем. Кроме того, при моделировании выбирают, оценивают и при необходимости корректируют дисциплину обслуживания заявок, которая при дальнейшем проектировании САПР должна быть реализована в мониторинговой системе.

Моделирование включает в себя формирование сетевой имитационной модели (СИМ), представляющей САПР как систему массового обслуживания, и выполнение численных экспериментов с этой моделью. Формирование СИМ осуществляется путем отображения структур маршрутов проектирования на синтезированную структуру КТС САПР. Уровни описания структуры КТС и маршрутов проектирования должны быть согласованы с характером и степенью полноты имеющихся исходных данных.

В СИМ САПР, как и при моделировании других систем массового обслуживания, имеются объекты двух типов: модели ресурсов (элементы КТС и в некоторых случаях элементы ПМК) и модели заявок (транзактов) — проекты, проходящие путь от ввода ТЗ до получения окончательного описания в виде комплекта конструкторской документации. Среди ресурсов выделяются устройства (обслуживающие аппараты), памяти, источники заявок и узлы. В устройствах происходит обработка данных. Математическая модель устройства представляет собой алгоритм, позволяющий рассчитать длительность обработки  $T_j$  входящих на вход заявок. Обычно длительность  $T_j$  рассчитывается как значение случайной величины, математическое ожидание которой есть функция производительности устройства, таких аргументов, как  $\alpha_i$  и  $N_k$ . Памяти отображают функцию хранения данных, их математические модели суть алгоритмы для расчета емкости памяти, требуемой входящей на вход заявке. Математические модели источников заявок — алгоритмы для расчета моментов появления заявок на входах системы массового обслуживания. Узлы служат для отображения в СИМ путей прохождения заявок в соответствии с маршрутами проектирования и принятым распределением проектных процедур по различным ПТК САПР. Узлы имитируют выбор одного из альтернативных путей прохождения заявок, т. е. отображают разветвления, а также такие операции, как размножение заявок, изменение их параметров, выход из системы и т. п. Параметрами заявок являются их имена, указатели на принадлежность к определенному классу, величины  $\alpha_i$  и т. п.

Для описания разрабатываемой СИМ, как правило, используют один из языков имитационного моделирования, например язык GPSS.

Вычислительный эксперимент с СИМ заключается в имитации поведения системы массового обслуживания на заданном отрезке времени или при обслуживании заданного количества заявок. Во время имитации источники заявок генерируют заявки, которые проходят в СИМ те или иные маршруты с задержками на обслуживание в устройствах, с задержками в очередях к занятым устройствам и памятьям, с выбором альтернатив в узлах разветвления по вероятностным критериям или по заданным условиям, с выходом из системы по окончании обслуживания или из-за превышения лимита времени на пребывание в системе и

т. п. В каждый момент модельного времени отображается состояние СИМ, выражаемое состоянием обслуживающих аппаратов (свободен или занят), памятьей (объем занятой памяти), длинами очередей на входах устройств и памятьей, местонахождением заявок и временем их пребывания в системе и т. п. По окончании моделирования очередного варианта САПР рассчитывают коэффициенты загрузки оборудования в КТС, средние и максимальные длины очередей, средние времена ожидания в очередях и обслуживания заявок и вероятности их обслуживания за время не выше заданного, производительность системы, понимаемую как количество заявок, обслуженных за единицу времени, и др. Перечисленные величины являются выходными параметрами САПР, им соответствуют заданные условия работоспособности. Установление факта выполнения условий работоспособности означает положительный итог верификации проекта САПР. В противном случае результаты моделирования позволяют выявить причины невыполнения условий работоспособности и внести в проект и в СИМ соответствующие изменения. После этого вновь производится имитация функционирования САПР.

Процесс итерационной обработки проекта САПР заканчивается получением варианта, принимаемого в качестве окончательного описания САПР на верхнем иерархическом уровне. Это описание содержит сведения о составе КТС (типы и число устройств) и программного обеспечения, включая перечни операционных систем и ПМК, технические задания на разработку оригинальных составных частей комплексов, а также маршруты проектирования и правила распределения проектных процедур по составляющим КТС.

Проектирование САПР должно осуществляться с учетом перспектив развития вычислительной техники и той области техники, для которой создается САПР. Поэтому САПР должна быть открытой для пополнения новыми техническими и программными средствами, а результаты прогноза объема проектных работ и размерностей решаемых задач должны учитываться в процессе имитационного моделирования.

**Пример построения фрагмента СИМ.** В качестве примера отображения структуры маршрута проектирования на структуру КТС рассмотрим формирование фрагмента СИМ для подсистемы схмотехнического проектирования БИС. На рис. 7.2 представлен граф принятого для реализации



маршрута проектирования функциональных ячеек (ФЯ) матричных БИС. На этом графе вершины соответствуют следующим процедурам и операциям:

1 — неавтоматизированному выполнению синтеза принципиальной электрической схемы ФЯ и принятию решений о путях достижения требований ТЗ;

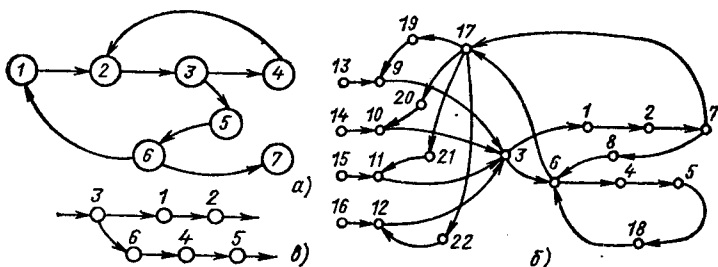


Рис. 7.2. Графы маршрута схмотехнического проектирования (а), сетевой имитационной модели (б) и ее фрагмент (в)

- 2 — детерминированному анализу ФЯ;
- 3 и 6 — проверке соответствия проекта требованиям ТЗ;
- 4 — изменению параметров или частичным модификациям структуры принципиальной схемы ФЯ;
- 5 — статистическому анализу ФЯ;
- 7 — оформлению данных для включения ФЯ в библиотеку.

Все процедуры, кроме 1, должны выполняться с помощью КТС САПР. Для КТС принята двухуровневая структура, включающая ЦВК и АРМ, связь между которыми осуществляется через связной процессор. Предполагается, что рассматриваемое АРМ имеет четыре рабочих места (РМ) и предназначено для обслуживания инженеров-схмотехников, а ЦВК связан со многими АРМ, реализующими маршруты приборно-технологического, функционально-логического и топологического проектирования.

Построение СИМ начнем с той ее части, которая соответствует вершине 2 на графе маршрута проектирования (ГМП) рис. 7.2, а. Отображение вершины 2 из ГМП на граф СИМ (ГСИМ) показано на рис. 7.2, в. Здесь вершины 1 и 2 отображают память и процессор ЭВМ, входящей в АРМ, соответственно. Но возможности ЭВМ в АРМ ограничены решением задач только детерминированного

анализа с размерностью моделей не выше некоторого предельного значения  $\alpha_1$ . Поэтому в ГСИМ вводится вершина 3 — это узел, в котором проверяется условие  $\alpha_i < \alpha_1$  ( $\alpha_i$  — размерность входящей в узел заявки); при его невыполнении заявка должна обрабатываться в ЦВК. Память и процессоры ЦВК в ГСИМ представлены вершинами 4 и 5. Передача данных между АРМ и ЦВК осуществляется через связной процессор, отображенный на рис. 7.2, в вершиной 6.

Граф СИМ в окончательном виде представлен на рис. 7.2, б. Он получен путем последовательного отображения вершин ГМП. Так, вершина 3 из ГМП представлена на рис. 7.2, б вершиной 7 в ГСИМ, из которой заявки могут либо возвращаться на соответствующее РМ, если условия работоспособности не выполнены, либо передаваться на вершину 6 и далее на вершины 4 и 5 для статистического анализа. Статистический анализ требует значительно больших затрат машинного времени, чем детерминированный, следовательно, заявка, передаваемая из вершины 7 на вершину 6, должна изменить значение одного из своих параметров, управляющих временем обслуживания в ЦВК. Изменение значения этого параметра происходит в узле, представленном в ГСИМ вершиной 8.

Вершина 4 из ГМП отображается в ГСИМ вершинами 9—12. Физически этим вершинам соответствуют РМ проектировщиков, основным устройством в которых является дисплей. На его экране высвечивается результат обработки данных в ЭВМ, с его помощью в диалоговом режиме проектировщик вводит необходимые коррективы в описание схемы и список значений параметров проектируемого объекта.

Вершине 1 из ГМП соответствуют вершины 13—16 в ГСИМ, они отображают источники заявок — запросы от сотрудников проектного подразделения на использование РМ после ручного синтеза схем ФЯ.

Вершина 5 из ГМП представлена вершинами 4 и 5 в ГСИМ, а вершина 6 из ГМП — вершиной 7 в ГСИМ. Кроме того, в ГСИМ включена вершина 17 — это узел, который распределяет заявки по РМ, через которые они ранее вошли в систему. Для осуществления такого распределения используют один из параметров заявки, приравненный номеру  $i$  РМ, через который данная заявка вошла в систему.

Вершина 18 — узел, предназначенный для отметки зая-

вок, вышедших из ЦВК. Эти отметки позволяют в вершине 6 правильно распределять поступающие заявки между альтернативными направлениями — «передача в АРМ (вершина 17)» или «передача в ЦВК (вершина 4)».

Вершины 19—22 отображают узлы, в которых заявки направляются либо на продолжение обслуживания, либо на выход из системы. Выход может произойти по двум причинам: проектирование ФЯ закончено или исследуемый вариант ФЯ бесперспективен и требуется синтез нового варианта, что происходит неавтоматизированным способом вне АРМ или ЦВК, т. е. происходит переход в вершину 1 ГМП.

Реализация остальных маршрутов проектирования, предусмотренных в создаваемой САПР, осуществляется через наращивание СИМ, выполняемое аналогичным образом.

После получения ГСИМ имеющиеся исходные данные привязывают к полученной модели, в частности устанавливают модели задержек в устройствах, правила распределения заявок или изменения их параметров в узлах.

Составление СИМ заканчивается переводом ГСИМ в программу имитации на принятом языке моделирования.

**Проектирование элементов САПР.** Некоторые из проектных процедур, формализация которых принципиально возможна и желательна, могут потребовать для своего выполнения имеющимися ПМК чрезмерно больших вычислительных ресурсов. Для другой части проектных процедур вообще могут отсутствовать ранее разработанные ПМК.

В таких ситуациях на основе результатов имитационного моделирования формулируются ТЗ на разработку новых ПМК или их элементов. Вопросы разработки сложных программных систем составляют содержание специальной технической дисциплины по программному обеспечению САПР.

Для процедур, многократно выполняемых в различных маршрутах проектирования и вносящих заметный вклад в общие затраты вычислительных ресурсов, прежде всего следует выбрать одну из двух возможных альтернатив: программную или аппаратную реализацию. Под аппаратной реализацией здесь понимается создание специализированной ЭВМ (спецпроцессоров), ориентированной на выполнение одной или нескольких родственных по характеру проектных процедур. Примерами спецпроцессоров для САПР могут служить спецпроцессоры для логического мо-

делирования, трассировки электрических соединений, быстрого преобразования Фурье и др. Повышение эффективности вычислений при аппаратной реализации алгоритма достигается с помощью распараллеливания или конвейеризации вычислений с учетом особенностей либо потока данных, либо потока команд, присущих решаемому классу задач. Возможности построения спецпроцессоров для САПР расширяются по мере развития технологии СБИС.

#### 7.5. ВЫБОР СТРУКТУРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ САПР

Программное обеспечение (ПО) совместно с информационным являются звеньями, связывающими все виды обеспечения САПР в единую систему проектирования.

К программному обеспечению САПР предъявляется ряд системных требований, от степени удовлетворения которых в значительной мере зависит эффективность всей системы автоматизированного проектирования, а именно: требование эффективной программной реализации алгоритмов, требование информационной согласованности программ друг с другом и с системой управления базы данных, требование модульности и наращиваемости ПО САПР.

Программное обеспечение САПР состоит из двух составных частей — общего и специального ПО. Общее ПО предназначено для планирования и организации процесса выполнения прикладных программ и определяется существующей номенклатурой операционных систем и комплексов программ технического обслуживания. Специальное ПО представляет собой пакеты прикладных программ, реализующих алгоритмы выполнения проектных операций и процедур. Эти прикладные программы необходимы группе пользователей, занимающихся автоматизированным проектированием в конкретной проблемной области.

**Функции и состав общего программного обеспечения.** Под операционной системой (ОС) ЭВМ понимают комплект системных программ, управляющих всеми ресурсами системы: оперативной памятью, временем работы процессора, каналами обмена данными и периферийными устройствами, программными модулями и др. Рассмотрим функции и состав ОС на примере распространенной операционной системы ЕС ЭВМ — ОС ЕС.

Операционная система ЕС предназначена для автома-

тизации обработки данных на вычислительных системах, построенных на основе моделей ЕС ЭВМ; увеличивает пропускную способность вычислительной системы; уменьшает время реакции системы и повышает надежность ее функционирования. Это достигается благодаря непрерывной обработке потоков заданий, их размещению в системных очередях, предварительному планированию и иницированию выполнения очередных заданий.

Основными компонентами ОС ЕС являются управляющие и обрабатывающие программы.

Управляющие программы ОС ЕС. Управляют вычислительным процессом, реализуют функции управления задачами, заданиями и данными; следят за выполнением обрабатывающих программ и прикладных программ, составленных пользователем.

*Программа управления задачами (супервизор)* управляет всем ходом вычислительного процесса в ЭВМ и предназначена для наблюдения и управления всеми задачами в системе. Супервизор выделяет задачам необходимые ресурсы для их решения в одно- или мультипрограммном режиме.

В однопрограммном режиме образуется только одна задача, в рамках которой выполняются как программы ОС, так и программы пользователей, при этом все технические средства ЭВМ находятся в распоряжении этой задачи.

При необходимости одновременного решения нескольких не зависящих друг от друга задач супервизор реализует либо мультипрограммный режим, либо режим разделения времени. В этом случае супервизор производит переключение с решения одной задачи на решение другой, осуществляет динамическое распределение памяти и других ресурсов ЭВМ, обрабатывает через систему прерываний поступающие запросы.

Одновременное выполнение нескольких задач в мультипрограммном режиме достигается благодаря тому, что при их выполнении всегда возникают паузы, связанные с ожиданием завершения операции ввода и вывода, с истечением заданного интервала времени, с ожиданием дополнительных данных для продолжения выполнения программы и др. На период ожидания выполнение данной задачи откладывается и процессор переходит к выполнению следующей задачи и т. д. Для оператора эти программы выполняются одновременно, при этом ОС обеспечивает динами-

ческое распределение памяти и возможность работы по приоритету со многими задачами и заданиями.

*Программы управления заданиями (планировщик заданий и главный планировщик)* выполняют функции планирования вычислительного процесса. Задание представляет собой внешнюю независимую единицу работы, выполняемой ОС. Задания состоят из пунктов или шагов, связанных между собой прямо или косвенно. Выполнение одного пункта задания — это выполнение некоторой обрабатываемой программы.

Управление прохождением одного или нескольких заданий через вычислительную систему осуществляют программы управления заданиями. Планируя структуру задания, программист может использовать любые программы, которые включены в библиотеку ОС. Перед обработкой задание объединяется с другими, образуя один входной пакет заданий.

Планировщик состоит из планировщика заданий и главного планировщика. Планировщик заданий подготавливает задания для выполнения и планирует их выполнение последовательно или с учетом приоритетов. Главный планировщик устанавливает связь оператора с ОС, выводит сообщения на консоль оператора и вводит с консоли его команд.

Язык управления заданиями содержит девять типов операторов: задания (JOB), исполнительный (EXEC), описания данных (DD), командный, разделительный, пустой, процедуры (PROC), конца процедуры (PEND), комментариев. Эти операторы имеют соответствующие параметры, которые позволяют планировщику заданий регулировать выполнение заданий и пунктов заданий, отыскивать и размещать данные, распределять ресурсы ввода/вывода, а главному планировщику — осуществлять связь с оператором вычислительной системы.

*Программы управления данными* предназначены для организации хранения, поиска данных и обмена данными между оперативной памятью и внешними устройствами.

Операционная система ЕС допускает четыре типа организации данных: последовательную, индексно-последовательную, прямую и библиотечную.

При последовательной организации наборов данных записи подвергаются обработке в порядке, совпадающем с их физической последовательностью на внешнем носителе информации.

Индексно-последовательная организация представляет возможность адресоваться непосредственно к записи без считывания предыдущих записей или без информации о расположении данной записи.

Прямая организация наборов данных допускает возможность произвольной обработки записей.

Наборы данных с библиотечной организацией состоят из последовательно организованных разделов, каждый из которых имеет свое имя.

Последовательные наборы данных могут быть организованы на любом внешнем носителе, а остальные вышеперечисленные способы организации наборов данных — только на устройствах прямого доступа (магнитных дисках или барабанах).

Все имеющиеся в системе наборы данных зарегистрированы в каталоге системы. Каталогизация позволяет обращаться к наборам данных только по имени, уменьшает степень ручного вмешательства в процессы идентификации и хранения информации, сводит к минимуму ошибки обслуживающего персонала. Каталог системы устроен подобно оглавлению библиотеки и расположен в устройствах прямого доступа.

**Обрабатывающие программы.** Служат для подготовки прикладных программ к исполнению на ЭВМ. Подготовка к исполнению включает в себя трансляцию исходной программы, ее редактирование и загрузку. К обрабатывающим программам относятся трансляторы с основных алгоритмических языков (ФОРТРАН, ПЛ/1, ПАСКАЛЬ, АЛГОЛ и др.), редактор связей, программы отладки и вспомогательные программы (утилиты).

**Трансляторы** предназначены для перевода текста исходных программ (модулей) на машинный язык, построения объектного модуля в форме, удобной для его последующей обработки, вывода на печать листингов исходных программ, а также диагностического сообщения о лексических, синтаксических и семантических ошибках.

**Объектный модуль** представляет собой последовательность машинных команд. Он не содержит стандартных подпрограмм, необходимых для работы программы пользователя. **Исходный модуль** в простейшем случае размещается на перфокартах, а объектный модуль и промежуточные наборы данных — на магнитных дисках.

Для организации машинной реализации к объектному модулю необходимо добавить недостающие стандартные

программы. Эту работу называют *редактированием*, а соответствующую программу — *редактором связей* или *просто редактором*. Недостающие стандартные программы редактор связей берет из библиотеки стандартных подпрограмм. Редактор создает из объектного модуля и стандартных подпрограмм *загрузочный модуль*, который может рассматриваться в качестве выполняемой машинной программы. Загрузка загрузочного модуля в оперативную память для непосредственного исполнения осуществляется программой выборки.

После получения загрузочного модуля происходит выполнение программы пользователя. В процессе выполнения программы ОС контролирует ее выполнение, а результаты контроля выводит на алфавитно-цифровое печатающее устройство. Взаимосвязь функционирования обрабатывающих программ в ОС ЕС представлена на рис. 7.3.

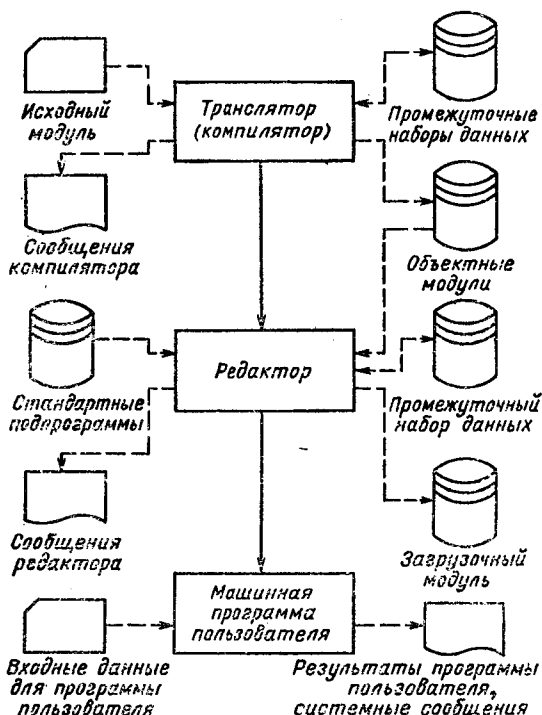


Рис. 7.3. Взаимосвязь функционирования обрабатывающих программ в ОС ЕС



К обрабатываемым программам относятся также программа сортировки-объединения для организации расположения информации во внешней памяти, программы-утилиты для преобразования набора данных и редактирования его записей, для копирования набора данных с носителя на носитель, перегруппировки записей и упорядочения массивов для распечатки сведений об имеющихся в библиотеках программах и данных и др.

Операционная система имеет модульную структуру, которая позволяет приспособлять систему к конкретным конфигурациям используемых технических средств. Процесс создания конкретной структуры ОС, учитывающей особенности конфигурации ЭВМ и задач потребителя, называют генерацией системы.

**Функции и состав специального программного обеспечения.** В состав программного обеспечения САПР входят пакеты прикладных программ (ППП), ориентированные на решение определенных задач проектирования и реализуемые как надстройка над ОС. Основу ППП составляет множество программных модулей, каждый из которых является программой реализации определенной проектной процедуры либо программой реализации некоторого алгоритма (или фрагмента алгоритма) проектирования.

Специальное ПО САПР может иметь собственную ОС или же использовать одну из базовых ОС ЭВМ. Программное обеспечение с собственной ОС имеет сложную структуру. В состав такого ПО входят универсальный или специализированный монитор САПР, организующий вычислительный процесс в соответствии с принятым алгоритмом проектирования; транслятор или интерпретатор с входного языка; набор программных модулей, составляющих тело ППП; набор обслуживающих программ и т. п. Типовая структура ПО САПР представлена на рис. 7.4.

Обращение к специальному ПО реализуется в форме запроса, который содержит требования к выполнению проектных процедур и исходные данные для проектирования. Последовательность подключения программных модулей для выполнения проектных процедур определяется *программой-монитором САПР*. Взаимосвязь отдельных программных модулей обеспечивает возможность организации сложных маршрутов проектирования.

Непосредственную реализацию проектных процедур производят с помощью *программных модулей*. Каждый программный модуль в свою очередь может разделяться

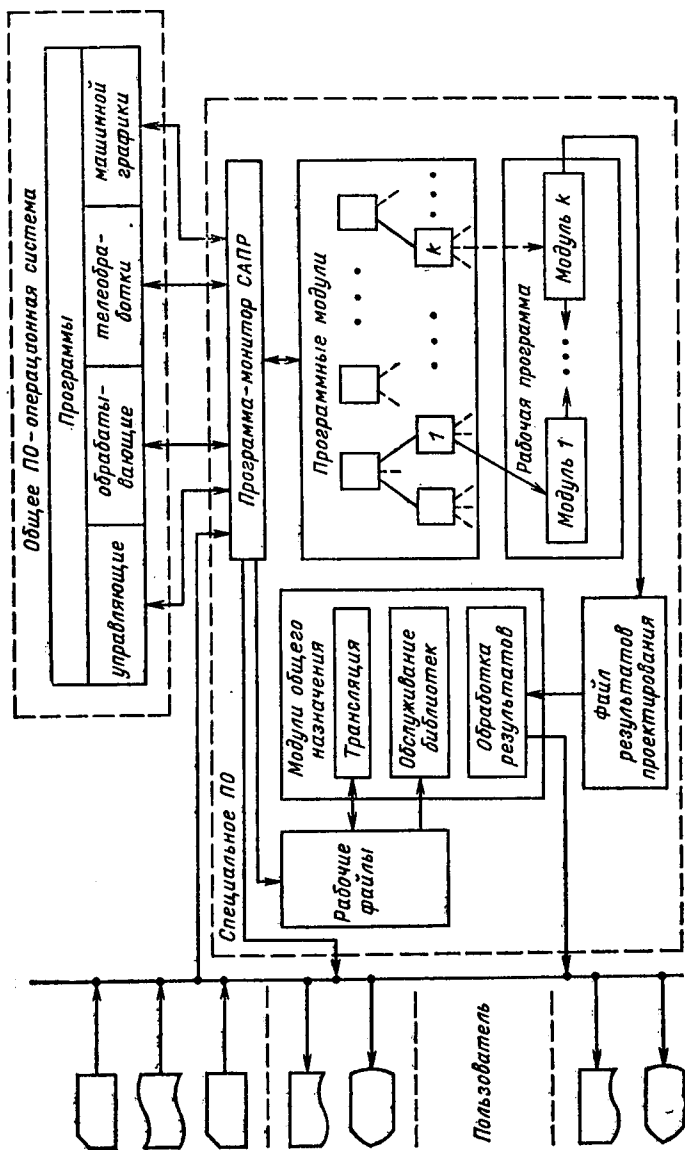


Рис. 7.4. Типовая структура ПО САПР

на более мелкие модули. В подобной структуре должны быть четко определены функции каждого программного модуля. Программы специального ПО группируются в ППП, каждый из которых проблемно ориентирован на обслуживание задач отдельных подсистем САПР. Поэтому в состав ППП входят как программные модули, ориентированные на реализацию узкоспециализированных проектных процедур и, как правило, создаваемые разработчиками конкретных САПР, так и модули общего назначения (оптимизации, математических моделей используемых элементов и т. п.). Модули общего назначения иногда называют *библиотечными модулями*.

В соответствии с алгоритмом проектирования для каждого нового объекта и маршрута его проектирования автоматически составляются рабочие программы. В зависимости от требований ТЗ и конкретных условий на каждом этапе разработки могут использоваться различные маршруты проектирования и соответственно различные рабочие программы.

Связь программных модулей осуществляется с помощью *буферных файлов*. В САПР используется два типа буферных файлов — рабочие и результатов проектирования. В рабочих файлах запоминается информация о структуре проектируемого объекта, типе и параметрах составляющих его элементов, в файле результатов накапливаются результаты автоматизированного проектирования, которые после обработки документируются и выводятся на устройства отображения информации.

Модульная структура ПО САПР позволяет легко наращивать функциональные возможности системы путем расширения набора программных модулей и их модификаций, а также позволяет пользователю легко развивать систему.

Многообразие языков программирования, сложность проектных процедур и разнообразие вариантов маршрутов проектирования требуют концентрации усилий разработчиков специального ПО САПР. Цикл разработки программного обеспечения включает в себя: анализ требований, предъявляемых к САПР; определение точного описания функций и проектных процедур (спецификаций), реализуемых с помощью ПО; разработку алгоритмов реализации функций, проектных процедур; программных модулей с использованием алгоритмических языков высокого уровня и методов структурного программирования; тестирование программ; эксплуатацию и сопровождение.

На первом этапе разработки ПО определяются требования, выполнение которых позволяет получить требуемое решение поставленной проблемы. Анализ требований способствует лучшему пониманию проблемы и компромиссных ситуаций, что помогает выбору наилучшего решения.

На этапе определения спецификаций задаются структура входных и выходных данных, возможные типы проектных процедур и маршрутов проектирования. Чем детальнее разработаны проектные процедуры и маршруты проектирования, тем меньше вероятность возникновения ошибок и тем легче организовать информационный интерфейс между программными модулями.

При организации разработки ПО целесообразно структуру пакета прикладных программ разбивать на отдельные по возможности независимые части таким образом, чтобы ответственность за реализацию каждой такой части можно было возложить либо на одного разработчика, либо на группу. При этом для каждой части ППП должны быть сформулированы предъявляемые к ней требования: реализуемые проектные процедуры, размерность и др.

Проверку корректности ПО и его отладку производят на этапе тестирования. Тестирование подразделяют на три стадии: автономное, комплексное и системное. При автономном тестировании каждый программный модуль проверяют с помощью данных, подготавливаемых программистом. Модуль, прошедший автономное тестирование, подвергают комплексному тестированию, при котором проверяют отдельные группы программных модулей. В результате комплексного тестирования возможно обнаружение ошибок, пропущенных при автономном тестировании. При системном тестировании испытывают ППП с помощью независимых тестов.

Практика разработки ПО показывает, что наибольшее число ошибок связано с разработкой информационного интерфейса. Большие трудности возникают при разработке интерфейса между программными модулями, написанными разными программистами. Поскольку число таких интерфейсов при  $N$  исполнителях составляет  $N(N-1)/2$  и возрастает пропорционально квадрату числа исполнителей, проблема становится весьма сложной при разработке ПО группой из нескольких человек, так как взаимодействие программистов друг с другом снижает производительность их труда и требует дополнительных затрат на тестирование. Решение вопросов унификации и стандартизации дол-

жно быть основополагающим при разработке специального ПО САПР.

**Мониторы и языковые процессоры.** К наиболее сложным программам САПР относятся мониторы ППП и языковые процессоры.

*Монитор* — это управляющая программа диалогового взаимодействия пользователя с комплексом технических средств САПР. В функции монитора входит: управление работой программного канала обмена с периферийными устройствами; создание различных режимов работы комплекса технических средств и управление ими; загрузка абсолютных программ (загрузочных модулей, настроенных на конкретные адреса) и управление ходом их выполнения; выбор программных модулей в соответствии с маршрутом проектирования; настройка связей выбранных программных модулей; распределение памяти и т. д.

*Языковые процессоры* предназначены для преобразования информации, выраженной на входном языке, в рабочую программу. Они могут генерировать рабочую программу способами *компиляции* или *интерпретации*, их соответственно называют компиляторами и интерпретаторами.

Компилятор осуществляет лексический и синтаксический анализ для трансляции программы с языка высокого уровня, выполняет основную работу по составлению объектной программы — генерирует объектные модули и команды обращения к библиотечным модулям, а также формирует управляющие предложения для редактора связей относительно состава используемых объектных модулей для формирования загрузочного модуля.

**Организация диалоговых процедур.** Для осуществления непосредственного контакта разработчика с ЭВМ в процессе автоматизированного проектирования предусмотрены диалоговые (интерактивные) режимы проектирования. В диалоговом режиме осуществляется управление алгоритмами, данными и различными подсистемами и программами САПР.

Управление алгоритмами включает в себя поиск и изменение алгоритмов проектирования, изменение проектных процедур, изменение и коррекцию процедур оптимального параметрического анализа и критериев прекращения поиска экстремума в задачах оптимизации.

Управление данными предполагает изменение параметров моделей проектируемых объектов в процессе поиска оптимального решения, изменение критериев и ограниче-

ний с целью изменения модели для получения достоверных результатов проектирования.

Управление подсистемами и программами САПР осуществляется с целью варьирования разработчиком отдельных параметров проектируемого изделия с целью анализа выбранного технического решения, выбора способа представления графической информации и конструкторской документации, а также улучшения определенных технико-экономических характеристик проектируемого объекта путем ослабления несущественных ограничений.

Диалоговый режим наиболее эффективен при подготовке данных и при решении различных оптимизационных задач.

Наиболее распространенным и удобным устройством для диалоговых систем проектирования является экранный пульт (дисплей), связанный с каким-либо устройством документирования. Дисплеи снабжены устройствами обратной связи в виде следящего перекрестия и светового пера, а также функциональной клавиатурой, позволяющей оперировать как с алфавитно-цифровой информацией, так и с графическими изображениями. Поэтому в состав комплексов технических средств САПР для организации диалогового взаимодействия включают мини- или микро-ЭВМ, обеспечивающие управление работой комплекса и реализацию функциональных программ обработки графической информации, устройства вывода и автоматического и полуавтоматического ввода графической информации (кодировщики) и устройства оперативного графического взаимодействия разработчика с ЭВМ (дисплеи).

Графическая информация на экране дисплеев формируется с помощью генератора векторов, который позволяет создавать изображения как векторов различной длины и направления, так и точек.

Программное обеспечение машинной графики в ОС включает в себя (рис. 7.5): графический метод доступа, средства разработки программ графических приказов, проблемно-ориентированные программы, пакет графических подпрограмм.

В настоящее время широко распространены автономные диалоговые комплексы — автоматизированные рабочие места проектировщиков (АРМ), которые представляют собой совокупность технических и программных средств для организации диалоговых режимов проектирования. В зависимости от назначения АРМ выпускают в

различных конфигурациях и с различным набором системных и прикладных программ (например, АРМ—Р — для разработчиков радиоэлектронной аппаратуры, АРМ—М — для автоматизированного проектирования в машиностроении и др.). Автоматизированное рабочее место может сты-

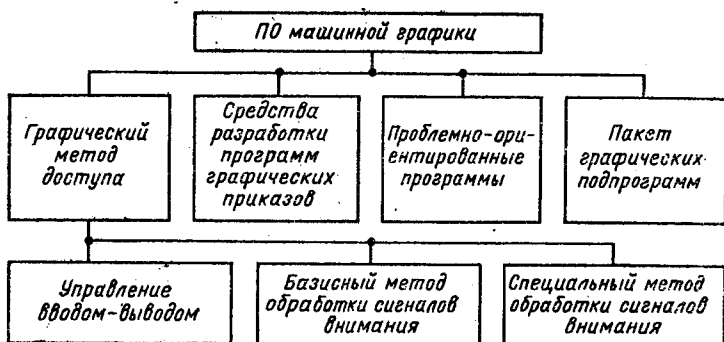


Рис. 7.5. Программное обеспечение машинной графики в операционной системе

коваться с ЭВМ более высокого уровня (например, ЕС ЭВМ) и в этом случае используется в качестве интеллектуальных терминалов.

#### 7.6. РОЛЬ И МЕСТО САПР В ГИБКИХ АВТОМАТИЗИРОВАННЫХ ПРОИЗВОДСТВЕННЫХ СИСТЕМАХ

Одним из основных условий технического прогресса является постоянное расширение и обновление номенклатуры выпускаемой продукции, а одним из главных требований к современному производству — обеспечение возможности проектирования, создания и освоения новой высококачественной продукции в кратчайшие сроки при минимальных затратах. Выполнение этих требований невозможно без крупномасштабной автоматизации на основе ЭВМ, для реализации которой необходим коренной пересмотр организационно-экономических и технологических характеристик производственной деятельности в направлении создания динамичных и интенсивных форм производства.

Главной особенностью решения проблемы интенсификации является то, что производится не интенсификация фи-

зического труда человека, которая практически исчерпала себя, а интенсификация практически неограниченного интеллектуального труда человека, использующего широкие возможности современных ЭВМ.

Основной стратегией по проведению крупных мероприятий по совершенствованию технической и технологической базы в промышленности, а также использовании новых методов организации производства являются широкое использование систем автоматизированного проектирования во всех сферах проектирования и производства и создание промышленной робототехники и гибких автоматизированных средств вычислительных систем, в которых современные средства вычислительной техники занимают в функциональном отношении центральное место.

Исторически сложилось так, что первоначально достаточно автономно развивались сферы автоматизации обработки информации [автоматизированные системы управления АСУ, САПР и др.] и автоматизации технологической подготовки производства (промышленные роботы, технологическое оборудование с ЧПУ, АСУ ТП и др.). Проектировщики разрабатывали изделия и детали с помощью САПР, а затем представляли чертежи в производство для технологической подготовки и изготовления изделия. Практика показала, что автоматизация обработки информации в отрыве от автоматизации технологии не приводит к существенной интенсификации производства.

Успехи, достигнутые в последние годы в области микроэлектроники, открыли принципиально новые возможности для осуществления высокоэффективной автоматизации производственных процессов, проектно-конструкторских и научно-исследовательских работ. Широкое внедрение мини- и микро-ЭВМ с разнообразным современным периферийным оборудованием позволило создать системы распределенной обработки информации, на основе которых строят интегрированные системы управления, получившие название гибких автоматизированных производств (ГАП). Компонентами ГАП являются САПР, АСУ ТП с использованием ЭВМ и числового программного управления, АСУ производством (АСУП) и средства промышленной робототехники. Создание таких производств связано с коренной перестройкой управления производственной технологией на основе крупномасштабной автоматизации со сквозным применением средств вычислительной техники и роботизированных средств автоматизации, включая автоматизиро-



ванное технологическое оборудование, транспортные системы, складское хозяйство и др. Внедрение ГАП дает возможность предприятиям разрабатывать и изготавливать продукцию от начала до конца под управлением комплекса средств вычислительной техники.

Важные достоинства ГАП — высокое качество и снижение сроков проектирования, уменьшение ручного труда и улучшение параметров технологических процессов при производстве изделий новой техники. Кроме того, с использованием совершенных программ контроля автоматизированного проектирования и производства изделий уменьшается число ошибок. Современные ГАП предполагают создание и использование централизованных баз данных, обеспечение простоты доступа к ним и широких возможностей документирования.

Гибкое автоматизированное производство представляет собой организационно-техническую производственную систему, позволяющую в средне- и мелкосерийном номенклатурном производстве в короткий срок и с минимальными затратами заменить выпускаемую продукцию на новую путем перестройки технологического процесса за счет замены управляющих программ; обеспечивает быструю перестройку технологии производства на выпуск новых изделий путем интеграции САПР, АСУ ТП, АСУП и средств робототехники.

Типовая структура ГАП приведена на рис. 7.6. Главная особенность структуры — комплексное использование ЭВМ как для этапов автоматизированного проектирования, конструирования, планирования и технологической подготовки производства, так и для этапов автоматизации технологических процессов изготовления, контроля и складирования продукции. При этом вопросы автоматизированного проектирования и моделирования принимают в ГАП принципиально новое значение. Если в традиционном производстве вопросами автоматизированного проектирования и моделирования занимались отдельные организации или подразделения в отрыве от самого производства, то в ГАП моделирование становится неотъемлемой частью производственного процесса, поскольку любая перестройка в ГАП требует анализа и моделирования в процессе эксплуатации. Высокий уровень и широкая номенклатура САПР технологических процессов различных видов позволяют повысить уровень автоматизации ГАП, а также улучшить их адаптацию к изменяющимся условиям производства.

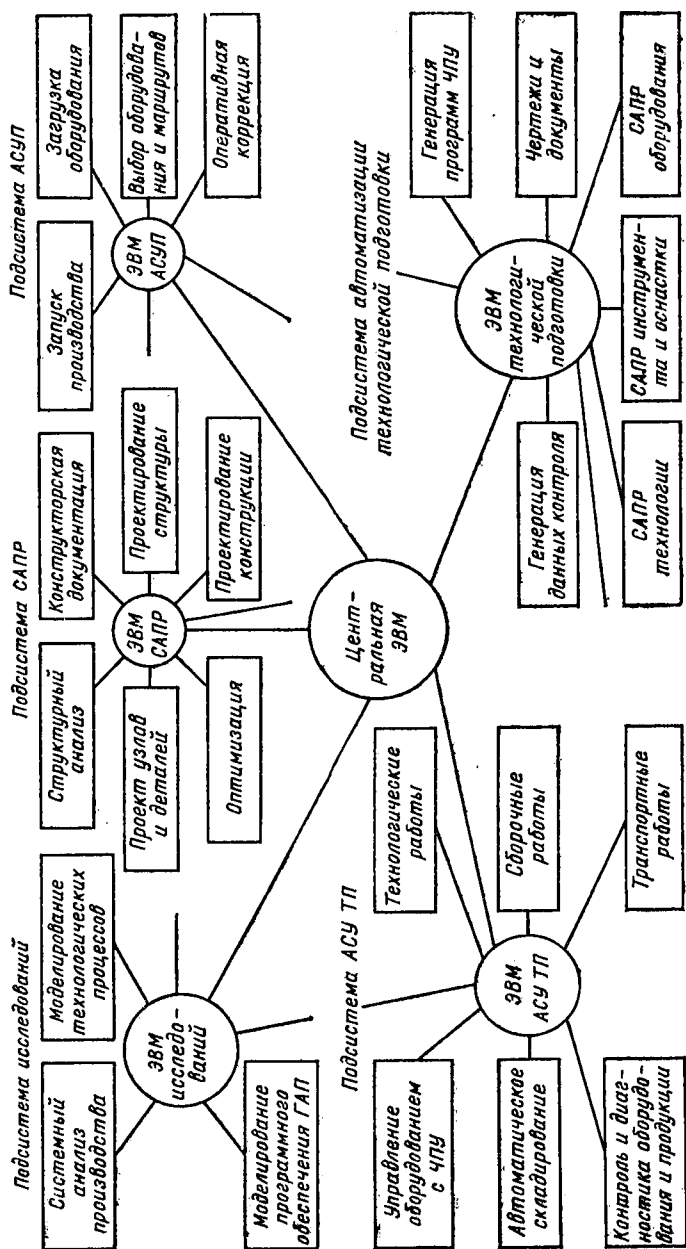


Рис. 7.6. Типовая структура гибких автоматизированных производств

Рассмотрим структурную схему ГАП. Информационно-вычислительной базой для ГАП являются локальные вычислительные сети, создаваемые на базе больших ЭВМ (серии ЕС или «Эльбрус»), мини- и микро-ЭВМ, а также разнообразного терминального оборудования. На базе сети реализуется ряд подсистем ГАП.

Подсистема исследований предназначена для осуществления системного анализа производства, моделирования основных технологических процессов и программного обеспечения ГАП.

Подсистема САПР производит автоматизированное проектирование структуры и конструкции изготавливаемых изделий, разрабатывает программы ЧПУ и выпускает конструкторскую документацию.

Подсистема АСУП осуществляет оптимальный допуск, загрузку и регулирование хода производства с учетом плановых заданий и распределения загрузки ресурсов системы.

Подсистема автоматизации технологической подготовки производства включает в себя САПР технологии, САПР управляющих программ, САПР инструмента и оснастки, генерирует исходные данные для контроля и измерений в процессе производства.

Подсистема АСУ ТП управляет оборудованием с ЧПУ по изготовлению оснастки, деталей и узлов изделий, управляет автоматическим складированием деталей и узлов, управляет транспортной сетью, технологическими, сборочными и транспортными роботами, осуществляет контроль и диагностирование продукции и технологического оборудования, обеспечивает надежность функционирования производственной системы в целом.

Анализируя роль каждой подсистемы в процессе функционирования ГАП, можно прийти к выводу, что для получения высокого технико-экономического эффекта от использования ГАП прежде всего требуется высокий уровень развития и внедрения САПР, что подразумевает: новый, более высокий уровень организации сквозной САПР технических объектов, включающий в себя все этапы и уровни проектирования объектов, в том числе и САПР технологической подготовки производства; повышение качества и надежности функционирования всех компонентов САПР, что должно обеспечить бездефектное проектирование объектов новой техники и их производства; разработку и организацию автоматизированных систем информационного обеспечения, предназначенных для сбора и обмена

информацией как внутри каждой отрасли, так и между отраслями.

### 7.7. СОЦИАЛЬНОЕ ЗНАЧЕНИЕ И ПЕРСПЕКТИВЫ РАЗВИТИЯ САПР

Автоматизация проектирования входит неотъемлемой составной частью в приоритетные направления научно-технического прогресса — электронизацию и комплексную автоматизацию машиностроительного и приборостроительного производства. От успехов в создании и развитии САПР во многом зависят возможности и сроки разработки образцов новой техники, внедрение интегрированных автоматизированных производств, рост производительности труда инженерно-технических работников, занятых проектированием. Совершенствование САПР ведется в ряде направлений.

**Техническое обеспечение САПР.** Разнообразие проектных задач, решаемых в САПР крупных предприятий на различных иерархических уровнях, заставляет использовать многоуровневую структуру технических средств.

Существует ряд задач, строгое решение которых в автоматическом режиме находится за пределами возможностей современных вычислительных средств. Примеры таких задач — нестационарные трехмерные задачи математической физики и NP-полные комбинаторные задачи. Для их решения предпринимается усилие как в направлении поиска более эффективных математических моделей и методов, так и в направлении построения и применения супер-ЭВМ, обладающих производительностью в несколько сотен миллионов операций в секунду и выше. Наиболее известными примерами супер-ЭВМ, созданных в начале 80-х годов, являются Cyber-205 и Cray-X—MP/48, производительность которых достигает 0,8 и 1,6 млрд. операций в секунду соответственно. В основе достижения столь высокой производительности лежит одновременная обработка нескольких потоков данных, конвейерная обработка или совместное использование обоих способов организации параллельных вычислений. Предполагается в ближайшие годы разработка в странах — членах СЭВ супер-ЭВМ с быстродействием около 10 млрд. операций в секунду. Однако стоимость супер-ЭВМ велика (для упомянутых супер-ЭВМ около 20 млн. долларов) и потому в большинстве САПР в центральных вычислительных комплексах будут применяться ЭВМ высокой производительности (до 100 млн. операций в секунду) из семейств Эльбрус и ЕС ЭВМ.

Средний уровень в структуре технического обеспечения САПР будет представлен в основном супермини-ЭВМ, имеющими 32-разрядную сетку, быстродействие в сотни тысяч — единицы миллионов операций в секунду и емкость оперативной памяти в единицы — десятки мегабайт. Машины этого уровня смогут взять на себя функции управления работой САПР, хранения и функционирования базы данных, выполнение большинства проектных процедур.

Для непосредственной связи с проектировщиком получат развитие рабочие места на базе супермикро-ЭВМ с длинной машинного слова в 32 двоичных разряда, с быстродействием в сотни тысяч операций в секунду и с емкостью оперативной памяти до нескольких мегабайт. Подобные рабочие места будут снабжены растровыми дисплеями, позволяющими проектировщику обмениваться с ЭВМ как символьной, так и графической информацией с возможностями многоцветной многооконной визуализации проектных решений.

Для облегчения доступа к разделяемым информационным и программным ресурсам рабочие места проектировщиков должны объединяться в локальные вычислительные сети, что будет способствовать реализации принципов сквозного автоматизированного проектирования.

Одним из наиболее перспективных путей развития технического обеспечения САПР является разработка и применение специализированных процессоров или ЭВМ, ориентированных на выполнение однотипных трудоемких проектных процедур. Выше (стр. 254) говорилось о специализированных ЭВМ для логического моделирования, позволяющих ускорить решение задач моделирования на несколько порядков. Другими примерами специализированных процессоров или ЭВМ для САПР служат трассировочные машины, процессоры для быстрого преобразования Фурье, процессоры графических процедур. Известны и такие специализированные процессоры, как процессоры СУБД, процессоры для ускорения выполнения матричных операций и т. п. Актуальность построения специализированных процессоров для САПР обусловлена наличием трудоемких вычислительных процедур, увеличением сложности решаемых задач, а возможности построения таких процессоров расширяются в связи с появлением СБИС, средств их проектирования и изготовления, с дальнейшим ростом степени интеграции микросхем.

**Математическое обеспечение САПР.** Наиболее научно-

емкое и труднопрогнозируемое направление развития САПР — это совершенствование математического обеспечения. В этом направлении проводятся научно-исследовательские работы, закладывающие фундамент автоматизации проектирования, определяющие и реализующие возможности формализации проектных процедур.

Важное место в маршрутах проектирования отводится процедурам моделирования и верификации. Принципиальные основы математического обеспечения этих процедур разработаны и реализованы во многих программно-методических комплексах. Однако трудности, связанные с высокой размерностью задач, не преодолены. В дополнение к упомянутым выше задачам математической физики можно назвать еще ряд задач, для которых имеющиеся средства недостаточны для получения удовлетворительного решения. Это, например, задача верификации конструкторских решений блоков радиоэлектронной аппаратуры, решаемая с учетом перекрестных помех и эффектов длинных линий; задача верификации результатов проектирования функциональных схем ЭВМ, связанная с выбором тестов и учетом возможных логических и динамических дефектов; задача анализа нелинейных устройств типа автогенераторов, источников вторичного электропитания, схем модуляции-демодуляции, требующих моделирования электрических процессов на протяжении многих периодов колебаний и др. Наряду с развитием таких подходов к ускорению моделирования, как диакоптика, макро моделирование, адаптация, комбинирование моделей и методов, учет пространственной и временной разреженности, рассмотренных выше, перспективны работы по созданию методов структурной верификации функциональных схем, по разработке параллельных алгоритмов различных проектных процедур, по поиску эффективных алгоритмов отдельного интегрирования систем обыкновенных дифференциальных уравнений и т. п.

Но наиболее актуальны усилия по созданию и развитию формальных методов структурного синтеза. Как правило, в автоматическом режиме удается решать лишь немногие задачи этого типа. К ним относятся, например, задачи коммутационно-монтажного проектирования печатных плат, проектирования топологии матричных БИС, синтеза тестов для цифровых устройств умеренной сложности. Однако практические потребности в оперативном проектировании разнообразных устройств вычислительной техники

и автоматики остро поставили проблему создания кремниевых компиляторов — систем автоматического проектирования БИС и СБИС.

В кремниевых компиляторах в качестве исходных данных задается либо описание алгоритма, который должна реализовать СБИС и который представлен в виде некоторой микропрограммы, либо описание схемы на языке уровня регистровых передач. Результатом работы кремниевого компилятора должно быть описание топологии кристалла, выдаваемое в форме управляющей информации для оборудования, изготавливающего фотошаблоны слоев СБИС. Все операции по преобразованию исходных данных в окончательный результат выполняются автоматически: это разбиение исходного описания на фрагменты, трансляция фрагментов исходной информации в фрагменты функциональной схемы и далее в фрагменты топологической схемы, выбираемые из заранее разработанного набора типовых ячеек, трассировка межсоединений, перевод топологии в управляющую информацию для фотонаборных установок. Библиотеки типовых ячеек тщательно отрабатываются предварительно с помощью средств автоматизации схемотехнического и топологического проектирования. Кремниевая компиляция уступает по показателю использования площади кристалла, но выигрывает по оперативности и стоимости проектирования по сравнению с автоматизированным проектированием СБИС.

Кремниевые компиляторы являются примером специализированных систем автоматического проектирования, основанного на адаптации множества возможных проектных решений к имеющимся средствам проектирования. Эта адаптация заключается во введении элементов унификации и регулярности в проектируемые структуры.

Другим направлением автоматизации структурного синтеза является создание экспертных систем. Экспертная система — искусственная система, выполняющая функции эксперта в конкретной предметной области. Для создания экспертной системы необходимо решить вопросы, каким образом будут формализованы знания эксперта, на каком языке будут задаваться запросы экспертной системе и каким образом знания эксперта будут извлекаться и использоваться для конструирования ответов на запросы.

В большинстве созданных экспертных систем знания представляются в виде конечного множества правил, называемых продукциями. Продукция по своей структуре

напоминает условный оператор алгоритмического языка, в соответствии с которым выполняются некоторые действия, если выполнено указанное в продукции условие. Совокупность продукций составляет основу базы знаний.

При работе экспертной системы продукции выбираются в определенном порядке из базы знаний в соответствии с некоторой управляющей структурой. Такая структура может быть представлена в виде графа (сети) или дерева, отражающих взаимосвязи между компонентами проектных решений в данной предметной области. Управляющая структура может быть воплощена в самих продукциях или быть отделенной от них. Выбор конкретного маршрута в управляющей структуре, т.е. выбор последовательности продукций, порождающих проектные решения, зависит от исходных данных, указанных в задании при обращении к экспертной системе. Эти исходные данные в сочетании с данными об условиях проектирования и текущем состоянии проекта, хранящимися в базе данных, позволяют присваивать конкретные значения переменным, фигурирующим в продукциях. Становится возможной проверка истинности условий, входящих в продукции, по результатам проверки активизируются действия в соответствующих продукциях, в том числе осуществляются переходы по сети между продукциями.

Типичные составные части экспертных систем — банк данных, банк знаний, подсистема связи с пользователем, подсистема логического вывода. Для построения экспертных систем широко применяют языки логического программирования ЛИСП или ПРОЛОГ.

В кремниевых компиляторах обычно используют как идеи трансляции описаний разных аспектов и иерархических уровней, так и экспертные знания.

Каждая современная экспертная система является глубоко специализированной, ориентированной на довольно узкую предметную область, но в будущем следует ожидать построения более универсальных систем, способных настраиваться на заданную предметную область. Этому должно способствовать развитие и выделение в экспертной системе соответствующего инвариантного ядра.

Поскольку структурный синтез формализуется на основе приближенных критериев и экспертных знаний, необходима верификация получаемых проектных решений. Поэтому очередной актуальной задачей развития математического обеспечения САПР является определение способов



и моделей для согласованного решения задач синтеза и верификации в условиях применения экспертных систем.

**Программно-методические комплексы САПР.** Программно-методические комплексы (ПМК) САПР относятся к числу наиболее сложных программных систем, создаваемых в промышленности. Типичный ПМК САПР насчитывает десятки — сотни тысяч операторов алгоритмического языка, трудоемкость его создания оценивается десятками — сотнями человеко-лет. В связи с этим одна из наиболее острых проблем создания САПР — это проблема ускорения и упрощения разработки ПМК. Решение этой проблемы ищется на путях автоматизации проектирования программного обеспечения.

Существующие методы проектирования программного обеспечения делятся на группы методов восходящего проектирования, нисходящего проектирования и расширения ядра.

В методах восходящего проектирования предварительно разрабатывается совокупность типовых модулей, из которых в дальнейшем составляются конкретные программные системы.

В методах нисходящего проектирования процесс разработки ведется последовательно на уровнях программного комплекса, программ, отдельных программных модулей. При этом решаются задачи разработки требований к программному комплексу, определяется его структура, разрабатываются спецификации, выбираются языки программирования и создаются при необходимости входные языки. Далее выбирается математическое обеспечение, разрабатываются алгоритмы, конкретизируются связи программ по информации. На уровне программных модулей осуществляется их кодирование на выбранном языке программирования. На каждом уровне после синтеза структуры должна выполняться верификация принятых решений с помощью тестирования.

В методах расширения ядра вначале создается исходная версия программной системы, которая в дальнейшем дополняется новыми программными модулями с целью расширения возможностей комплекса. Методы расширения ядра позволяют постепенно расширять возможности первоначально сданной в эксплуатацию версии комплекса.

В рамках каждой из групп методов проектирования программного обеспечения рекомендуется использовать идеи и принципы модульного программирования, в соот-

ветствии с которыми осуществляется структурирование сложных программных комплексов, регламентируются связи модулей по управлению. Рекомендуется разделять программный комплекс на модули по функциональному признаку, придерживаться принципов вертикального управления, при котором каждый модуль имеет один вход и один выход и возвращение из модуля допускается только в вызвавший его модуль. Модульное программирование вошло составной частью в различные технологии программирования, например в структурное программирование. Для структурного программирования характерно использование нисходящего программирования и нисходящего тестирования программ, отказ от использования оператора *goto*, построение программ из ограниченного числа типов управляющих структур. В каждой технологии программирования устанавливаются свои правила представления и документирования проектных решений, принимаемых на различных иерархических уровнях разработки. На верхних уровнях распространено применение метаязыков, называемых также псевдоязыками или языками спецификаций, в этих языках наряду со средствами алгоритмических языков общего назначения допускается использование различных макрооператоров, часто выражаемых средствами естественного языка. Другими способами представления проектных решений на уровне проектирования программных комплексов являются граф-схемы, функциональные диаграммы, диаграммы «вход — обработка — выход» и другие графические средства укрупненного выражения структуры программного обеспечения. Получаемое описание отражает основные связи между модулями по управлению и информации и служит исходным документом для дальнейшей конкретизации программного обеспечения на последующих иерархических уровнях.

Существующие инструментальные средства автоматизации разработки программного обеспечения позволяют проводить структурирование программ с разделением их на модули, производить оценку показателей связности и сцепления модулей, документировать результаты разработки, производить трансляцию отдельных фрагментов программ на терминальный язык программирования, моделировать работу программного комплекса по его функциональным и (или) эксплуатационным спецификациям. По результатам моделирования можно на ранних этапах проектирования приблизительно оценить запросы систем-

ных ресурсов, выявить наличие несогласованности в принимаемых проектных решениях, отладить систему синхронизации различных ветвей, на которые разбивается вычислительный процесс при использовании многопроцессорных вычислительных систем. В состав инструментальных подсистем САПР зачастую входят кросс-средства, предназначенные для разработки программного обеспечения проектируемой ЭВМ, называемой целевой, на имеющейся в составе САПР технологической (инструментальной) ЭВМ. Кросс-средства включают в себя трансляторы с языка целевой машины на язык инструментальной ЭВМ и программы моделирования и отладки разрабатываемого программного обеспечения на инструментальной ЭВМ.

Дальнейшее развитие инструментальных средств разработки программного обеспечения должно иметь целью автоматизацию синтеза программ различных классов, в том числе диалоговых мониторов, различных языковых процессоров, генерацию экономичных версий ПМК моделирования и верификации проектных решений из имеющихся инвариантных средств и т. п.

Важное значение имеет унификация внутреннего представления данных для САПР конкретной отрасли и на этой основе интеграция усилий многих коллективов по созданию взаимосогласованных компонентов программного обеспечения, способных объединяться в составе тех или иных версий САПР.

Большинство существующих САПР создавалось в условиях применения однопроцессорных ЭВМ, действующих автономно или в составе технических средств САПР радиальной структуры. Переход на применение многопроцессорных вычислительных систем и персональных ЭВМ, объединенных линиями передачи данных в локальную вычислительную сеть, потребует развития протоколов связи, разработки новых операционных систем, создания распределенных банков данных. Вновь создаваемые пакеты прикладных программ целесообразно делать мобильными. Предпочтительными языками программирования становятся языки СИ, ФОРТРАН-77, МОДУЛА-2.

Особое место в САПР придается средствам машинной графики. В настоящее время создано несколько систем, являющихся ядрами графических пакетов программ. Примером такой системы является ГКС, в рамках которой установлен перечень элементарных операций интерактивной графики и вывода графической информации из ЭВМ

вместе с соответствующим лингвистическим и программным обеспечением. Опираясь на средства такого ядра, можно создавать более сложные программы обработки графической информации для конкретных приложений. Основные проблемы машинной графики связаны с созданием цветных графических дисплеев с высокой разрешающей способностью и построением высокопроизводительных процессоров обработки трехмерных изображений, например, для построения проекций, сечений, объединения изображений деталей в изображение узла и т. п. операций, выполняемых в реальном масштабе времени.

**Комплексная автоматизация проектирования и производства изделий техники.** Комплексная автоматизация охватывает проектирование и производство изделий и обеспечивается совокупностью автоматизированных систем. В эту совокупность входят автоматизированная система научных исследований (АСНИ), система автоматизированного проектирования (САПР), автоматизированная система технологической подготовки производства (АСТПП), автоматизированная система управления производством (АСУП) и гибкая производственная система (ГПС). В этом ряду АСНИ служит для выполнения научно-исследовательских работ и часто рассматривается как подсистема САПР. Функциями АСТПП являются разработка технологических процессов, проектирование оснастки, инструмента, специализированного технологического оборудования. АСТПП также может рассматриваться как подсистема САПР. АСУП используется для планирования производства, распределения ресурсов, решения задач материально-технического снабжения. ГПС представляет собой совокупность технологического оборудования и средств обеспечения его функционирования в автоматическом режиме, причем в ГПС должна быть обеспечена возможность автоматизированной переналадки при производстве любых изделий в пределах установленного класса и установленного диапазона их характеристик.

Применение ГПС целесообразно только при мелкосерийном производстве изделий. В остальных случаях предпочтение следует отдать более производительным и менее дорогим способам. При единичном производстве используют универсальное оборудование, при крупносерийном и массовом производстве — автоматические линии, например роторно-конвейерные линии в машиностроении.

Основой ГПС являются гибкие производственные моду-

ли (ГПМ). ГПМ — это единица технологического оборудования, оснащенная устройствами автоматической смены инструмента и заготовок, удаления отходов, контроля и подналадки технологического процесса и управляемая с помощью сменяемых программ, настраивающих оборудование на изготовление нужного изделия. Для организации автоматизированного производства ГПМ объединяются в единую систему с помощью автоматизации транспортных операций и централизации управления от ЭВМ. Вычислительная машина выполняет распределение технологических операций между модулями, осуществляет синхронизацию их работы, управляет материальными потоками в системе. В таких системах проектирование и управление технологическими процессами также требуют интеграции, что и осуществляется созданием соответствующего программного и информационного обеспечения для ЭВМ, служащих техническим обеспечением АСТПП и АСУ ТП в составе гибкого автоматизированного производства (ГАП).

Применение ГАП имеет важное социально-экономическое значение для развития страны.

Преимущества гибких производств заключаются в быстром переходе на изготовление новой продукции, в резком сокращении численности работающих и, следовательно, в приближении к идеалу будущих «безлюдных» производств, в повышении коэффициента сменности работы оборудования.

В настоящее время лишь закладываются основы интегрированных автоматизированных производственных систем. САПР в составе ГАП будут развиваться в направлении совершенствования средств машинной графики, методов и программ автоматического синтеза технологических процессов и конструкций. Но роль САПР в автоматизации производства не ограничивается функциями автоматизации конструирования и технологической подготовки производства в уже созданных ГАП. Не менее важная задача САПР — проектирование самих автоматизированных производств, включая проектирование робототехнических комплексов, технологического оборудования, их компоновку, размещение и т. п. Для этого в САПР должны быть мощные средства имитационного моделирования работы производственных линий, участков, цехов; синтеза и анализа объектов с физически разнородными элементами, каковыми являются различные виды роботов, манипуляторов, тех-

нологических аппаратов; инструментальные средства проектирования программного обеспечения ГАП; средства разработки вычислительных сетей в составе ГАП и др.

### **Контрольные вопросы и задания**

1. Сформулируйте основные показатели качества САПР.
2. Каким образом связаны показатели качества технического и программного обеспечения САПР?
3. Почему методы имитационного моделирования находят широкое применение в САПР?
4. Как можно оценить качество имитационных моделей?
5. Какие системные требования предъявляются к программному обеспечению САПР?
6. Почему нельзя существенно ускорить процесс разработки программного обеспечения путем привлечения к разработкам дополнительного числа программистов?
7. Объясните роль и место САПР в гибких автоматизированных производственных системах.
8. Объясните специфику применения различных режимов работы КТС САПР.
9. Сформулируйте пути повышения надежности аппаратуры КТС САПР.
10. Каким образом оценивается надежность программного обеспечения САПР?
11. Объясните назначение и сущность основных процедур имитационного моделирования.

## СПИСОК ЛИТЕРАТУРЫ

1. Атре Ш. Структурный подход к организации баз данных. М.: Финансы и статистика, 1983.
2. Бойченко Е. В., Кальфа В., Овчинников В. В. Локальные вычислительные сети. М.: Радио и связь, 1985.
3. Брахман Т. Р. Многокритериальность и выбор альтернативы в технике. М.: Радио и связь, 1984.
4. Гемингерн В. И., Каган Б. М. Методы оптимального проектирования. М.: Энергия, 1980.
5. Комплекс общепромышленных руководящих методических материалов по созданию АСУ и САПР. М.: Статистика, 1980.
6. Коршунов Ю. М. Математические основы кибернетики. М.: Энергия, 1980.
7. Лорин Г., Дейтель Х. М. Операционные системы. М.: Финансы и статистика, 1984.
8. Михалевич В. С., Волкович В. Л. Вычислительные методы исследования и проектирования сложных систем. М.: Наука, 1982.
9. Морозов К. К., Одинокое В. Г., Курейчик В. М. Автоматизированное проектирование конструкций радиоэлектронной аппаратуры. М.: Радио и связь, 1983.
10. Норенков И. П. Введение в автоматизированное проектирование технических устройств и систем. М.: Высшая школа, 1986.
11. Норенков И. П., Маничев В. Б. Системы автоматизированного проектирования электронной и вычислительной аппаратуры. М.: Высшая школа, 1983.
12. Петренко А. И. Основы автоматизации проектирования. Киев: Техника, 1982.
13. Петренко А. И., Тетельбаум А. Я. Формальное конструирование электронно-вычислительной аппаратуры. М.: Советское радио, 1979.
14. Ткори Т., Фрай Дж. Проектирование структур баз данных. М.: Мир, 1985.
15. Хог Э., Арора Я. Прикладное оптимальное проектирование. М.: Мир, 1983.
16. Система автоматизированного проектирования в радиоэлектронике. Справочник/Е. В. Авдеев, А. Т. Еремин, И. П. Норенков, М. И. Песков; Под ред. И. П. Норенкова. М.: Радио и связь, 1986.
17. Батищев Д. И. Методы оптимального проектирования. М.: Радио и связь, 1984.
18. Заморин А. П., Мячев А. А., Селиванов Ю. П. Вычислительные машины, системы, комплексы: Справочник/Под ред. Б. Н. Наумова, В. В. Пржиялковского. М.: Энергоатомиздат, 1985.

## АЛФАВИТНЫЙ УКАЗАТЕЛЬ

- Абстрагирование 8  
Автоматизация проектирования 4  
Автоматизированное проектирование 6  
— рабочее место 39, 64, 375  
Адаптивное моделирование 249  
Алгебраизация 175  
Алгоритм волновой 327  
— вписывания 294  
— Гомори 312  
— последовательный 322  
— Рота 259  
— циклический 247  
Анализ 12  
— многовариантный 255  
— регрессионный 153  
— статистический 256  
— чувствительности 255  
Аппаратура передачи данных 85  
АРМ 39, 64, 375  
Аспект проектирования 8  
Атрибут 93  
  
База данных 95  
— знаний 140  
Банк данных 40  
Блочный-иерархический подход 8  
  
Вектор нагрузок 165  
Вентильный подуровень 189  
Верификация 359  
Вершина графа 198  
Восстановление базы данных 126  
  
Гибкое автоматизированное производство 377  
  
Гиперграф 214  
Граф 198  
— Гамильтонов 204  
— гомеоморфный 212  
— изоморфный 211  
— Кенига 215  
— планарный 211  
— плоский 211  
— связный 204  
— Эйлеров 204  
  
Двухполюсник 168  
Декомпозиция 225  
Дерево графа 205  
— — фундаментальное 179  
Диакоптика 225, 243  
Диалоговое средство 58  
Диалоговый режим 58  
Дискретизация задачи 155, 160  
Дисплей 74  
Документ проектный 7  
Дополнение графа 203  
Дуга графа 199  
  
Задание техническое 48, 50  
Задача выбора 12  
— коммивояжера 207  
— принятия решения 12  
Запись данных 93  
Защита данных 129  
  
Интегрирование раздельное 246  
Интегрированная САПР 42  
Интерпретация 374  
Интерфейс 97



- Канал связи 85  
 Ключ 106  
 Комбинирование моделей и методов 247  
 Коммутация пакетов 81  
 Компиляция 374  
 Комплекс средств автоматизации проектирования 36  
 — технических средств 330  
 — центральный вычислительный 37  
 Компоновка 269, 323  
 Конечный элемент 163  
 Контур 213  
 Конфигурация технических средств 63  
 Координата базисная 175  
 Коэффициент весовой 19, 28  
 — влияния 255  
 Критерий аддитивный 18  
 — минимаксный 16, 22  
 — мультипликативный 16, 20  
 — обобщенный 16  
 — оптимальности 14  
 — статистический 16  
 — частный 16  
 Латентность фрагмента 248  
 Линеаризация 176  
 Макромодель 145  
 Макроуровень 146, 166  
 Максимум 22  
 Маршрут в графе 203  
 — проектирования 356  
 Матрица Гессе 279  
 — жесткости 165  
 — инцидентности 201  
 — контуров и сечений 179  
 — разреженная 230  
 — расстояний 206  
 — смежности 200  
 — цепей 217  
 — Якоби 178  
 Машина логического моделирования 254  
 Машинная графика 70  
 Метауровень 146  
 Метод ветвей и границ 313  
 — Гаусса 229, 243  
 — Гаусса—Зейделя 284  
 — Гира 237  
 — градиентный 285  
 — доступа 115  
 — инвертированный 117  
 — Зейделя 227, 251  
 — золотого сечения 290  
 — конечных разностей 160  
 — элементов 162  
 — Монте-Карло 256  
 — наискорейшего спуска 286  
 — неявный 236  
 — Ньютона 228, 244, 287  
 — однонаправленных моделей 245  
 — ортогональных направлений 284  
 — отсекаемых плоскостей 310  
 — параллельных касательных 284  
 — ПВР 227  
 — переменной метрики 287  
 — переменных направлений 248  
 — состояния 180  
 — подсхем 243  
 — покоординатного спуска 284  
 — полиномиальной аппроксимации 290  
 — приписывания баллов 28  
 — приращений 256  
 — прогнозируемых реакций 247  
 — прогонки 231  
 — продолжения решения по параметру 228  
 — простой итерации 227, 251  
 — разреженных матриц 230  
 — ранжирования 28  
 — релаксационный 227  
 — Розенброка 284  
 — РФС 246  
 — случайного поиска 290  
 — сопряженных градиентов 287  
 — статистических испытаний 256

- Метод статистического градиента 296
- табличный 179
  - трапеций 238
  - узловых потенциалов 176
  - установления 229
  - ФДН 237, 241
  - чисел Фибоначчи 289
  - штрафных функций 290
  - явный 236
- Методика макро моделирования 151
- Микроуровень 146, 154
- Минимакс 24
- Многополюсник 168
- Мобильность программного обеспечения 346
- Моделирование адаптивное 249
- асинхронное 194, 250
  - имитационное 147, 348
  - логико-электрическое 254
  - многоуровневое 226
  - параллельное 253
  - потактовое 250
  - синхронное 192, 251
  - событийное 226
- Модель алгоритмическая 147
- аналитическая 147
  - асинхронная 189
  - базы данных концептуальная 101
  - данных 101, 105
  - иерархическая 107
  - реляционная 105
  - сетевая 108
  - имитационная 147
  - математическая 61, 142
  - полная 144
  - синхронная 189
- Модуль программного обеспечения 368
- Монитор 370, 374
- Моноканал 79
- Мультиграф 201
- Набор данных 93
- Надежность метода 224
- Накопитель 122
- Обеспечение САПР информационное 40, 92
- лингвистическое 41
  - математическое 36
  - методическое 41
  - организационное 41
  - программное 39
  - техническое 36, 63, 330
- Область адекватности 148
- работоспособности 293
- Операционная система 39, 365
- Операция проектная 7
- Оптимизация 12, 277
- векторная 16
- Орграф 200
- Оценка экспертная 28
- Пакет прикладных программ 39 370
- Параметр 61
- Переменная проектирования 263
- состояния 181
- Планирование экспериментов 152
- Планировщик заданий 367
- Подграф 202
- Подсистема САПР 47
- Предложение техническое 48
- Предпроектные исследования 48
- Принцип гарантированного результата 23
- Лагранжа 158
- Программа обрабатывающая 368
- управляющая 366
- Программирование математическое 17, 264
- Программно-методический комплекс САПР 356
- Проект 7
- рабочий 49
  - технический 49
  - эскизный 48

- Проектирование 6  
 — алгоритмическое 10  
 — внешнее 49  
 — внутреннее 49  
 — восходящее 9  
 — конструкторское 10  
 — нисходящее 9  
 — технологическое 11  
 — функциональное 10  
 Проектное решение 7  
 Протокол связи 79  
 Процедура проектная 7  
 Пульт экранный 75  
 Путь в графе 213
- Размещение 271, 325  
 Разреженность матрицы 225  
 Ранжирование 252  
 Раскраска графа 210  
 Расплывчатое множество 196  
 Расстояние 206  
 Ребро графа 198  
 Регистровый подуровень 195  
 Редактор связей 369  
 Режим интерактивный 35  
 — мультипрограммный 69  
 — однопрограммный 69  
 — пакетный 58, 70  
 — разделения времени 70  
 Риск сбоя 192
- САПР 35  
 Связь гираторная 170  
 — трансформаторная 170  
 Сетевая имитационная модель 359  
 Сеть вычислительная 78  
 Симплекс-метод 308  
 Синтез 261  
 — параметрический 273  
 — структурный 267, 304  
 — тестов 259  
 Система автоматизированного проектирования 35
- автоматического проектирования 35  
 — массового обслуживания 359  
 — мониторинга 57  
 — управления базами данных 40, 96  
 Системный подход 59  
 Словарь данных 101  
 Служба САПР 54  
 Событийность 225, 253  
 Спецпроцессор 254, 364  
 Способ несущей 188  
 — огибающей 188  
 Среда передающая 79  
 Стация проектирования 7  
 Станция терминальная 39  
 Степень вершины 202  
 Структура модели 152  
 — объекта 261  
 СУБД 40, 96  
 Суграф 203
- Тест 258  
 Точка стационарная 278  
 Транслятор 40, 368  
 Трассировка 271, 326
- Уравнение диффузии 157  
 — компонентное 167  
 — конечное 222  
 — Ламе 157  
 — Навье—Стокса 159  
 — непрерывности 155  
 — неразрывности 159  
 — теплопроводности 221  
 — топологическое 167  
 — Фурье 173  
 — Эйлера 159  
 Уровень иерархический 8  
 — — компонентный 10  
 — — логический 10, 189  
 — проектирования 8  
 — — системный 10  
 — — схмотехнический 10

Условие работоспособности 17,  
292  
— сходимости 227  
Устойчивость 238  
Устройство ввода информации 71  
— вывода информации 72  
Фазовая переменная 166  
Файл 93  
— индексный 115  
Фрагментация 249  
Функция выходов 195  
— координатная 164  
— переходов 195  
— пробная 164  
— формы 164  
— целевая 17  
Хеширование 118  
Хорда 179  
Целостность данных 95  
Центрирование 296  
Цепь графа 203

Цикл 203  
— Гамильтонов 204  
Чертежный автомат 72  
Число внутренней устойчивости  
210  
— графа хроматическое 340  
— — цикломатическое 209  
— обусловленности 232  
Шаблон 160  
Экстремум безусловный 277  
— локальный 278  
— условный 277  
Элемент данных 93  
Эмулятор 40  
Этап проектирования 7  
Язык манипулирования данными  
99  
— моделирования 352  
— описания данных 99  
— управления заданиями 367  
Языковый процессор 374

## О Г Л А В Л Е Н И Е

Предисловие . . . . .	3
<b>Глава 1. Методология автоматизированного проектирования . . . . .</b>	<b>6</b>
1.1. Общие сведения о проектировании . . . . .	6
1.2. Задачи принятия решений в САПР . . . . .	12
1.3. Выбор критериев оптимальности . . . . .	14
1.4. Методы задания предпочтения на множестве частных критериев . . . . .	27
1.5. Виды обеспечения и классификация САПР . . . . .	35
1.6. Системный подход к автоматизации проектирования и принципы организации САПР . . . . .	46
Контрольные вопросы и задания . . . . .	61
<b>Глава 2. Типовые структуры САПР . . . . .</b>	<b>63</b>
2.1. Организация технических средств САПР . . . . .	63
2.2. Режимы работы технических средств САПР . . . . .	69
2.3. Технические средства машинной графики . . . . .	70
2.4. Организация взаимодействия конструктора с ЭВМ . . . . .	76
2.5. Вычислительные сети САПР . . . . .	78
2.6. Аппаратура передачи данных . . . . .	85
2.7. Примеры структур САПР ЭВА и РЭА . . . . .	87
Контрольные вопросы и задания . . . . .	92
<b>Глава 3. Информационное обеспечение САПР . . . . .</b>	<b>92</b>
3.1. Основные понятия и определения . . . . .	92
3.2. Проектирование базы данных. Языки базы данных . . . . .	98
3.3. Модели данных . . . . .	105
3.4. Реализация базы данных . . . . .	113
3.5. Построение физической модели базы данных . . . . .	121
3.6. Краткая техническая характеристика СУБД РАПИРА—5.3—82 . . . . .	129
Контрольные вопросы и задания . . . . .	141
<b>Глава 4. Математические модели проектируемых объектов . . . . .</b>	<b>142</b>
4.1. Классификация математических моделей . . . . .	142
4.2. Требования к математическим моделям . . . . .	148
4.3. Методы получения моделей элементов . . . . .	151
4.4. Математические модели объектов проектирования, используемые на микроуровне . . . . .	154
4.5. Математические модели объектов проектирования, используемые на макроуровне . . . . .	166
4.6. Получение математических моделей систем на макроуровне . . . . .	175
4.7. Математические модели на метауровне . . . . .	185

4.8. Математические модели для задач конструирования РЭА	196
Контрольные вопросы и задания	220
<b>Глава 5. Анализ технических объектов в САПР</b>	<b>222</b>
5.1. Требования к методам анализа	222
5.2. Методы анализа статических режимов	226
5.3. Методы анализа переходных процессов	235
5.4. Методы анализа повышенной эффективности	243
5.5. Методы анализа логических и функциональных схем электронной вычислительной аппаратуры	250
5.6. Методы многовариантного анализа	255
Контрольные вопросы и задания	260
<b>Глава 6. Синтез технических объектов в САПР</b>	<b>261</b>
6.1. Постановка задачи синтеза	261
6.2. Структурный синтез и параметрическая оптимизация	267
6.3. Разновидности задач оптимизации	277
6.4. Показатели эффективности и выбор методов поиска экстремума	281
6.5. Задача оптимизации допусков и технических требований	292
6.6. Методы оптимизации технологических процессов	298
6.7. Особенности решения задач структурного синтеза	304
6.8. Модели математического программирования в структур- ном синтезе технических объектов	308
6.9. Последовательные методы в задачах проектирования сложных объектов	319
Контрольные вопросы и задания	328
<b>Глава 7. Вопросы разработки САПР</b>	<b>330</b>
7.1. Разработка технического обеспечения САПР	330
7.2. Определение характеристик и оценка качества создавае- мой САПР	339
7.3. Имитационное моделирование в автоматизированном проектировании	348
7.4. Проектирование САПР	356
7.5. Выбор структуры программного обеспечения САПР	365
7.6. Роль и место САПР в гибких автоматизированных про- изводственных системах	376
7.7. Социальное значение и перспективы развития САПР	381
Контрольные вопросы и задания	391
Список литературы	392
Алфавитный указатель	393

## **УЧЕБНИК**

**ВЯЧЕСЛАВ ПЕТРОВИЧ КОРЯЧКО  
ВИКТОР МИХАЙЛОВИЧ КУРЕИЧИК  
ИГОРЬ ПЕТРОВИЧ НОРЕНКОВ**

### **Теоретические основы САПР**

Редактор В. Г. Одинокоев  
Редактор издательства В. И. Петухова  
Художественные редакторы Т. А. Дворецкова, А. А. Белоус  
Технический редактор В. В. Хапаева  
Корректор М. Г. Гулина  
ИБ № 1387

Сдано в набор 30.01.87. Подписано в печать 14.07.87. Т-16432. Формат 84×108<sup>1</sup>/<sub>32</sub>. Бумага типографская № 2. Гарнитура литературная. Печать высокая. Усл. печ. л. 21,0. Усл. кр.-отт. 21,0. Уч.-изд. л. 22,82. Тираж 45 000 экз. Заказ 785. Цена 1 р. 20 к.

Энергоатомиздат, 113114, Москва, М-114, Шлюзовая наб., 10

Владимирская типография Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли, 600000, г. Владимир, Октябрьский проспект, д. 7