

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

В. Н. ЛАНЦОВ

ОСНОВЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

Учебное пособие



Владимир 2023

УДК 004.942
ББК 30.2
Л22

Рецензенты:

Доктор технических наук, профессор
зам. директора Института радиоэлектроники и информационных технологий
Нижегородского государственного технического университета
им. Р. Е. Алексеева
В. П. Хранилов

Доктор технических наук, профессор
зав. кафедрой информатики и защиты информации
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых
М. Ю. Монахов

Издается по решению редакционно-издательского совета ВлГУ

Ланцов, В. Н.

Л22 Основы автоматизации проектирования : учеб. пособие /
В. Н. Ланцов ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. –
Владимир : Изд-во ВлГУ, 2023. – 183 с. – ISBN 978-5-9984-1541-8.

Содержит описание всех уровней автоматизации проектирования электронно-вычислительной аппаратуры. Изложены особенности и методы системного, функционально-логического, схемотехнического, компонентного и конструкторского уровней, а также все основные виды обеспечения систем автоматизированного проектирования (САПР): математическое, программное и лингвистическое.

Предназначено для студентов, обучающихся по направлению подготовки 09.03.01 – Информатика и вычислительная техника (бакалавриат), а также по всем техническим направлениям и специальностям, связанным с автоматизацией проектирования. Может быть использовано при подготовке в аспирантуре по соответствующим направлениям.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Ил. 83. Табл. 9. Библиогр.: 66 назв.

УДК 004.942
ББК 30.2

ISBN 978-5-9984-1541-8

© ВлГУ, 2023

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ.....	5
ВВЕДЕНИЕ	6
Глава 1. ОБЩИЕ СВЕДЕНИЯ О САПР.....	7
1.1. Введение	7
1.2. Этапы и уровни проектирования ЭВА	8
1.3. Общие сведения о проектировании	10
1.4. Структура САПР	13
1.5. Схема процесса проектирования в САПР	16
<i>Вопросы для самопроверки</i>	19
Глава 2. ВИДЫ ОБЕСПЕЧЕНИЯ САПР.....	20
2.1. Математическое обеспечение САПР	20
2.2. Методы многовариантного анализа.....	23
2.3. Постановка задач оптимизации в САПР	26
2.4. Методы оптимизации	29
2.5. Лингвистическое и программное обеспечение	47
<i>Вопросы для самопроверки</i>	52
Глава 3. ЭТАПЫ СИСТЕМНОГО И ФУНКЦИОНАЛЬНОГО ПРОЕКТИРОВАНИЯ В САПР ЭВА.....	53
3.1. Этап системного проектирования.....	53
3.2. Задачи и особенности функционального проектирования.....	54
3.3. Моделирование цифровых функционально-логических схем	56
3.4. Моделирование аналоговых функциональных схем	71
<i>Вопросы для самопроверки</i>	81
Глава 4. АВТОМАТИЗАЦИЯ СХЕМОТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ.....	82
4.1. Задачи схемотехнического проектирования.....	82
4.2. Математические модели элементов.....	82
4.3. Математическая модель схемы	92
4.4. Анализ линейного режима	94
4.5. Анализ статического режима	101

4.6. Анализ переходного режима	104
<i>Вопросы для самопроверки</i>	110
Глава 5. ДОПОЛНИТЕЛЬНЫЕ ВИДЫ АНАЛИЗА СХЕМОТЕХНИЧЕСКОГО МОДЕЛИРОВАНИЯ	111
5.1. Анализ чувствительности	111
5.2. Расчет собственных шумов схемы.....	117
5.3. Расчет нелинейных искажений в режиме малого сигнала	120
5.4. Расчет спектра в установившемся режиме	125
5.5. Моделирование установившегося режима	126
5.6. Метод гармонического баланса.....	129
<i>Вопросы для самопроверки</i>	132
Глава 6. МНОГОУРОВНЕВОЕ МОДЕЛИРОВАНИЕ	133
6.1. Макромоделирование	133
6.2. Сложности многоуровневого моделирования	138
6.3. Организация компиляторов для многоуровневого моделирования.....	144
6.4. Математическая модель схемы в расширенном координатном базисе и связь с механическими системами	147
6.5. Модели и методы решения дифференциальных уравнений в частных производных	151
<i>Вопросы для самопроверки</i>	157
Глава 7. АВТОМАТИЗАЦИЯ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ	158
7.1. Введение	158
7.2. Задача компоновки	161
7.3. Задача размещения	163
7.4. Задачи трассировки.....	165
<i>Вопросы для самопроверки</i>	176
ЗАКЛЮЧЕНИЕ	177
СПИСОК БИБЛИОГРАФИЧЕСКИХ ССЫЛОК	178
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ	182

*Памяти Игоря Петровича
Норенкова посвящается*

ПРЕДИСЛОВИЕ

Кафедра вычислительной техники и систем управления – одна из ведущих кафедр в Российской Федерации в области систем автоматизированного проектирования (САПР) в электронике. До 1993 года на кафедре в течение 20 лет разрабатывалась система схемотехнического моделирования, которая использовалась во многих вузах и научных организациях СССР. С 1994 года тогда еще ВлГТУ включился в европейскую программу EURIPRACTICE, которая обеспечивала ведущие вузы самыми передовыми САПР для электроники, проводились семинары, поставлялись библиотеки моделей ведущих мировых лидеров в микроэлектронике. За эти годы на кафедре получены лицензии практически всех ведущих САПР, преподаватели кафедры прошли обучение на европейских курсах, внедрили указанные системы в учебный процесс и научные исследования.

Пособие написано в память об одном из основателей направления САПР в СССР, а затем и в России, докторе технических наук, профессоре, лауреате Государственной премии Игоре Петровиче Норенкове. С 1983 года он принимал активное участие в становлении научно-преподавательской группы во Владимирском политехническом институте по направлению САПР в электронике, много раз приезжал во Владимир, оказывал методическую помощь кафедре. Именно на материалах лекций И. П. Норенкова по основам САПР в МГТУ им. Н. Э. Баумана, а также его учебных пособий и учебников [1 – 6] был построен курс «Основы автоматизации проектирования», читаемый автором с 1991 года.

Еще один из основателей САПР в России, оказавший огромное влияние на автора в области автоматизации схемотехнического проектирования, – доктор технических наук, профессор Валерий Николаевич Ильин [7 – 10], научный руководитель автора в докторантуре МАИ (1988 – 1991).

Автор искренне благодарен рецензентам: доктору технических наук, профессору Хранилову Валерию Павловичу, а также доктору технических наук, профессору Монахову Михаилу Юрьевичу, критические советы и замечания которых способствовали повышению уровня качества изложенного в пособии материала.

ВВЕДЕНИЕ

Электронная и вычислительная аппаратура (ЭВА) относится к наиболее сложным техническим объектам. Поэтому автоматизация проектирования здесь особенно актуальна. Именно при решении задач проектирования ЭВА зародились основы САПР.

В настоящее время нет необходимости убеждать специалистов в целесообразности использования средств моделирования и САПР, особенно при разработке сложных электронных устройств ЭВА, где без применения САПР не обойтись. Повышение степени интеграции современных интегральных микросхем, сокращение сроков проектирования еще более увеличивают потребность в САПР, которые позволяют значительно снизить трудоёмкость и затраты на проектирование, повысить качественные показатели проектируемых изделий.

Средства САПР совершенствуются, поэтому постоянно ощущается необходимость в профессионалах, способных применять сложные и дорогие средства САПР для ускорения процесса проектирования, а также разработки принципиально новых решений. Для этого следует четко представлять основные методы и приёмы автоматизированного проектирования, возможности и ограничения программных средств для моделирования устройств и систем в различных режимах их работы, оценивать и правильно интерпретировать результаты проектирования.

Среди всех случаев применения САПР особое место занимает использование их для проектирования таких сложных изделий, как большие и сверхбольшие интегральные схемы (БИС и СБИС). Применение математических методов моделирования и расчета на всех этапах разработки БИС дает значительный экономический эффект. Поэтому знание основ построения САПР – важный аспект их успешного применения и совершенствования.

Глава 1. ОБЩИЕ СВЕДЕНИЯ О САПР

1.1. Введение

Автоматизация проектирования характеризуется систематическим применением ЭВМ для выполнения проектных операций и процедур при рациональном использовании творческих способностей человека и вычислительных возможностей ЭВМ [2]. В определении выделяются слова «систематическим», что отражает применение при постоянном (повседневном) проектировании; «проектных», что характеризует выполнение именно проектных операций и процедур; «творческих», что подтверждает – человек со своими творческими способностями является частью САПР; «вычислительных возможностей ЭВМ», ЭВМ – только как хороший и быстрый вычислитель.

Система, реализующая автоматизированное проектирование, представляет собой САПР (в англоязычном написании CAD System – Computer Aided Design System).

В соответствии с ГОСТом *система автоматизированного проектирования* – это организационно-техническая система, состоящая из комплекса средств автоматизации проектирования, взаимодействующего с подразделениями проектной организации, и выполняющая автоматизированное проектирование [3].

Аспекты применения САПР [2]:

1) автоматизация проектирования трудоемких рутинных работ, не требующих творческого подхода (например, оформление и выпуск конструкторской документации на изделие, размещение и трассировка печатных плат);

2) вместо дорогостоящей и длительной экспериментальной отработки объекта проектирования на макетах – применение численных экспериментов на ЭВМ, т. е. математическое моделирование;

3) решение задач, не поддающихся полной формализации (некоторые задачи синтеза, выбор структуры и принципов организации проектируемых систем). ЭВМ и САПР здесь играют вспомогательную роль (архив предыдущих решений, работа в интерактивном режиме, использование методов искусственного интеллекта и т. п.).

1.2. Этапы и уровни проектирования ЭВА

ЭВА и другие радиоэлектронные системы относятся к самым сложным объектам, проектирование которых – трудоемкий процесс. Поэтому выполнение проектных работ распределяется как по времени, так и между проектными подразделениями (службами, исполнителями) [2].

Распределение работ по времени приводит к разделению процесса проектирования на этапы и стадии [Там же]:

1. Научно-исследовательская работа (НИР) включает предпроектное исследование, разработку технического задания и технического предложения. Здесь выполняется исследование по новым принципам, новым структурам, новым физическим процессам, новой элементной базе и т. п. Результат выполнения этого этапа – техническое предложение в виде отчета по НИР.

2. Опытно-конструкторская работа (ОКР) объединяет две стадии проектирования – эскизное и техническое. Результат выполнения этого этапа – детальная проработка всех возможных построений изделия с изготовлением опытного образца или макета.

3. Рабочее проектирование включает следующие стадии: рабочее проектирование, изготовление, отладку, испытание и ввод в эксплуатацию. Здесь тщательно прорабатываются все схемные, конструкторские и технические решения, а также выполняется изготовление и испытание изделия.

Основное назначение САПР – решение задач второго и третьего этапов [Там же].

В зависимости от порядка, в котором выполняются этапы проектирования, различают восходящее и нисходящее проектирование. *Восходящее проектирование* (проектирование снизу вверх) характеризуется решением задач более низких иерархических уровней перед решением задач более высоких уровней. Противоположная последовательность приводит к *нисходящему проектированию* (проектированию сверху вниз) [Там же].

Распределение работ между подразделениями и исполнителями выполняется на основе блочно-иерархического подхода, что приводит в процессе проектирования к выделению ряда уровней, когда общую сложную задачу делят на ряд более простых.

Уровни проектирования можно разделить по степени подробности (горизонтальные уровни) и по характеру учитываемых свойств (вертикальные уровни).

В табл. 1 приведены уровни проектирования для ЭВА [2].

Таблица 1

Вертикальные и горизонтальные уровни проектирования ЭВА

Функциональный	Алгоритмический	Конструкторский	Технологический
Системный (структурный)	Программирование системы	Стойки, шкафы	Технологический процесс
Функционально-логический	Программирование модулей	Блоки, панели	Технологический маршрут
Схемотехнический	Проектирование микропрограмм	Платы	Технологическая операция
Компонентный (топологический)	—	Кристаллы, ячейки	—

Функциональный (вертикальный) уровень проектирования связан с разработкой структурных, функциональных и принципиальных электрических схем ЭВА. Здесь определяются основные особенности структуры, принципы функционирования, важнейшие параметры и характеристики. Для выполнения задач на этом уровне обычно используется нисходящее проектирование. В состав вертикального функционального уровня входят следующие горизонтальные уровни:

1) *системный* (структурный), где проектируются структурные схемы ЭВА. На этом уровне рассматривается вся ЭВА или система в целом;

2) *функционально-логический*, где проектируются функциональные и принципиальные схемы устройств ЭВА. Здесь выделяют несколько подуровней, а также используют разные подходы к проектированию цифровых и аналоговых устройств;

3) на *схемотехническом* уровне проектируются принципиальные электрические схемы устройств, интегральные схемы или фрагменты БИС;

4) на *компонентном* (топологическом) уровне разрабатываются отдельные компоненты, в первую очередь компоненты и топология интегральных схем, состоящих из участков полупроводникового кристалла.

Алгоритмическое проектирование связано с разработкой алгоритмов функционирования, общего математического и программного обеспечения ЭВА. Верхние уровни алгоритмического проектирования служат для создания программного обеспечения ЭВА и ЭВМ. Для сложных программных систем выполняется планирование программной системы, разрабатываются схемы алгоритмов и программные модули. Здесь используется нисходящее проектирование. Низшие иерархические уровни алгоритмического проектирования тесно взаимосвязаны с соответствующими уровнями функционального проектирования.

Конструкторский уровень проектирования включает в себя вопросы конструкторской реализации результатов функционального проектирования. Здесь решаются следующие группы задач: коммутационно-монтажное проектирование (задачи размещения и трассировки); обеспечение допустимых тепловых режимов; конструирование электромеханических узлов ЭВА; изготовление конструкторской документации. Для задач конструкторского проектирования характерно восходящее проектирование.

Технологическое проектирование охватывает вопросы технологической реализации результатов конструкторского уровня, т. е. вопросы создания технологических процессов изготовления изделий.

В САПР задачи нескольких близких по математическому аппарату уровней часто объединяют в отдельные подсистемы. Кроме того, на инженерном сленге часто вместо слова «уровень» используют слово «этап». Таким образом, этапами иногда будем называть как последовательные этапы работ по времени, так и этапы проектирования ЭВА на отдельных уровнях.

1.3. Общие сведения о проектировании

Проектирование – процесс создания описания, необходимого для построения еще не существующего объекта на основе первичного описания этого объекта.

Способы создания описания:

- неавтоматизированный (весь процесс проектирования выполняется человеком; иногда называют ручным проектированием);
- автоматизированный (проектирование выполняется при взаимодействии человека и ЭВМ);
- автоматический (все преобразования описаний выполняются полностью ЭВМ без участия человека).

Автоматическое проектирование возможно лишь в отдельных частных случаях для сравнительно легких объектов. В настоящее время превалирует автоматизированное проектирование.

Задание на проектирование – первичное описание ЭВА в заданной форме (назначение, параметры, способы функционирования, конструкторская реализация и т. п.).

Проектное решение – промежуточное или конечное описание объекта проектирования, достаточное для определения дальнейших решений.

Результат проектирования – проектное решение или их совокупность, удовлетворяющие заданным требованиям для объекта проектирования (обязательны требования к форме представления проектного решения).

Проектный документ – документ по заданной форме, в котором есть какое-либо проектное решение.

Проект – совокупность проектных документов, в которых представлены результаты проектирования.

Проектная процедура – формализованная совокупность действий, результатом выполнения которых выступает проектное решение (процедуры оптимизации, контроля, поиска решения, трассировка и т. п.).

Проектная операция – действие или совокупность действий составляющих часть проектной процедуры, алгоритм которой остается неизменным для ряда проектных процедур.

Системный подход к проектированию. Основные идеи и принципы проектирования сложных систем выражены в системном подходе. Основной его принцип заключается в рассмотрении частей явления или сложной системы с учетом их взаимодействия. Системный подход включает в себя выявление структуры системы, типизацию связей, выявление атрибутов, анализ влияния внешней среды [3].

Системы автоматизированного проектирования относятся к числу наиболее сложных современных систем. Их проектирование и сопровождение невозможны без системного подхода. Интерпретация и конкретизация системного подхода имеют место в ряде известных подходов, например структурном, блочно-иерархическом и объектно-ориентированном [3].

При *структурном подходе* (как разновидности системного) требуется синтезировать варианты системы из компонентов (блоков) и оценивать варианты при их частичном переборе с предварительным прогнозированием характеристик компонентов.

Блочно-иерархический подход к проектированию использует идеи декомпозиции сложных описаний объектов и соответственно средств их создания на иерархические уровни и аспекты, вводит понятие стиля проектирования (восходящее и нисходящее), устанавливает связь между параметрами соседних иерархических уровней.

Ряд важных структурных принципов, используемых при разработке информационных систем и прежде всего их программного обеспечения (ПО), выражен в *объектно-ориентированном подходе* к проектированию.

На каждом этапе проектирования совершается очередной «виток» прохождения будущего проекта через все критерии и ограничения, выдвигаемые целями проектирования и возможностями проектного предприятия. Процесс проектирования можно представить в виде сходящейся спирали (рис. 1.1).

По мере продвижения к окончательному проекту спираль скручивается, отбрасывая все негодные, побочные и малозначащие проектные решения и другие ненужные продукты процесса проектирования и оставляя только те проектные решения, которые, будучи воплощенными в технической документации, и составят окончательный проект.

Диаметр каждого витка этой условной спирали сужается, отражая характерную черту процесса проектирования – продвижение от расплывчатой, широкой, неопределенной во многих деталях начальной итерации проекта к конкретному, завершенному в технической документации и опытном образце, окончательному проекту.

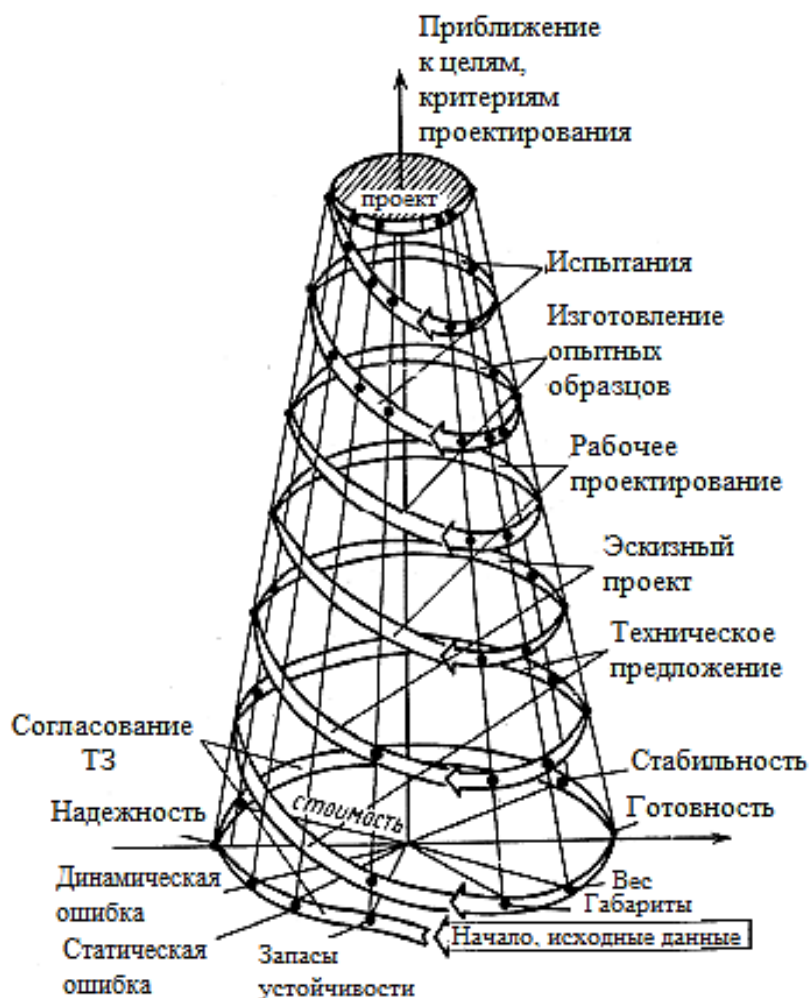


Рис. 1.1. Представление процесса проектирования в виде спирали

1.4. Структура САПР

САПР – сложная система. Как и любая сложная система, она содержит ряд подсистем, которые обладают всеми свойствами систем и могут создаваться как самостоятельные (т. е. это часть системы, выделенная по некоторым признакам) [2; 3].

Различают *проектирующие* и *обслуживающие подсистемы*:

– проектирующие подсистемы (подсистемы логического, схемотехнического, конструкторского проектирования и т. п.) выполняют проектные процедуры;

– обслуживающие подсистемы (обеспечение функционирования проектирующих подсистем, системная среда, подсистемы информационного поиска, выпуска конструкторской документации, графической поддержки проектирования и т. п.).

На рис. 1.2 представлена укрупненно структура САПР [3].

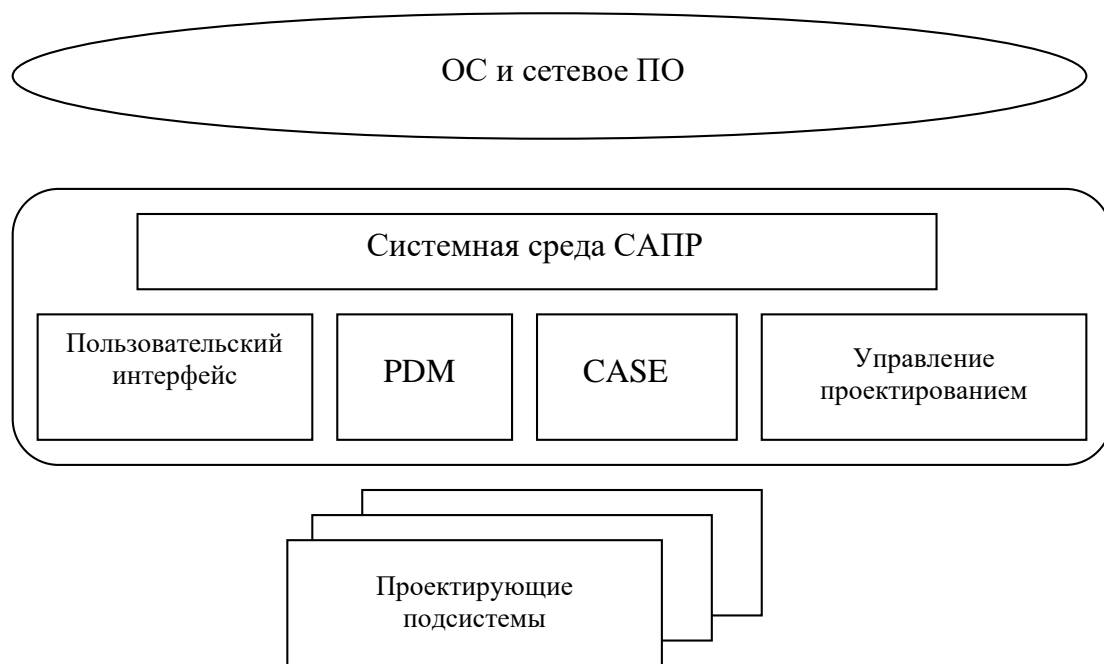


Рис. 1.2. Структура ПО САПР

CALS-технология – это технология комплексной компьютеризации сфер промышленного производства, цель которой – унификация и стандартизация спецификаций промышленной продукции на всех этапах ее жизненного цикла [3]. В CALS-системах предусмотрены хранение, обработка и передача информации в компьютерных средах, оперативный доступ к данным в нужное время и в нужном месте. Соответствующие системы автоматизации назвали автоматизированными логистическими системами, или CALS (Computer Aided Logistic Systems). Поскольку под логистикой обычно понимают дисциплину, посвященную вопросам снабжения и управления запасами, а функции CALS намного шире и связаны со всеми этапами жизненного цикла промышленных изделий, то применяют и более соответствующую предмету расшифровку аббревиатуры CALS – Continuous Acquisition and Lifecycle Support [Там же].

Применение CALS позволяет существенно сократить объемы проектных работ, так как описания многих составных частей оборудования, машин и систем, проектировавшихся ранее, хранятся в базах данных сетевых серверов, доступных любому пользователю технологии CALS. Существенно облегчается решение проблем ремонтпригодности, интеграции продукции в различного рода системы и среды,

адаптации к меняющимся условиям эксплуатации, специализации проектных организаций и т. п.

Важные проблемы, требующие решения при создании комплексных САПР, – управление сложностью проектов и интеграция ПО. Эти проблемы включают вопросы декомпозиции проектов, распараллеливания проектных работ, целостности данных, межпрограммных интерфейсов и др. Для решения проблем совместного функционирования компонентов САПР различного назначения разрабатываются системы управления проектными данными – системы PDM (Product Data Management). Они либо входят в состав модулей конкретной САПР, либо имеют самостоятельное значение и могут работать совместно с разными САПР [3].

Услуги системы SCM – системы управления цепочками поставок – требуются уже на этапе производства. Управление данными в едином информационном пространстве на протяжении всех этапов жизненного цикла изделий возлагается на систему управления жизненным циклом продукции, или PLM-систему (Product Lifecycle Management). Характерная особенность PLM – обеспечение взаимодействия различных автоматизированных систем многих предприятий.

Функции автоматизированных систем управления производством (АСУП) выполняются системами на нескольких иерархических уровнях. Автоматизацию управления на верхних уровнях от корпорации до цеха осуществляют АСУП, классифицируемые как системы ERP (Enterprise Resource Planning) или MRP-2 (Manufacturing Resource Planning) [Там же]. Наиболее развитые системы ERP выполняют различные бизнес-функции, связанные с планированием производства, закупками, сбытом продукции, анализом перспектив маркетинга, управлением финансами, персоналом, складским хозяйством, учетом основных фондов и т. п. Системы MRP-2 ориентированы главным образом на бизнес-функции, непосредственно связанные с производством [Там же].

Система диспетчерского управления и сбора данных SCADA (Supervisory Control and Data Acquisition), кроме диспетчерских функций, выполняет роль инструментальной системы разработки программного обеспечения для промышленных систем компьютерной автоматизации [Там же].

Разновидности САПР. В зависимости от приложений выделяют следующие САПР [3]:

- 1) САПР для машиностроения (MCAD – Mechanical CAD);
- 2) САПР для электроники (ECAD – Electronic CAD или EDA – Electronic Design Automation);
- 3) САПР для архитектуры и строительства;
- 4) технологические САПР (TCAD, CAM – Computer Aided Manufacturing);
- 5) специализированные САПР (САПР интегральных микросхем, САПР летательных аппаратов, САПР электрических машин и др.).

1.5. Схема процесса проектирования в САПР

Задачи, решаемые на каждом этапе и уровне проектирования в САПР, делятся на задачи синтеза и анализа [2]. Задачи синтеза связаны с получением проектных вариантов, а задачи анализа – с оценкой полученных вариантов. Далее будем описывать модель в виде упрощенной зависимости

$$Y = F(Q, X),$$

где Y – вектор выходных параметров; F – функция преобразований модели; Q – вектор внешних параметров; X – вектор внутренних параметров.

Параметры элементов, из которых состоит проектируемое устройство, называются *внутренними*, параметры устройства, по которым оценивается его качество, – *выходными*, параметры действующих на устройство внешних сигналов – *входными*, а параметры окружающей среды – *внешними*. Например, при проектировании схемы усилителя параметры транзисторов, номиналы компонентов – внутренние параметры; коэффициент усиления, потребляемая мощность – выходные параметры; амплитуда и частота входных сигналов – входные параметры; температура окружающей среды – внешний параметр.

Различают синтез параметрический и структурный. *Параметрический* синтез связан с определением численных значений параметров объекта. Цель *структурного* синтеза – получение структуры (состав и связи) объекта, например синтез структуры логической схемы (перечень логических элементов и их связей). При параметрическом синтезе определяются значения параметров выбранной структуры (параметры логических элементов и номиналы в схеме). Если результаты синтеза

будут наилучшими в заданном смысле, то перед нами – задача оптимизации (структурная или параметрическая) [4].

Задачи анализа достаточно разнообразны. Различают *одновариантный* анализ, при котором исследуются свойства объекта в заданной точке пространства параметров, т. е. при заданных значениях внутренних и внешних параметров ($Y_i = F(Q_i, X_i)$), и *многовариантный* анализ, заключающийся в исследовании свойств объекта в окрестности заданной точки пространства параметров ($\Delta Y = F(\Delta Q, \Delta X)$) [2].

Формализация проектных задач. Формализация – возможность составления формального алгоритма проектирования. Обычно это самое необходимое условие для решения задачи на ЭВМ [2].

По способу формализации задачи проектирования делятся:

1) на полностью формализуемые (практически все рутинные задачи, большинство задач анализа); решаются на ЭВМ без вмешательства человека;

2) неформализуемые (задачи творческого характера: выбор принципов построения чего-либо; синтез структур для принципиально новых задач); решаются инженером без помощи ЭВМ;

3) частично формализуемые (очень многочисленная группа), например задачи параметрического или структурного синтеза; решаются на ЭВМ с активным участием человека в интерактивном режиме.

В процессе развития постоянно совершенствуются методы и задачи третьей группы, что приводит к переходу их в первую группу. В то же время совершенствование методов и алгоритмов синтеза приводит к расширению второй группы за счет третьей. Общий научно-технический прогресс в электронике и вычислительной технике всегда будет порождать новые задачи, которые, как правило, первоначально будут относиться к третьей группе. Последнее обстоятельство служит причиной того, что проектирование ЭВА останется автоматизированным при активном участии человека [Там же].

Схема процесса проектирования. Исходными данными для проектирования на очередном уровне является техническое задание (ТЗ), включающее перечисление функций объекта, технические требования (ограничения) на выходные параметры, допустимый диапазон изменений внешних параметров [Там же].

На рис. 1.3 представлена типовая структура процесса проектирования для одного уровня при нисходящем проектировании. Вначале

синтезируется исходный вариант структуры, для него составляется модель и рассчитываются исходные параметры элементов. Анализ модели позволяет принять то или иное решение о дальнейших действиях. Если выполняется условие работоспособности, то синтезированный вариант считают окончательным на данном этапе и, следовательно, приступают к оформлению конструкторской документации и формированию ТЗ для следующего уровня проектирования.



Рис. 1.3. Схема процесса проектирования

Если удовлетворительный результат еще не достигнут, то принимается решение о дальнейших путях совершенствования проекта. Первый вариант – изменение значений параметров элементов, или процедура оптимизации. Второй вариант – изменение структуры объекта, или процедура структурной оптимизации. Данная типовая структура проектирования используется для ручного и автоматизированного проектирования и для всех уровней проектирования.

Особенности данной структуры:

- процесс проектирования носит итерационный характер, итерации могут захватывать несколько уровней;
- наиболее часто повторяемый блок – блок анализа, к нему предъявляют очень жесткие требования по эффективности и вычислительным затратам. С ростом номера итерации модель может уточняться и усложняться;
- блок принятия решений – место проектировщика. Здесь возможны три пути принятия решений. Обычно сначала пытаются добиться выполнения ТЗ в заданной структуре (параметрическая оптимизация), затем при невозможности выполнения условия пытаются изменить структуру (структурная оптимизация). При невозможности удовлетворения ТЗ в различных структурах – переход к предыдущему уровню и изменение ТЗ (смягчение условий), и тогда итерации могут захватить несколько уровней.

Вопросы для самопроверки

1. Роль человека в системах автоматизированного проектирования.
2. Различие нисходящего и восходящего проектирования.
3. Три этапа проектирования ЭВА.
4. Основные задачи функционального уровня проектирования.
5. Основные задачи алгоритмического уровня проектирования.
6. Основные задачи конструкторского уровня проектирования.
7. Основные определения теории проектирования.
8. Особенности системного подхода к проектированию.
9. Состав САПР.
10. Разновидности САПР.
11. Особенности анализа и синтеза.
12. Способы формализации проектирования.
13. Три особенности обобщенной схемы проектирования.

Глава 2. ВИДЫ ОБЕСПЕЧЕНИЯ САПР

Виды обеспечения. Как и в любой другой информационной системе, в САПР выделяют следующие виды обеспечения:

1. Математическое обеспечение (МО) – математические модели и методы, алгоритмы выполнения проектных процедур.

2. Техническое обеспечение (ТО) – технические средства, с помощью которых решаются задачи проектирования.

3. Программное обеспечение (ПО) – совокупность программ, необходимых для выполнения автоматизированного проектирования.

4. Лингвистическое обеспечение (ЛО) – языковые средства, используемые в САПР (языки, используемые для написания программного обеспечения, и входные языки систем проектирования).

5. Информационное обеспечение (ИО) – совокупность сведений, необходимых для проектирования (базы данных и системы управления базами данных).

6. Методическое и организационное обеспечение (МиОО) – совокупность документов по составу и правилам эксплуатации САПР, документация по связям подразделений, где функционирует САПР.

Техническое обеспечение САПР. В настоящее время для решения задач в САПР ЭВА используются стандартные универсальные ЭВМ, обладающие высоким быстродействием и достаточным объемом оперативной и внешней памяти, а также стандартные периферийные устройства. В середине 1980-х годов для задач САПР были разработаны специализированные ЭВМ, рабочие станции (Work Station), а также автоматизированные рабочие места (АРМ) [2]. Но в дальнейшем мировые разработчики САПР ЭВА перешли на разработку программных средств с ориентацией на широко распространенные стандартные технические средства.

2.1. Математическое обеспечение САПР

Математическое обеспечение состоит из математических моделей объектов проектирования, методов анализа и синтеза, а также алгоритмов выполнения проектных операций и процедур. Исходя из схемы процесса проектирования (см. рис. 1.3) основными проектными процедурами, выполняемыми на различных иерархических уровнях, являются синтез структуры, формирование математической модели,

анализ (одновариантный и многовариантный), параметрическая оптимизация. Как правило, основу математического обеспечения САПР составляет математический аппарат для моделирования, анализа и оптимизации.

В математическом обеспечении САПР выделяют *специальную* часть, отражающую специфику объекта проектирования. Эта часть охватывает математические модели, методы и алгоритмы одновариантного анализа, а также большую часть алгоритмов синтеза. *Инвариантная* часть включает в себя методы и алгоритмы многовариантного анализа и параметрической оптимизации [2].

Специальное МО будет рассмотрено в гл. 3 – 7, посвященных различным уровням проектирования ЭВА. Инвариантное МО рассматривается в данной главе.

Отметим, что математическое обеспечение оказывает существенное, а иногда и определяющее влияние на возможности и показатели САПР.

Требования к МО САПР:

1. Универсальность, или применимость к широкому классу объектов. МО САПР обладает высокой степенью универсальности. Например, математическая модель транзистора, разработанная в качестве основы еще в 1960 году, способна отражать основные эффекты для любой области работы транзистора (активная область, области отсечки и насыщения, инверсный режим, включение с общей базой, общим эмиттером, общим коллектором, на низкой и высокой частоте и т. д.), т. е. очень универсальна.

Оценки универсальности не существует, поэтому во всех описаниях и документации должны указываться четкие границы применимости.

2. Алгоритмическая надежность. Все модели и методы представляются в САПР в численном виде, т. е. используются приближенные численные методы расчета, поэтому МО может давать при неверном использовании некорректные результаты. Свойство МО давать правильные результаты называется алгоритмической надежностью. Количественная оценка – это вероятность получения правильных результатов при соблюдении оговоренных ограничений. Если вероятность близка к единице, то метод или модель алгоритмически надежны.

3. Точность. Под точностью понимают степень совпадения расчетных и истинных результатов. Алгоритмически надежные методы могут давать разную точность. Точность оценивают с помощью специально построенных вычислительных экспериментов на ряде тестовых задач [12].

4. Затраты машинного времени. Обычно в САПР затраты времени очень велики, именно они – главный ограничивающий фактор при увеличении размерности решаемых задач. Требование уменьшения временных затрат – одно из главных требований экономичности МО.

5. Используемая память. Это второй показатель экономичности МО (после затрат времени). Память современных ЭВМ растет очень быстро, поэтому разработчики САПР предъявляют требования к сокращению памяти, но не такие жесткие.

Названные требования взаимно противоречивы. Например, более точная и надежная модель часто требует больших вычислительных затрат, чем менее точная и надежная модель. Это противоречие в САПР разрешается с помощью применения библиотек методов, моделей и алгоритмов, в которых элементы различаются по точности, быстродействию и памяти.

Общие способы повышения экономичности МО [2]:

1. Учет разреженности матриц позволяет во многих алгоритмах, в которых используются операции над матрицами, добиться существенного сокращения затрат машинного времени и памяти. Этот учет основан на хранении в памяти ЭВМ только ненулевых элементов матриц и на выполнении арифметических действий только над ними.

2. Исследование сложных моделей по частям (диакоптика). В диакоптических методах производится расчленение математических моделей на части, исследуемые по отдельности.

3. Макромоделирование позволяет использовать при анализе объектов несколько моделей, различающихся сложностью, точностью и полнотой отображаемых свойств.

4. Событийность анализа (избирательность по времени и месту) заключается в том, что при моделировании процессов в каждый момент модельного времени вычисления проводятся только для небольшой части математической модели объекта. Эта часть включает в себя те элементы, состояние которых на очередном временном шаге может изменяться [2].

2.2. Методы многовариантного анализа

Основные виды многовариантного анализа в задачах проектирования – анализ чувствительности и статистический анализ.

Анализ чувствительности. Под чувствительностью понимают влияние изменения внутренних параметров модели на ее выходные характеристики [10; 13]. Цель анализа – определение коэффициентов чувствительности (иногда их называют коэффициентами влияния) для модели $Y = f(X)$, т. е. чувствительности элементов вектора Y к изменению элементов вектора X :

$$S_{ij} = dY_i/dX_j.$$

Анализ чувствительности необходим:

– при определении требований к допускам на параметры компонентов системы, к точности реализации, а следовательно, и к технологии изготовления элементов с целью увеличения процента выхода годных:

$$\Delta X_i = \Delta Y_i/S_i;$$

– параметрической оптимизации градиентными методами (коэффициенты чувствительности являются составляющими вектора градиента):

$$F = \text{grad}(dY/dX_1, dY/dX_2, \dots, dY/dX_n);$$

– анализе отклонений:

$$\Delta Y_i = S_i \Delta X_i.$$

По значениям коэффициентов чувствительности разработчик системы может оценить влияние каждого внутреннего параметра на выходные параметры, выделить наиболее значимые для улучшения выходных характеристик.

Наиболее простым для реализации в программах моделирования и наиболее универсальным по отношению к различным сферам применения считается *метод малых приращений*. Суть метода заключается в замене частных производных малыми приращениями:

$$\frac{dF}{dX} \approx \frac{\Delta F}{\Delta X}. \quad (2.1)$$

В этом случае для расчета чувствительности одного выходного параметра F к изменению n внутренних параметров X_i , $i = 1, 2, \dots, n$ требуется выполнение $(n + 1)$ вариантов расчета (анализа) схемы. В каждом из вариантов делают отклонение ΔX_i одному из параметров X_i , вычисляют выходную функцию $F(X_i + \Delta X_i)$, определяют разность

$\Delta F = F(X_i + \Delta X_i) - F(X_i)$ и вычисляют чувствительность, используя выражение (2.1).

Основное достоинство метода малых приращений – его универсальность, так как он применим для расчета коэффициентов чувствительности любых выходных параметров в любых режимах работы при изменении любого внутреннего параметра схемы X_i . Кроме того, реализация метода в программном обеспечении очень проста.

Однако для данного метода характерны серьезные недостатки, ограничивающие его применение. Один из них – очень большие вычислительные затраты: $(n + 1)$ обращений к программе анализа. Более существенным недостатком представляется его низкая точность вследствие замены частных производных малыми приращениями. Попытки уменьшения ΔX_i могут привести к возникновению очень малой величины ΔF , что из-за погрешности анализа (как разности очень близких значений) может дать неверный результат.

Статистический анализ. Цель анализа – получить оценки рассеяния выходных параметров Y при заданных параметрах рассеяния X . Причины рассеяния параметров – нестабильность внешних факторов, случайность и разброс внутренних параметров, случайность входных воздействий и др. [2].

Результаты анализа – гистограммы выходных параметров, оценки математических ожиданий и среднеквадратических отклонений для выходных параметров и др. Исходными данными могут быть: статистические сведения о рассеянии внутренних параметров, диапазоны изменения и законы распределения внешних параметров.

Статистический анализ применяют для оценки процента выхода годных изделий, серийнопригодности, разбросов выходных параметров и др. Наибольшее распространение в САПР при статистическом анализе получили методы наихудшего случая и статистических испытаний.

Метод наихудшего случая. Служит для получения диапазона возможного рассеяния выходных параметров без оценки плотности распределения этого рассеяния. Законы распределения X_i неизвестны, но можно указать предельно допустимые отклонения ΔX_i .

Пусть на выходной параметр Y_i задано условие работоспособности в виде ограничения $Y_i < T_i$. В этом случае интерес представляет верхняя граница рассеяния, которая достигается в наихудшем случае,

когда все аргументы зависимости $Y = F(X)$ принимают неблагоприятные значения.

Самым неблагоприятным значением аргумента X_i будет максимально возможное значение

$$X_{i\max} = X_{i\text{ном}} + \Delta X_i, \text{ если } Y_i < T_i \text{ и } dY/dX_i > 0 \text{ или } Y_i > T_i \text{ и } dY/dX_i < 0$$

и

$$X_{i\min} = X_{i\text{ном}} - \Delta X_i, \text{ если } Y_i < T_i \text{ и } dY/dX_i < 0 \text{ или } Y_i > T_i \text{ и } dY/dX_i > 0.$$

Каждому выходному параметру будет соответствовать свой наихудший случай. Для m выходных параметров и n внутренних параметров потребуется $(m + n + 1)$ вариантов расчета модели.

Достоинства метода:

- простота реализации;
- малые вычислительные затраты;
- не требуется знания законов распределения внутренних параметров (только отклонения ΔX_i).

Недостатки метода:

- завышенная оценка (мала вероятность наихудшего случая);
- нет возможности получения закона распределения выходных параметров.

Метод статистических испытаний (метод Монте-Карло).

Метод позволяет получить более полные статистические данные о выходных параметрах. Далее приведен алгоритм расчета:

1. Задание значений случайной величины параметра X_i по принятому закону распределения.
2. Расчет Y_i при заданных X_i путем анализа модели $Y = F(X)$.
3. Накопление статистики изменения Y_i .
4. Построение гистограмм и обработка результатов.

Достоинства метода:

- полнота получаемых статистических результатов;
- универсальность.

Недостатки метода:

- большие вычислительные затраты (тысячи испытаний);
- необходимы полные данные о статистических параметрах X_i .

Важно отметить, что с помощью такого анализа можно получить такой производственный показатель, как процент бракованных изделий в готовой продукции (рис. 2.1) [4].

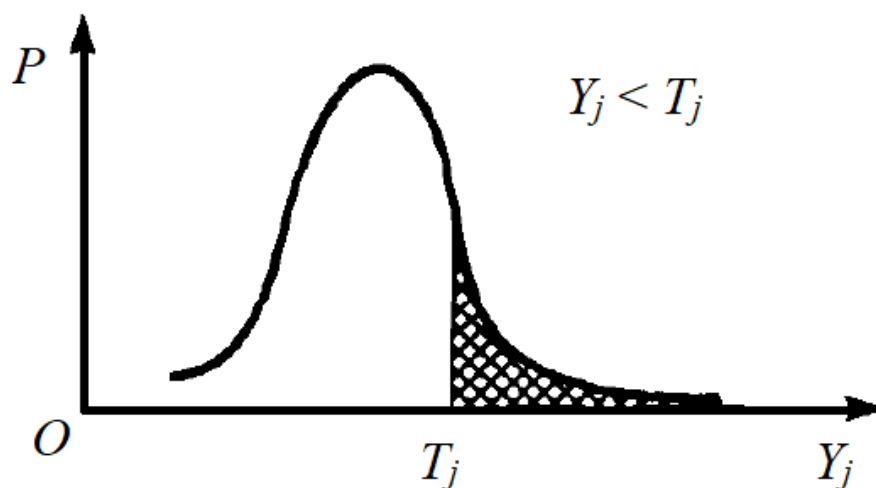


Рис. 2.1. Определение процента выпуска бракованных изделий

На рис. 2.1 представлена плотность распределения выходного параметра Y_j , имеющего условие работоспособности $Y_j < T_j$. Затемненный участок характеризует долю изделий, не удовлетворяющих условию работоспособности параметра Y_j [4].

2.3. Постановка задач оптимизации в САПР

Общие определения. Любая задача проектирования – это выбор оптимального в каком-то смысле решения. Цель оптимизации выражается в критерии оптимальности – правиле предпочтения одного варианта перед другим. Основу критерия составляет целевая функция $F(X)$, где вектор X – вектор варьируемых параметров.

При наличии ограничений задача оптимизации есть задача условной оптимизации, при отсутствии ограничений – задача безусловной оптимизации. Задача оптимизации часто называется задачей математического программирования. Если целевая функция и ограничения линейные, то это задача линейного программирования. Если целевая функция или ограничения нелинейные, то это задача нелинейного программирования. Если часть варьируемых параметров дискретна, то это задача дискретного программирования. Если варьируемые параметры дискретные и целочисленные, то это задача целочисленного программирования.

Постановка задачи оптимизации в САПР ЭВА. Необходимо определить, какими методами решаются задачи оптимизации на различных уровнях.

На *системном уровне* рассматривается задача оптимизации системы, чаще всего задача ставится как задача нелинейного программирования. Здесь целевая функция – один из выходных параметров системы (стоимость, производительность, коэффициент использования оборудования), остальные выходные параметры относят к ограничениям. Варьируемыми параметрами являются емкости динамической памяти, параметры распределений и т. п.

На *функционально-логическом уровне* решаются следующие задачи оптимизации:

1. Минимизация логических функций или числа внутренних состояний конечного автомата – это методы структурной оптимизации, имеющие специфическую постановку.

2. Расчет оптимальных значений параметров логических схем (задержек, мощностей и т. п.): для этого применяют методы нелинейного программирования.

На *схемотехническом уровне* в основном применяют методы нелинейного программирования (параметрической оптимизации). В качестве целевой функции выбирается один из выходных параметров схем (мощность, коэффициент усиления, средняя задержка, уровень нуля, единицы и т. п.), остальные выходные характеристики относят к ограничениям.

На *конструкторском уровне* решаются задачи выбора структуры и их оптимизации; нередко они носят комбинаторный характер. Наиболее часто используются методы целочисленного линейного программирования.

Подведем некоторые итоги:

1. Большинство задач оптимизации в САПР относятся к группе параметрической оптимизации и представляют собой задачи нелинейного программирования. Для него характерна высокая степень универсализации и проработки. Есть большое число отлаженных программных продуктов.

2. Другая группа задач относится к методам структурной оптимизации, для которых характерны разнообразие критериев, преобладание приближительных методик и отсутствие стандартных программ и алгоритмов, а также низкая степень универсализации.

Критерии оптимальности. Имеется два типа критериев оптимальности: частные и обобщенные.

Частные критерии. Характеризуются тем, что в качестве целевой функции выбирается один из показателей качества, а остальные относятся к ограничениям.

Примером может служить критерий максимального приближения (рис. 2.2). Здесь целевая функция представляется в виде взвешенной суммы квадратов разностей полученной и желаемой характеристик:

$$F(x) = \sum W_i \times [P_{\text{расч}}(x, t_i) - P_{\text{жел}}(t_i)]^2, \quad (2.2)$$

где W_i – весовая функция; $P_{\text{расч}}$ – выходной показатель, рассчитываемый по модели; t_i – отсчеты (дискреты) выходного показателя качества; $P_{\text{жел}}$ – желаемые значения выходных параметров.

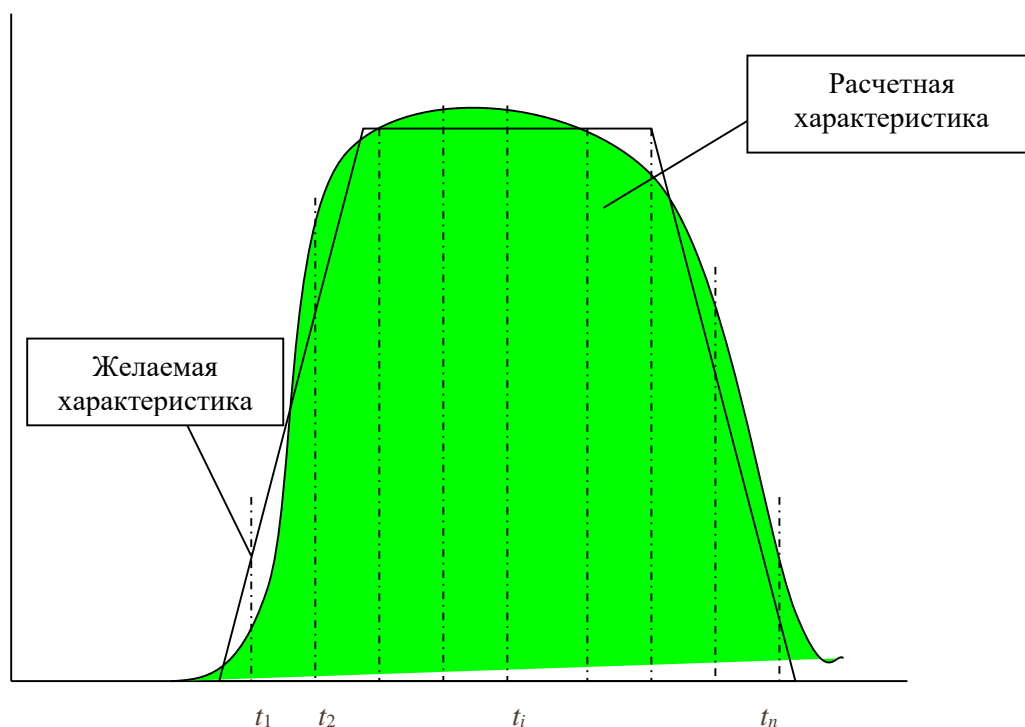


Рис. 2.2. Критерий максимального приближения

Целевая функция – это площадь, подлежащая минимизации, например амплитудно-частотная или переходная характеристики. Вместо (2.2) можно использовать критерий максимального по абсолютной величине отклонения [2]:

$$F(x) = \max [P_{\text{расч}}(x, t_i) - P_{\text{жел}}(t_i)].$$

Недостаток частных критериев – необходимость выбора в качестве целевой функции только одного из показателей качества и отнесение всех остальных в ограничения, т. е. улучшается только один из показателей, а все остальные поддерживаются на определенном уровне.

Обобщенные критерии оптимальности. Один из самых распространенных критериев – *максиминный* или минимаксный подход.

В названном подходе целевая функция представляется как функция минимума от различных Z_i (запасов работоспособности), подлежащая максимизации

$$F(x)_{\max} = \min Z_i(x).$$

Под запасом работоспособности Z_i может быть использована разность между $P_{\text{жел}}$ и $P_{\text{расч}}$ для всех выходных показателей. Кроме максиминного критерия находит применение *мультипликативный* критерий

$$F(x) \prod_{i=1}^R Z_i / \prod_{k=R+1}^n Z_k$$

и *аддитивный* критерий

$$F(x) = \sum_{i=1}^R Z_i - \sum_{k=R+1}^n Z_k.$$

Если минимизировать целевую функцию, то Z_i – выходные параметры, которые необходимо уменьшать, а Z_k – параметры, которые необходимо увеличивать.

Основное достоинство обобщенных критериев – они позволяют оптимизировать модель по всем показателям качества, недостаток – трудность соизмерения различных показателей качества (рубли, вольты, ватты и т. д.) [12].

2.4. Методы оптимизации

При изучении методов оптимизации удобны геометрические представления [2]. *Оптимальной точкой X^** является точка, в которой $F(X)$ достигает наибольшего при максимизации или наименьшего при минимизации значения. В пространстве варьируемых (управляемых) параметров выделяют *гиперповерхности равного уровня*. В точках конкретной гиперповерхности целевая функция равна постоянной величине a . Уравнение гиперповерхности есть $F(X) = a$. В случае двумерного пространства $X = (x_1, x_2)$ гиперповерхности вырождаются в линии равного уровня (рис. 2.3).

Рассмотрим функцию двух переменных Розенброка (рис. 2.4):

$$F(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Минимум этой функции достигается в точке (0,0) и равен нулю.

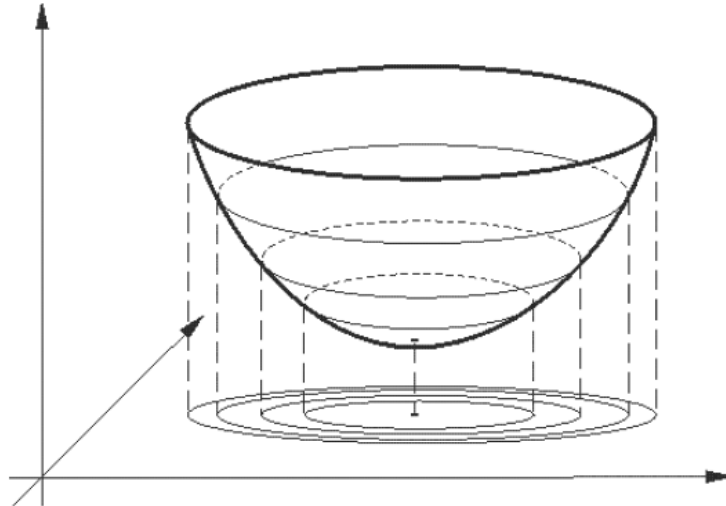


Рис. 2.3. Линии равного уровня

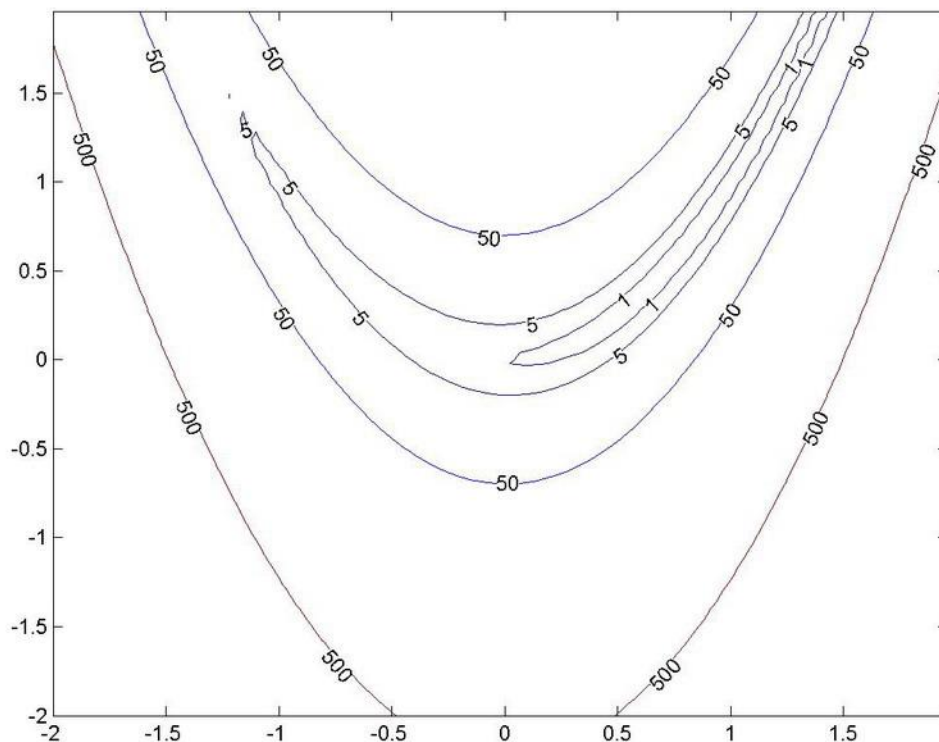


Рис. 2.4. Линии уровня функции Розенброка

Классификация методов оптимизации. Методы оптимизации классифицируют по ряду признаков. Для решения задач безусловной

оптимизации используются численные методы поиска экстремума и методы поиска с помощью производных. Их можно разделить на следующие методы:

1) *прямые методы*, требующие только вычислений целевой функции в точках приближений;

2) *методы первого порядка*, в которых используются вычисления первых частных производных функции;

3) *методы второго порядка*, в которых используются вычисления вторых частных производных.

В зависимости от числа управляемых параметров различают методы *одномерной* и *многомерной оптимизации*. В первом случае управляемый параметр – единственный, во втором – размер вектора X – не менее двух. Реальные задачи в САПР многомерны, а методы одномерной оптимизации играют вспомогательную роль на отдельных этапах многомерного поиска [4]. По наличию или отсутствию ограничений различают методы *условной* и *безусловной* оптимизации. Для реальных задач характерно наличие ограничений, однако методы безусловной оптимизации также представляют интерес, поскольку задачи условной оптимизации с помощью специальных методов могут быть сведены к задачам без ограничений.

В зависимости от числа экстремумов различают задачи одно- и многоэкстремальные. Если метод ориентирован на определение какого-либо локального экстремума, то такой метод относится к *локальным методам*. Если же результатом является глобальный экстремум, то метод называют *методом глобального поиска*. Удовлетворительные по вычислительной эффективности методы глобального поиска для общего случая отсутствуют, и потому на практике в САПР используют методы поиска локальных экстремумов.

Методы одномерного поиска. К методам одномерного поиска относятся методы, использующие правило исключения интервалов, методы золотого сечения и чисел Фибоначчи, методы, использующие производные.

Методы исключения интервалов. Пусть $a < x_1 < x_2 < b$. Сравнив значения $f(x)$ в точках x_1 и x_2 (пробных точках), можно сократить отрезок поиска точки x^* , перейдя к отрезку $[a; x_2]$, если $f(x_1) \leq f(x_2)$, или к отрезку $[x_1; b]$, если $f(x_1) > f(x_2)$. Описанную процедуру можно повто-

рительное число раз, последовательно уменьшая отрезок, содержащий точку минимума. Когда длина последнего из найденных отрезков станет достаточно малой, следует положить $x^* \approx x^0$, где x^0 – одна из точек этого отрезка, например его середина (рис. 2.5).

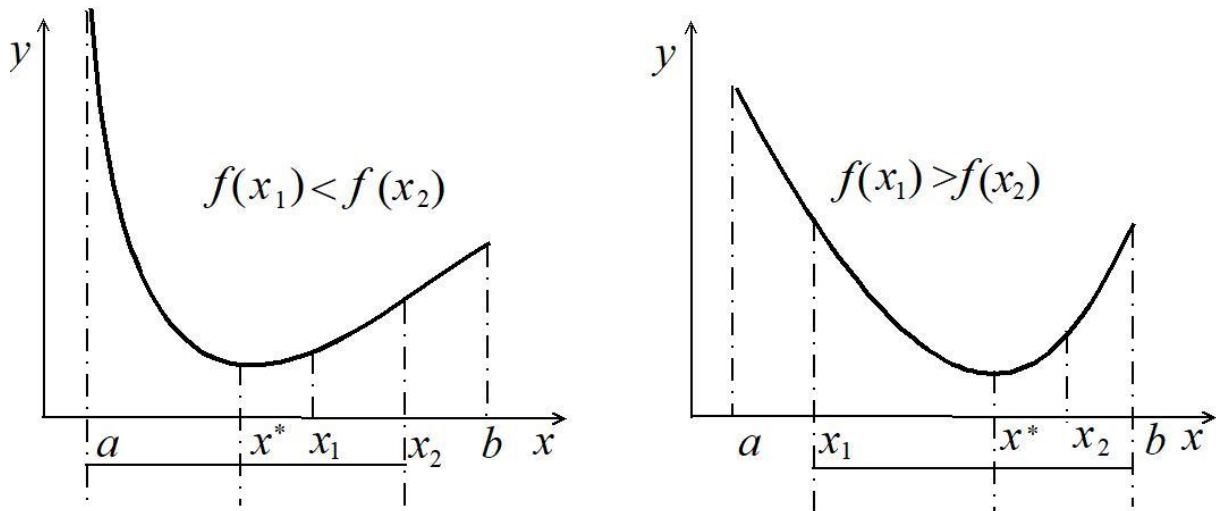


Рис. 2.5. Уменьшение отрезка поиска точки минимума методами исключения отрезков

В зависимости от способа выбора пробных точек получаются различные методы исключения отрезков [14].

Метод половинного деления (дихотомия). С помощью этого метода на каждой итерации получают отрезок, содержащий точку минимума (локализирующий отрезок), такой, что длина каждого следующего отрезка меньше длины предыдущего. Вначале примем $a_1 = a$, $b_1 = b$, выберем некоторое положительное число $\delta > 0$ и вычислим

$$c_1 = \frac{a_1 + b_1}{2} - \frac{\delta}{2}, \quad d_1 = \frac{a_1 + b_1}{2} + \frac{\delta}{2}.$$

Получим и сравним значения функции $f(x)$ в точках c_1 и d_1 . Если $f(c_1) \leq f(d_1)$, то $a_2 = a_1$, $b_2 = d_1$. Если же $f(c_1) > f(d_1)$, то $a_2 = c_1$, $b_2 = b_1$.

Далее берем

$$c_2 = \frac{a_2 + b_2}{2} - \frac{\delta}{2}, \quad d_2 = \frac{a_2 + b_2}{2} + \frac{\delta}{2}$$

и опять сравниваем $f(c_2)$ с $f(d_2)$, определяя новые значения a_3 , b_3 и так далее до тех пор, пока не выполнится условие требуемой погрешности определения точки минимума. Название метода отражает тот факт, что

при малых δ длины локализирующих интервалов на каждом шаге уменьшаются почти в два раза. В отличие от пассивного поиска в методе дихотомии не требуется вычислять значения функции в точках, принадлежащих отбрасываемым на каждом шаге полуинтервалам.

Метод золотого сечения. В методе дихотомии на каждой итерации приходится вычислять значение функции в двух новых точках c_i и d_i . Более эффективным подходом будет алгоритм, в котором одна из этих точек совпадает с одной из аналогичных точек в предыдущей итерации, – это бы позволило на каждой итерации вычислять значение функции лишь в одной новой точке. Эти итерации возможны с помощью деления отрезка в пропорциях золотого сечения. Отрезок AB (рис. 2.6) разделен точкой D в пропорции золотого сечения, если отношение всей длины отрезка к длине большей его части равно отношению длины большей его части к длине меньшей, т. е.

$$\frac{AB}{AD} = \frac{AD}{DB}.$$

Пусть длина $AB = 1$, а $AD = x$. Тогда

$$\frac{1}{x} = \frac{x}{1-x},$$

откуда $x = \frac{\sqrt{5}-1}{2}$. Если больший отрезок отложить не от левого, а от правого конца отрезка, тогда получим точку золотого сечения C , симметричную точке D относительно центра, и $AC = \frac{3-\sqrt{5}}{2}$. Точку C называют первой, а D – второй точкой золотого сечения.



Рис. 2.6. Первая и вторая точки золотого сечения

Использование точек золотого сечения для определения границ локализирующих отрезков позволяет на каждой итерации вычислять значение функции лишь в одной новой точке. Опишем алгоритм метода золотого сечения [14].

На первой итерации принимаем $a_1 = a$, $b_1 = b$ и вычисляем

$$c_1 = \frac{3-\sqrt{5}}{2}(b_1 - a_1) + a_1, \quad d_1 = \frac{\sqrt{5}-1}{2}(b_1 - a_1) + a_1.$$

Получив значения функции f в точках c_1 и d_1 , сравниваем их.

Если $f(c_1) \leq f(d_1)$, то $a_2 = a_1, b_2 = d_1, d_2 = c_1, c_2 = \frac{3 - \sqrt{5}}{2}(b_2 - a_2) + a_2$.

Если $f(c_1) > f(d_1)$, то $a_2 = c_1, b_2 = b_1, c_2 = d_1, d_2 = \frac{\sqrt{5} - 1}{2}(b_2 - a_2) + a_2$.

Далее сравниваем $f(c_2)$ с $f(d_2)$, определяя новые значения a_3, b_3 и так далее до тех пор, пока не выполнится условие требуемой погрешности определения точки минимума.

Метод чисел Фибоначчи. Числа Фибоначчи определяются следующим образом: $F_1 = 1; F_2 = 1; F_{n+2} = F_n + F_{n+1}; n > 1$.

Примем

$$c_1 = a_1 + (b_1 - a_1) \frac{F_n}{F_{n+2}}, \quad d_1 = a_1 + (b_1 - a_1) \frac{F_{n+1}}{F_{n+2}},$$

где n – некоторое натуральное число, которое больше единицы. Получим две симметричные относительно центра отрезка точки, так как

$$c_1 - a_1 = (b_1 - a_1) \frac{F_n}{F_{n+2}} = (b_1 - a_1) \left(1 - \frac{F_{n+1}}{F_{n+2}}\right) = b_1 - d_1.$$

Далее, сравнив значения $f(c_1)$ с $f(d_1)$, нужно отбросить один из полуинтервалов $[a_1; c_1)$ или $(d_1; b_1]$.

Если $f(c_1) \leq f(d_1)$, то отбрасываем $[d_1; b_1]$. Точка c_1 , принадлежащая новому локализирующему отрезку $[a_2; b_2] = [a_1; d_1]$, будет делить его в соотношении

$$\frac{c_1 - a_1}{d_1 - a_1} = \frac{F_n}{F_{n+2}} \bigg/ \frac{F_{n+1}}{F_{n+2}} = \frac{F_n}{F_{n+1}}.$$

В этом случае $d_1 = a_2 + (b_2 - a_2) \frac{F_n}{F_{n+1}} = c_1$.

Если $f(c_1) > f(d_1)$, то отбрасываем $[a_1; c_1]$. Точка d_1 , принадлежащая новому локализирующему отрезку $[a_2; b_2] = [c_1; b_1]$, будет делить его в соотношении

$$\frac{d_1 - c_1}{b_1 - c_1} = \frac{d_1 - c_1}{d_1 - a_1} = \left(\frac{F_{n+1}}{F_{n+2}} - \frac{F_n}{F_{n+2}}\right) \bigg/ \frac{F_{n+1}}{F_{n+2}} = \frac{F_{n-1}}{F_{n+1}}.$$

В этом случае $d_2 = a_2 + (b_2 - a_2) \frac{F_{n-1}}{F_{n+1}} = d_1$. Таким образом, по-
жив $a_1 = a$, $b_1 = b$, $c_i = a_i + (b_i - a_i) \frac{F_{n+1-i}}{F_{n+3-i}}$, $d_i = a_i + (b_i - a_i) \frac{F_{n+2-i}}{F_{n+3-i}}$, на
каждой итерации опять будем вычислять значение функции только в
одной новой точке по соотношению

$$b_n - a_n = (b_1 - a_1) \frac{F_3}{F_{n+2}}.$$

В отличие от методов дихотомии и золотого сечения в методе Фибоначчи не нужно на каждом шаге проверять критерий остановки. Количество шагов здесь задается заранее.

Методы более высокого порядка. К этим методам относятся методы, использующие производные.

Метод касательных. Это метод первого порядка, т. е. такого, который использует первую производную рассматриваемой функции.

Строим касательные к графику $f(x)$ в точках a_1 и b_1 и находим точку их пересечения. Обозначим ее c_1 .

Если $f'(c_1) > 0$, то принимаем $a_2 = a_1$, $b_2 = c_1$.

Если $f'(c_1) < 0$, то принимаем $a_2 = c_1$, $b_2 = b_1$.

Если $f'(c_1) = 0$, то задача решена и точка c_1 – искомая.

Повторяем процедуру до тех пор, пока не выполнится критерий остановки. На рис. 2.7 показаны три итерации метода касательных.

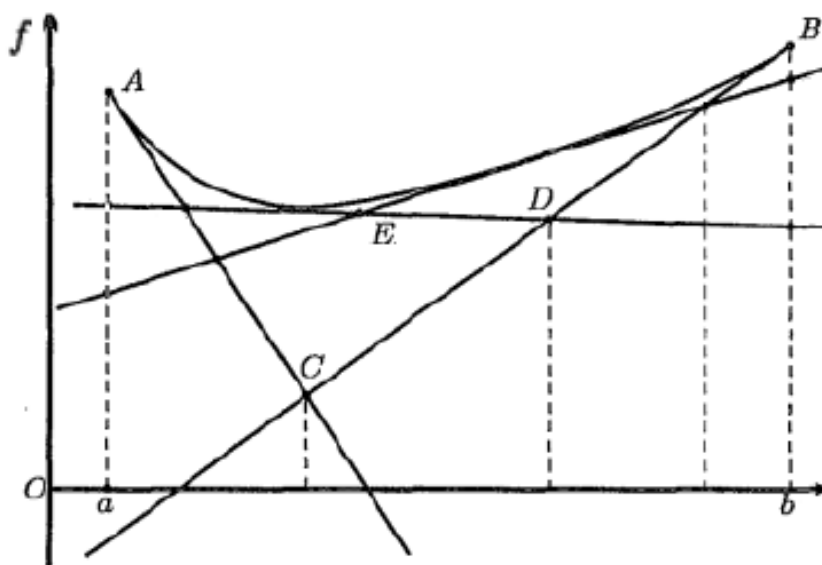


Рис. 2.7. Геометрическая иллюстрация метода касательных

Метод Ньютона – Рафсона. Цель метода – поиск решения уравнения $f'(x) = 0$.

Решение: задаем x_0 , проводим в этой точке касательную к функции $f'(x)$, находим точку пересечения этой касательной с осью Ox . Обозначаем эту точку как x_1 . Продолжаем процедуру до тех пор, пока не выполнится критерий останова.

Ордината точек касательной описывается уравнением

$$f'(x_k) + f''(x_k)(x - x_k),$$

поэтому, приравняв это выражение к нулю, найдем

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}.$$

Данный алгоритм является методом второго порядка (рис. 2.8).

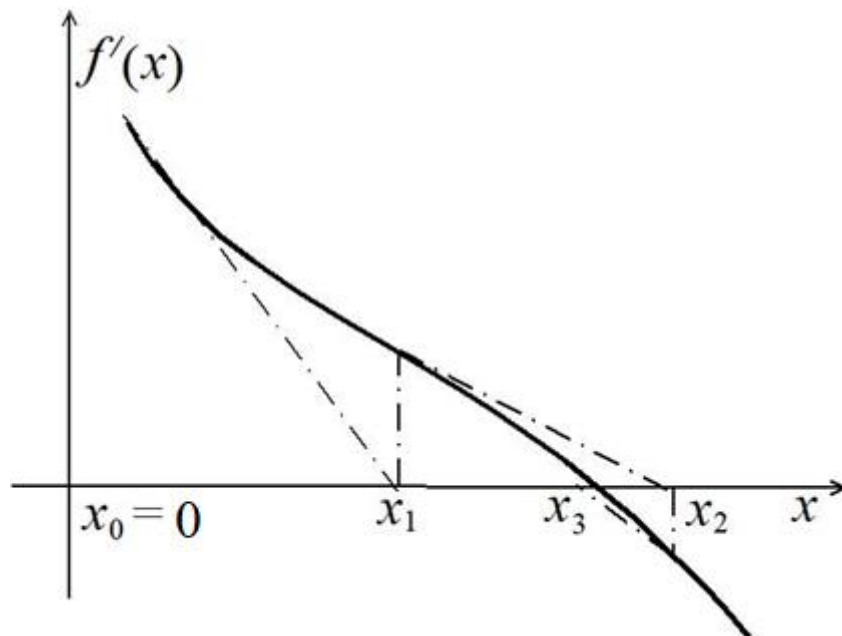


Рис. 2.8. Геометрическая иллюстрация метода Ньютона – Рафсона

Метод секущих. Если в методе Ньютона – Рафсона вместо второй производной воспользоваться ее приближением

$$f''(x_k) \approx -\frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}},$$

то получим метод секущих (метод хорд), являющийся методом первого порядка:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k).$$

Последнее выражение описывает точку пересечения прямой, проходящей через точки $(x_k, f'(x_k))$ и $(x_{k-1}, f'(x_{k-1}))$, с осью Ox (рис. 2.9).

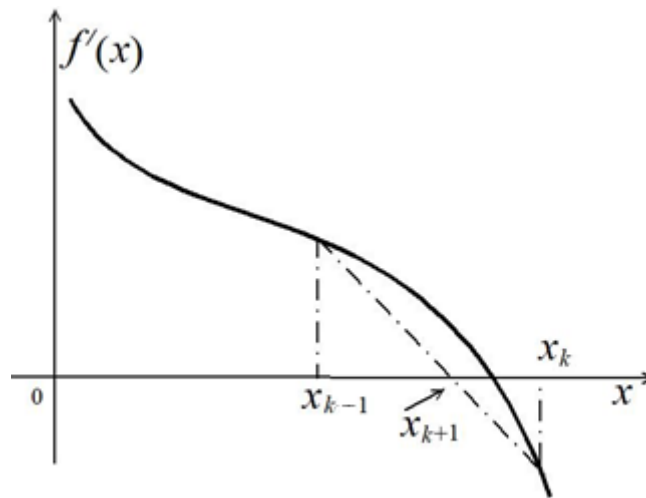


Рис. 2.9. Геометрическая иллюстрация метода секущих

Методы многомерной оптимизации. Среди методов нулевого порядка в САПР широко применяются методы покоординатного спуска, Розенброка, конфигураций. К методам с использованием производных относятся градиентные методы [4].

Метод покоординатного спуска характеризуется выбором направлений поиска поочередно вдоль всех n координатных осей, шаг рассчитывается на основе одномерной оптимизации. Траектория покоординатного спуска для двумерного пространства параметров показана на рис. 2.10.

Пусть $x^k = (x_1^k, x_2^k, \dots, x_n^k)^T$. Алгоритм метода покоординатного спуска осуществляется циклами, по n шагов в каждом.

1. Фиксируем значения x_2^k, \dots, x_n^k и проводим минимизацию по переменной x_1 . Получаем $(x_1^{k+1}, x_2^k, \dots, x_n^k)^T$, на этом первый шаг окончен.

2. Фиксируем $x_1^{k+1}, x_3^k, \dots, x_n^k$ и проводим минимизацию по x_2 . Получаем $(x_1^{k+1}, x_2^{k+1}, x_3^k, \dots, x_n^k)^T$.

3.

4. Фиксируем $x_1^{k+1}, x_2^{k+1}, \dots, x_{n-1}^{k+1}$ и проводим минимизацию по x_n . Получаем $(x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1})^T$.

На рис. 2.10 видно, что с помощью метода покоординатного спуска минимизация ведется поочередно по направлениям, параллельным осям

координат. После n шагов, соответствующих каждой из переменных, завершаем первый цикл, получая $x^{k+1} = (x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1})^T$. Процесс повторяется до тех пор, пока не выполнится условие остановки.

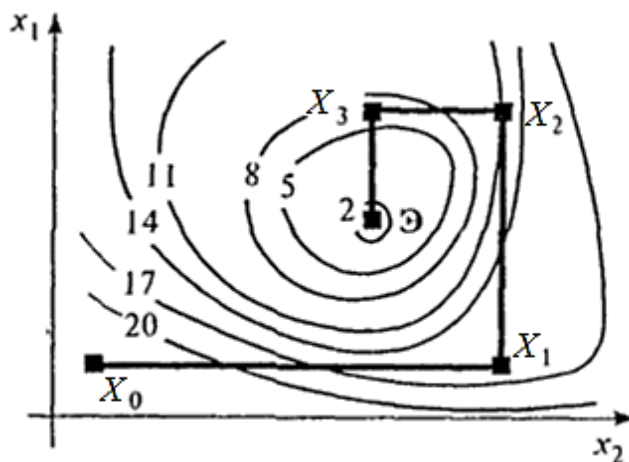


Рис. 2.10. Иллюстрация метода покоординатного спуска [4]

При использовании метода покоординатного спуска велика вероятность застревания на «дне оврага» вдали от точки экстремума. На рис. 2.11 видно, что после попадания в точку A , расположенную на «дне оврага», дальнейшие шаги возможны лишь в направлениях aa или bb , но они приводят к ухудшению целевой функции. Следовательно, поиск прекращается в точке A .

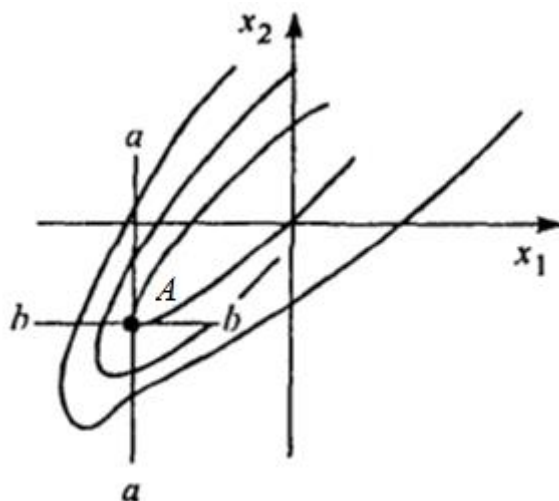


Рис. 2.11. Застревание покоординатного спуска на «дне оврага»

В то же время при благоприятной ориентации «дна оврага», а именно при положении одной из координатных осей, близком к параллельности с «дном оврага», поиск оказывается весьма быстрым. Эта ситуация показана на рис. 2.12.

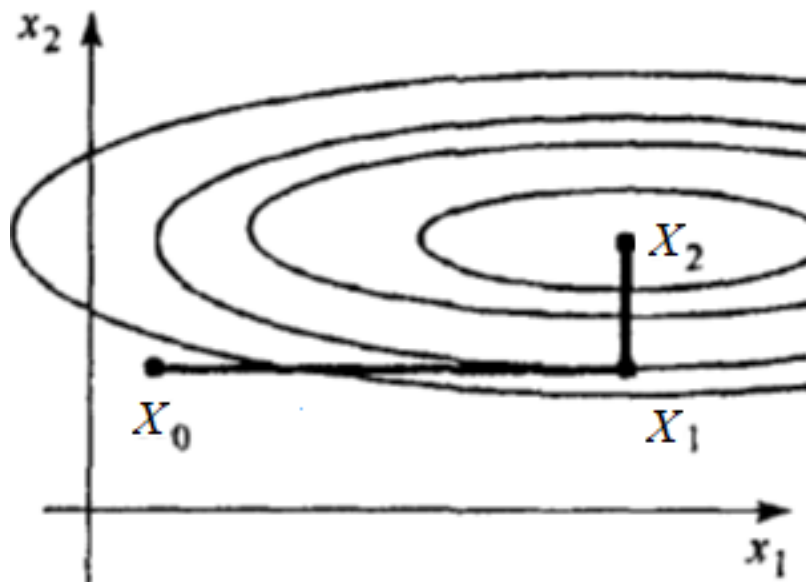


Рис. 2.12. Траектория покоординатного спуска при благоприятной ориентации координатных осей

Метод Розенброка представляет собой улучшенный вариант метода покоординатного спуска и заключается в таком повороте координатных осей, при котором одна из них оказывается квазипараллельной «дну оврага». Такой поворот осуществляют на основе данных, полученных после серии из n шагов покоординатного спуска.

Другая удачная модификация покоординатного спуска – **метод конфигураций (метод Хука – Дживса)**. В соответствии с этим методом вначале выполняют обычную серию из n шагов покоординатного спуска, затем делают дополнительный шаг в направлении, как показано на рис. 2.13, где дополнительный шаг выполняют в направлении вектора $X_3 - X_1$, что и приводит в точку X_4 .

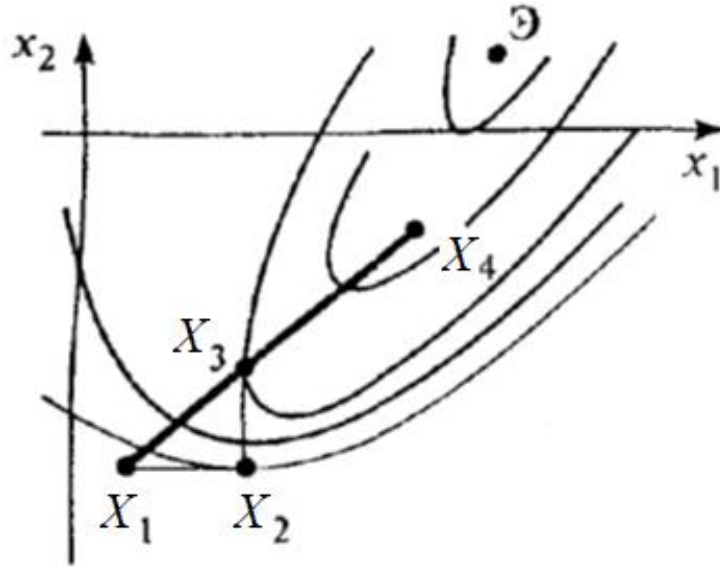


Рис. 2.13. Иллюстрация метода конфигураций

Градиентные методы. Пусть $f'(x^k)$ – градиент функции f в точке x^k . Тогда метод будет определять следующую последовательность шагов $x^{k+1} = x^k - \alpha f'(x^k)$, где α – достаточно малое положительное число. В этом случае имеем $f(x^{k+1}) < f(x^k)$ и можем построить последовательность $\{x^k\}$, на которой значения функции будут убывать. Число α представляет собой длину шага, который необходимо сделать в направлении антиградиента.

Существует несколько вариантов этого метода, отличающихся способом выбора длины шага.

Градиентный метод с дроблением шага. Этот метод предложен в работе [15].

1. Вначале выбирают некоторое $\alpha > 0$ и произвольное число $\varepsilon \in (0,1)$.

2. Далее принимают $\alpha = \check{\alpha}$.

3. Определяют точку $x = x^k - \alpha f'(x^k)$.

4. Вычисляют $f(x) = f(x^k - \alpha f'(x^k))$.

5. Проверяют выполнение неравенств $f(x) - f(x^k) \leq -\alpha\varepsilon \|f'(x^k)\|^2$.

6. Если это неравенство выполняется, то принимают $\alpha_k = \alpha$. Если же неравенство не выполняется, то производят деление α путем умножения на произвольное положительное $\lambda < 1$ и переходят к шагу 3.

7. Получают $x^{k+1} = x^k - \alpha_k f'(x^k)$, проверяют критерий остановки (например, $\|f'(x^{k+1})\| < \delta$, для некоторого малого $\delta > 0$) и переходят к следующей итерации.

Значение функции на каждом шаге меньше, чем на предыдущем. Таким образом, на последовательности $\{x^k\}$ функция убывает, но это убывание становится все медленнее по мере приближения к точке экстремума (рис. 2.14).

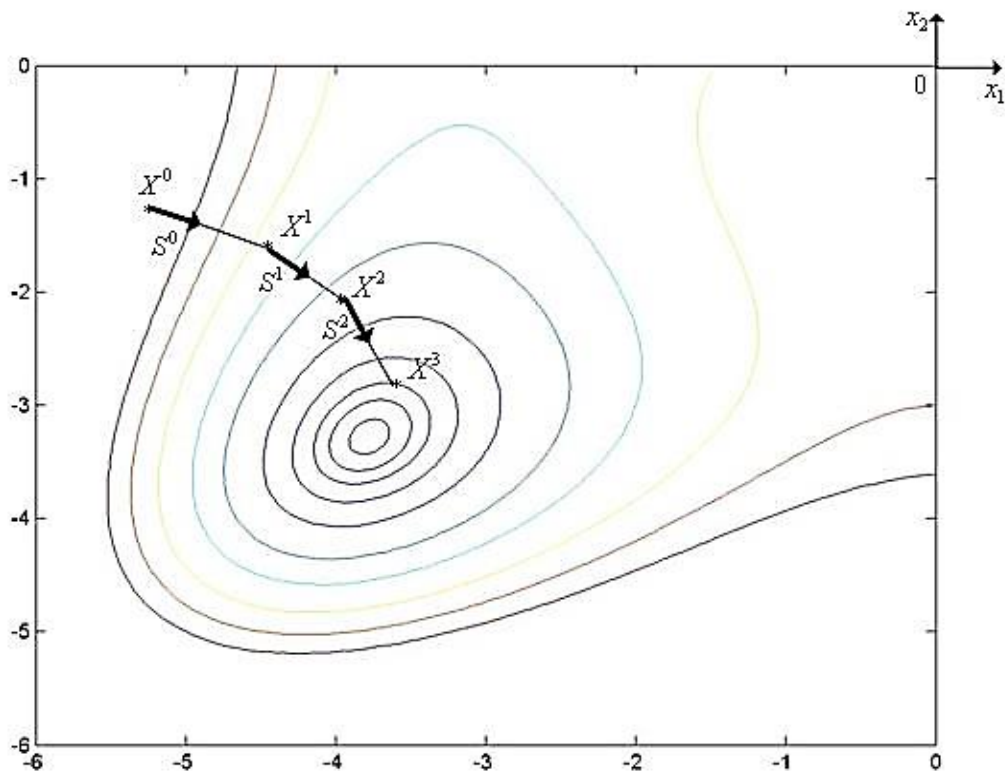


Рис. 2.14. Траектория поиска минимума функции Химмельблау градиентным методом с дроблением шага

Метод наискорейшего градиентного спуска. Можно искать оптимальную длину шага, т. е. α_k , доставляющее минимум функции в направлении антиградиента:

$$f(x^k - \alpha_k f'(x^k)) = \min f(x^k - \alpha f'(x^k)).$$

Заметим, что по необходимому условию экстремума оптимальное α должно удовлетворять

$$\frac{d}{d\alpha} f(x^k - \alpha f'(x^k)) = 0,$$

т. е. $\langle -f'(x^k), f'(x^k - \alpha f'(x^k)) \rangle = 0$. Определив отсюда α_k , получим

$$\langle -f'(x^k), f'(x^{k+1}) \rangle = 0,$$

т. е. направления, вдоль которых ведется минимизация с помощью метода наискорейшего градиентного спуска, на соседних шагах ортогональны (рис. 2.15).

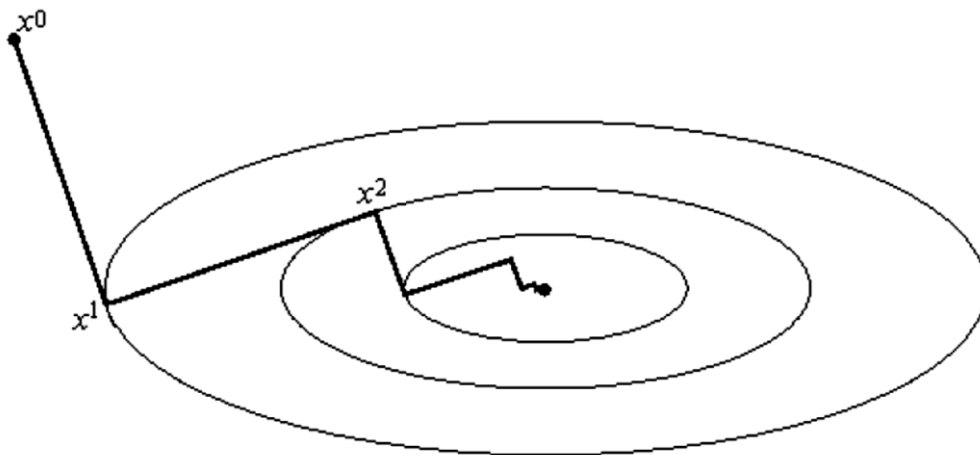


Рис. 2.15. Метод наискорейшего градиентного спуска

Метод сопряженных направлений (градиентов). Рассмотрим квадратичную функцию

$$q(x) = \frac{1}{2} x^T A x + b^T x + c,$$

где A – симметричная матрица; b – некоторый вектор, $b = (b_1, b_2, \dots, b_n)^T$; c – константа.

Линейно независимые направления $d_i \in R^n$, $i = 0, \dots, n - 1$ называются попарно сопряженными относительно квадратичной формы $q(x)$, если

$$d_i^T A d_j = 0, \text{ при } i, j = 0, \dots, n - 1, i \neq j.$$

Если минимизировать $q(x)$ из начальной точки x_0 последовательно вдоль направлений d_i , то точка минимума будет найдена не более чем за n шагов.

Данный подход хорошо работает на квадратичных функциях, позволяя минимизировать их за конечное число шагов. Для произвольных функций метод также дает хорошие результаты.

Метод Ньютона. Это метод второго порядка, в котором применяется матрица Гессе $f''(x)$ – матрица вторых производных. Ввиду использования большего количества информации этот метод должен сходиться быстрее методов первого порядка.

Точка минимума x^* , в которой выполнено условие минимума $f'(x^*) = 0$. Пусть имеется некоторая точка x^k . Рассмотрим разложение градиента в ряд Тейлора в окрестности этой точки:

$$f'(x) = f'(x^k) + \langle f''(x^k), x - x^k \rangle + o(\|x - x^k\|).$$

Идея метода Ньютона заключается в замене функции $f'(x)$ в окрестности точки x^k ее аппроксимацией $f'(x^k) + \langle f''(x^k); x - x^k \rangle$. Точку, в которой эта аппроксимация равна нулю, берут в качестве x^{k+1} :

$$x^{k+1} = x^k + [f''(x^k)]^{-1} f'(x^k). \quad (2.3)$$

Выражение (2.3) – основное в методе Ньютона. Отметим, что этот процесс может расходиться, когда начальная точка далека от x^* . Для того чтобы преодолеть эту проблему, алгоритм (2.3) часто модифицируют. Направление $p_k = -[f''(x^k)]^{-1} f'(x^k)$ будет направлением спуска в точке x^k . Таким образом, классический метод Ньютона можно рассматривать как процесс: $x^{k+1} = x^k + \alpha_k p_k$, где $\alpha_k = 1$.

В модифицированном методе Ньютона предлагается искать α_k с помощью решения задачи одномерной минимизации.

Квазиньютоновский метод. Основная трудность при реализации метода Ньютона – вычисление и обращение матрицы вторых производных. Если удастся заменить в выражении (2.3) матрицу вторых производных на некоторую ее аппроксимацию H_k так, чтобы последовательность H_k за конечное число шагов сходилась к матрице вторых производных, то последний шаг этого метода совпадет с шагом по методу Ньютона, который для квадратичной функции сходится за одну итерацию.

Можем записать приближенное равенство

$$[f''(x^k)]^{-1} (f'(x^{k+1}) - f'(x^k)) \approx x^{k+1} - x^k,$$

которое для квадратичной функции с отличным от нуля гессианом, очевидно, превращается в точное равенство.

Используя это приближенное равенство и информацию о полученной точке x^{k+1} , можно определить следующую матрицу H_{k+1} :

$$H_{k+1} \left(f'(x^{k+1}) - f'(x^k) \right) = x^{k+1} - x^k. \quad (2.4)$$

Условие (2.4) называют квазиньютоновским. Будем искать H_{k+1} в виде

$$H_{k+1} = H_k + U_k,$$

где матрица U_k – матрица поправок. Примем $U_k = uv^T$, где u и v – n -мерные векторы-столбцы, $(H_k + uv^T)\gamma_k = \delta_k$, откуда

$$u = \frac{1}{v^T \gamma_k} (\delta_k - H_k \gamma_k).$$

Окончательное выражение [14]

$$H_{k+1} = H_k + \frac{(\delta_k - H_k \gamma_k)(\delta_k - H_k \gamma_k)^T}{(\delta_k - H_k \gamma_k)^T \gamma_k}.$$

Условная оптимизация. Решается задача

$$\begin{cases} F(x) \rightarrow \min, \\ x \in X, \end{cases}$$

где X – некоторое множество, называемое допустимым.

Будем считать, что X определяется ограничениями в виде неравенств и равенств нулю некоторых достаточно гладких функций:

$$\begin{cases} F(x) \rightarrow \min, \\ \varphi_i(x) > 0, \quad i = 1, \dots, m, \\ h_j(x) = 0, \quad j = 1, \dots, l. \end{cases}$$

Метод штрафных функций. Если взять функцию

$$H(x) = \begin{cases} 0, & x \in X, \\ +\infty, & x \notin X, \end{cases}$$

то задачу условной оптимизации можно заменить на задачу безусловной оптимизации функции $F(x) + H(x)$, тогда минимум $F(x) + H(x)$ на всем пространстве совпадает с минимумом $F(x)$ на X . Проблемой на этом пути является отсутствие непрерывности у построенной таким образом функции $F(x) + H(x)$. Поэтому $H(x)$ аппроксимируют непрерывной функцией, которая принимает положительные значения при $x \notin X$ и равна нулю при $x \in X$, т. е. x штрафуются за выход из допустимого множества.

Рассмотрим функцию

$$\Phi(x) = F(x) + rH(x),$$

где r – скаляр, называемый коэффициентом штрафа, $r > 0$. На практике используют идею постепенного увеличения параметра штрафа. Причем на каждой итерации в качестве начальной берут точку, полученную на предыдущем шаге. Это позволяет каждый раз запускать алгоритм безусловной оптимизации из некоторой окрестности точки минимума.

Среди методов штрафных функций различают методы внутренней и внешней точки. Согласно методам внутренней точки (иначе называемым методами барьерных функций), исходную для поиска точку можно выбирать только внутри допустимой области, а для методов внешней точки – как внутри, так и вне допустимой области (важно лишь, чтобы в ней функции целевая и ограничений были определены). Ситуация появления барьера у целевой функции $\Phi(x)$ и соотношение между условным в точке x_2 и безусловным в точке x_1 минимумами $F(x)$ в простейшем одномерном случае показана на рис. 2.16 [4], где под функцией $S(x)$ принято $H(x)$.

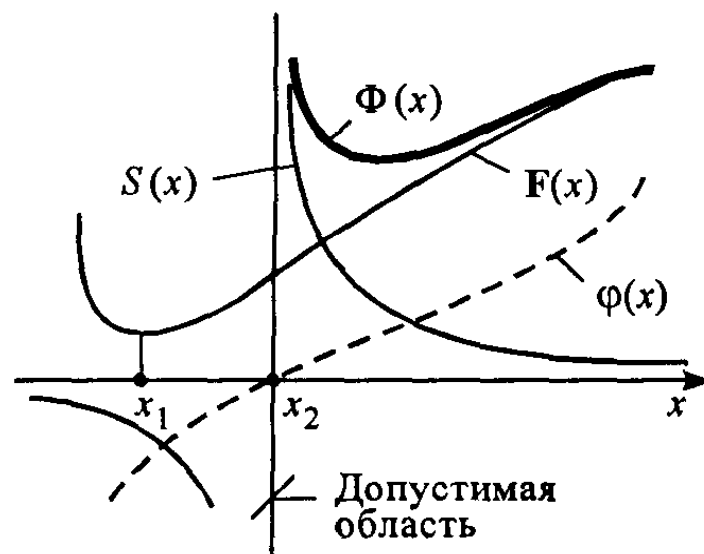


Рис. 2.16. Пояснение метода штрафных функций

Примеры штрафных функций [4]:

1) для метода внутренней точки при ограничениях $\varphi_i(x) > 0$

$$H(x) = \sum_{i=1}^m (1/\varphi_i(x)),$$

где m – число ограничений типа неравенств;

2) для метода внешней точки при таких же ограничениях

$$H(x) = \sum_{i=1}^m (\min[0, \varphi_i(x)])^2,$$

здесь штраф сводится к включению в $\Phi(x)$ суммы квадратов активных (т. е. нарушенных) ограничений;

3) в случае ограничений типа равенств $h_j(x) = 0$

$$H(x) = \sum_{j=1}^l (h_j(x))^2.$$

Основной вариант *метода проекции градиента* ориентирован на задачи математического программирования с ограничениями типа равенств.

Поиск при выполнении ограничений реализуется в подпространстве $(n - m)$ измерений, где n – число управляемых параметров; m – число ограничений. При этом движение осуществляется в направлении проекции градиента целевой функции $F(x)$ на гиперплоскость, касательную к гиперповерхности ограничений (точнее, к гиперповерхности пересечения гиперповерхностей ограничений) [4].

Поиск минимума начинают со спуска из исходной точки на гиперповерхность ограничений. Далее выполняют шаг в указанном выше направлении (шаг вдоль гиперповерхности ограничений). Поскольку этот шаг может привести к заметному нарушению ограничений, вновь повторяют спуск на гиперповерхность ограничений и т. д. Другими словами, поиск заключается в выполнении пар шагов, каждая пара включает спуск на гиперповерхность ограничений и движение вдоль нее.

Идею метода легко пояснить для случая поиска в двумерном пространстве при одном ограничении $h(x) = 0$. На рис. 2.17 это ограничение представлено жирной линией, а целевая функция – совокупностью более тонких линий равного уровня. Спуск обычно выполняют по нормали к гиперповерхности ограничений (в данном случае к линии ограничения). Условие окончания поиска основано на сопоставлении значений целевой функции в двух последовательных точках, получаемых после спуска на гиперповерхность ограничений [Там же].

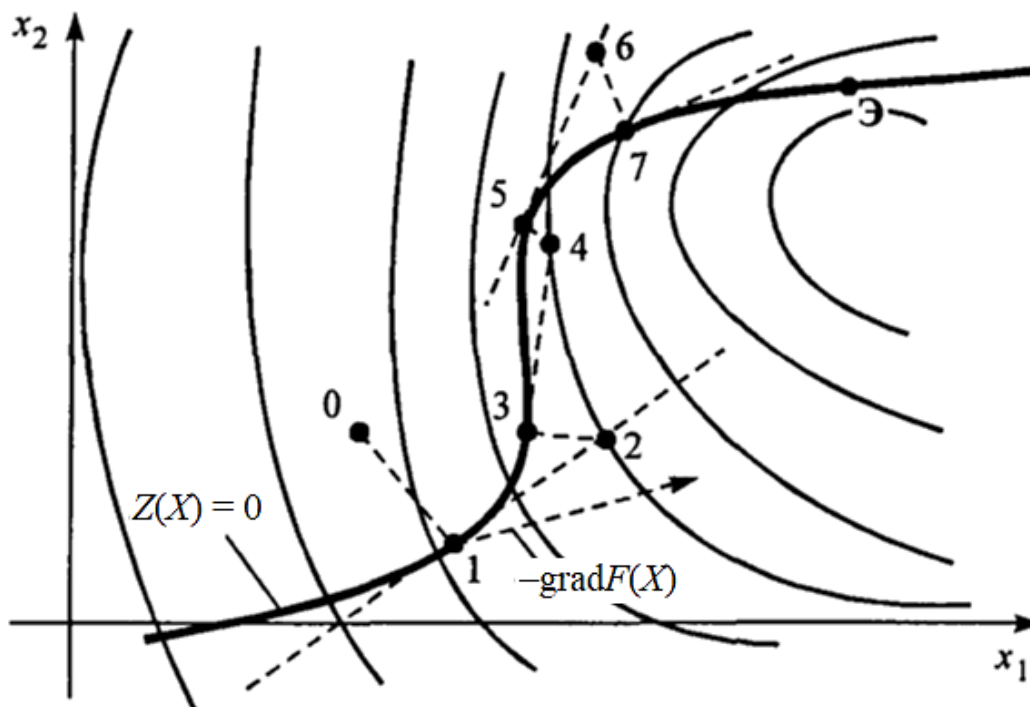


Рис. 2.17. Траектория поиска в соответствии с методом проекции градиента: Э – условный экстремум; 0, 1, 2, ..., 7 – точки на траектории поиска

2.5. Лингвистическое и программное обеспечение

Лингвистическое обеспечение (ЛО) САПР делится на две группы языков: языки, на которых пишутся программные средства САПР, и входные языки систем САПР (языки подготовки заданий на проектирование – языки общения пользователя с САПР) [2].

В качестве языков программирования используют стандартные языки: ранее это были Фортран, ПЛ/1, Паскаль, а в настоящее время чаще всего – язык Си.

Требования к входным языкам:

1. Универсальность (возможность описания на входном языке любых объектов, на которые ориентирована САПР).
2. Удобство (проблемная ориентация входного языка). Максимальное удобство для описания и восприятия данных.
3. Максимальная лаконичность описания и однозначность истолкования элементов и конструкций языка.
4. Возможность развития и расширения языка.

При описании задания на проектирование выделяют:

- 1) язык описания объекта (ЯОО) проектирования;
- 2) язык описания задания на проектирование (на выполнение проектных операций и процедур).

В свою очередь ЯОО делятся:

- на схемные (описание принципиальных, функциональных и структурных схем и т. п.);
- графические (геометрические и топологические объекты);
- языки моделирования (алгоритмы функционирования, имитационные эксперименты).

Кроме этого, ЯОО делят:

- а) на процедурные (для описания развивающихся во времени процессов проектирования; похожи на алгоритмические языки);
- б) непроцедурные (для описания структур объектов, более удобны для пользователя).

Программное обеспечение. Это совокупность программных средств, необходимых для выполнения автоматизированного проектирования. Именно программное обеспечение (ПО) связывает воедино все остальные виды обеспечений в САПР. ПО САПР – одно из самых сложных видов ПО, требует огромных трудозатрат на создание (до тысячи человеко-лет). Сложность и необходимость ПО САПР определяет и стоимость САПР (например, стоимость полной версии самой дорогой САПР фирмы CADENCE – 2 млн долл.). Сроки разработки сложных САПР превышают 5 – 7 лет.

ПО САПР делят:

- на общее ПО САПР (ОПО). Его функция – обеспечение организации функционирования САПР и технических средств (планирование и управление вычислительным процессом и проектированием, распределение ресурсов). Все перечисленные выше функции – это функции операционных систем (ОС). В настоящее время нет специализации ОС для САПР, т. е. используются стандартные универсальные ОС ЭВМ;
- специальное ПО САПР (СПО). Отвечает за математическое обеспечение для непосредственных задач проектирования. Чаще всего СПО реализуется в виде пакетов прикладных программ (ППП).

Иногда выделяют базовое программное обеспечение (БПО) САПР – это вспомогательные средства САПР (подсистемы машинной графики, СУБД и др.).

Состав и функции СПО САПР. В общем виде состав СПО САПР представлен на рис. 2.18.

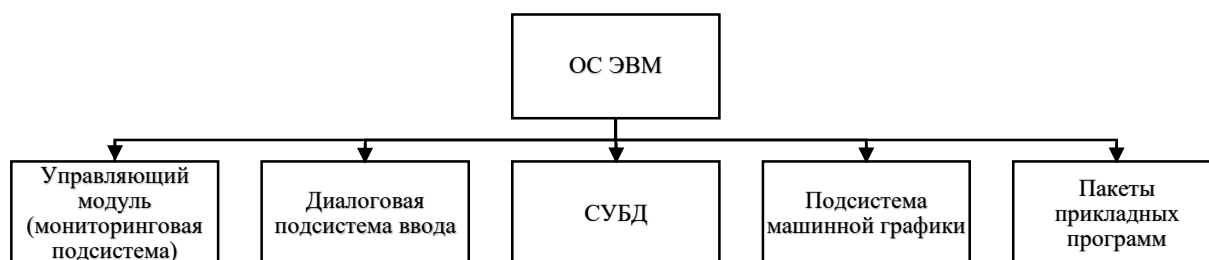


Рис. 2.18. Состав СПО САПР

Основная часть СПО – *мониторинговая подсистема САПР*. Её назначение – организация и оптимизация управления процессом при выполнении проектных процедур и взаимодействия других подсистем САПР.

Функции мониторинговой системы:

- управление процессом реализации проектных процедур и операций;
- организация взаимодействия подсистем САПР;
- интерпретация языковых форм заданий на проект;
- распределение ресурсов САПР в процессе проектирования (время, память и т. п.);
- обеспечение диалоговых и интерактивных режимов работы.

Функции похожи на ОС ЭВМ. Отличие состоит в том, что здесь не дублируются функции ОС ЭВМ, а выполняются функции внутри системы САПР.

Диалоговая подсистема обеспечивает прямое интерактивное взаимодействие пользователя со средствами автоматизированного проектирования в реальном времени.

Программное обеспечение для выполнения проектных операций и процедур формируется в виде пакетов прикладных программ. Существуют две организационные структуры ППП: простая и сложная. Для ППП простой структуры все управляющие функции возложены на общую ОС ЭВМ. К сожалению, ППП простой структуры удобны при реализации лишь небольшой части маршрутов проектирования для слабосвязанных задач или для единственной задачи.

При сложных взаимодействиях программ в маршрутах проектирования возможностей ППП простой структуры недостаточно. Необходимы дополнительные функции управления заданиями, что очень похоже на организацию ПО на самом верхнем уровне подсистем проектирования и САПР. Поэтому ППП сложной структуры можно представить в виде управляющей части и набора функциональных модулей (рис. 2.19).

В зависимости от назначения и сложности ППП управляющая часть может включать один или несколько следующих элементов:

- локальную диалоговую подсистему ввода/вывода;
- локальную БД ППП;
- локальный монитор.

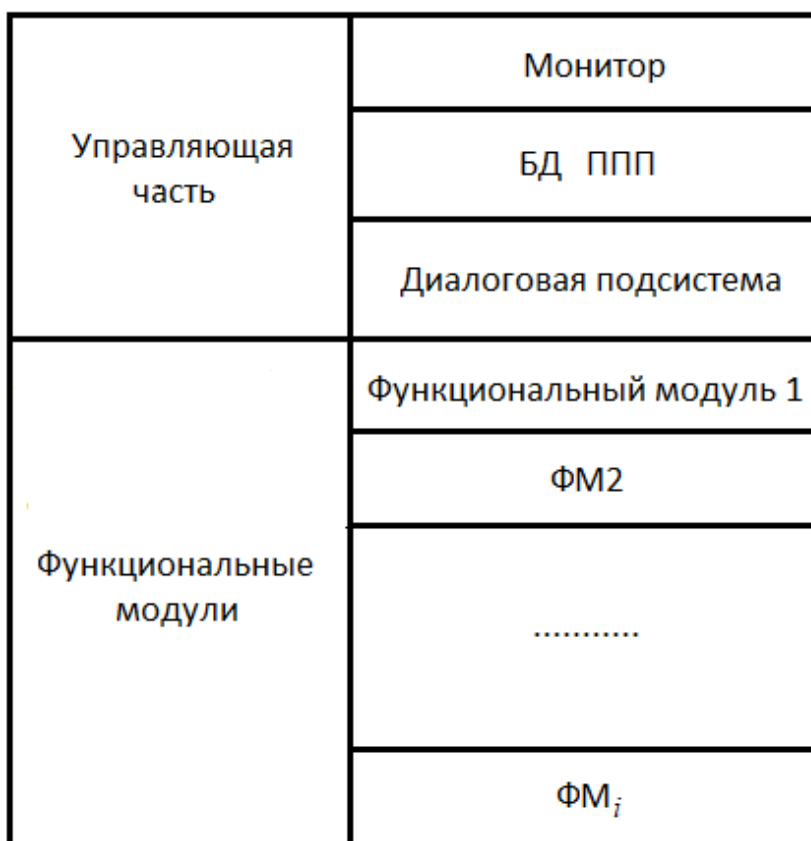


Рис. 2.19. Структура ППП

Функциональные модули (ФМ) реализуют типовые элементы математического обеспечения, типовые проектные процедуры и операции. ФМ предназначены для многократного использования (это математические модели элементов, численные методы, модели внешних

воздействий, графические операции – обычно они все объединяются в библиотеки).

Языковые процессоры (способы получения объектных программ). Языковые процессоры (ЯП) предназначены для преобразования информации, выраженной на входных языках, в рабочую программу.

Различают следующие виды ЯП:

- 1) интерпретатор;
- 2) компилятор (транслятор).

Интерпретатор. Работу интерпретатора можно представить в виде следующего упрощенного алгоритма:

а) описание объекта и маршрута (задания) на проектирование на входном языке воспринимается языковым процессором и генерируется структура данных в виде таблиц, векторов, матриц, списков и т. п., удобная для дальнейшей обработки;

б) затем эта структура данных интерпретируется обрабатывающей подсистемой.

До начала работы системы имеется полностью скомпонованная рабочая программа, состоящая из универсальных модулей, пригодных для всех классов решаемых задач. Вызов необходимых модулей выполняется на этапе расчетов.

Отметим, что здесь интерпретация противоположна широко известному языку Бейсик, где интерпретация выполняется по отношению к операторам языка программирования. Здесь же интерпретация используется не по отношению к входному языку, а к промежуточным данным – внутреннему представлению данных (в виде таблиц, векторов, матриц и т. д.).

Компилятор. В работе используется следующий алгоритм:

а) до начала работы не имеется полностью скомпонованной рабочей программы;

б) ЯП считывает задание на проект; выходом процессора будет текст программы вычислений по заданному объекту и маршруту проектирования на каком-то объектном языке;

в) если текст программы или язык – это язык машинных команд, то это означает конец работы программы, в ином случае требуется трансляция.

Таким образом компилятор на первом шаге выполняет всю основную работу по составлению рабочей программы из исполнимых библиотек функциональных модулей.

Наиболее эффективная форма ЯП – гибридная (когда используются и методы компиляции, и методы интерпретации). Самые чувствительные места программы (по затратам времени и памяти) – используют компиляторы, а типовые процедуры – в виде интерпретаторов.

Сравнение видов ЯП:

1. Компиляторы сложнее для реализации, но требуют меньших вычислительных затрат (память и время), так как убрана лишняя логика (операторы *if-else*) в программах, а также вызов лишних модулей программ (модели, анализы и т. п.).

2. Если сравнить отдельные части программ, полученных при компиляции и интерпретации, то больший объем памяти требуется для компиляторов, но при этом общий объем памяти ЯП меньше.

3. Так как компиляция выполняется один раз и занимает небольшую долю всего времени процесса проектирования, то требования к компиляторам ниже, чем к интерпретаторам.

4. В компиляторах удобно обрабатывать любые произвольные функции, что трудно выполнить для интерпретатора.

Вопросы для самопроверки

1. Требования к математическому обеспечению.
2. Суть метода малых приращений для расчета чувствительности.
3. Метод наихудшего случая.
4. Особенности метода статистических испытаний.
5. Постановка задач оптимизации в САПР.
6. Методы одномерной оптимизации.
7. Методы многомерной оптимизации.
8. Суть методов условной оптимизации.
9. Классификация входных языков.
10. Состав программного обеспечения САПР.
11. Состав пакетов прикладных программ.
12. Особенности языковых процессоров.

Глава 3. ЭТАПЫ СИСТЕМНОГО И ФУНКЦИОНАЛЬНОГО ПРОЕКТИРОВАНИЯ В САПР ЭВА

3.1. Этап системного проектирования

Нисходящее проектирование вычислительных систем (ВС) начинается с решения задач системного уровня. На системном уровне разрабатываются структурные схемы ЭВА, поэтому его часто называют структурным [2]. На этом уровне ведется укрупненное рассмотрение всей ЭВМ или ВС в целом. Элементы, из которых будет строиться система, – это довольно крупные узлы ЭВМ, такие как процессоры, каналы, запоминающие устройства, периферийные устройства, хотя степень детализации системы может быть очень глубокой. С алгоритмической точки зрения это разработка архитектуры ЭВА: описание структуры ЭВА, т. е. перечня ее составных частей и связей между ними; форматы представления различных данных, системы и форматы команд, организация адресации; принципы выполнения операций, приоритеты, дисциплины обслуживания; общие технические характеристики ЭВА и ее частей, т. е. производительность или пропускная способность, надежность и др.

Основные задачи:

- 1) определение принципов организации ВС;
- 2) выбор архитектуры и разделение функций на программную и аппаратную реализацию;
- 3) разработка структурной схемы ВС, удовлетворяющей требованиям таких системных параметров, как производительность, пропускная способность, надежность и др.;
- 4) определение требований к выходным параметрам и характеристикам компонентов, формирование ТЗ на отдельные компоненты ВС.

Особенности системного уровня проектирования:

1. Функционирование ВС рассматривается с информационной точки зрения (процесс преобразования информации) при большой степени абстрагирования от физической сущности протекающих в системе процессов.

2. Характер информационных процессов в ВС в значительной мере определяется программным обеспечением, т. е. ВС рассматривается как совокупность аппаратных и программных средств.

3. Моделирование выполняется как многовариантный анализ: варьируя характеристики элементов и связи между ними, определяют такую структуру ВС, которая обладает заданными (необходимыми) характеристиками.

4. Задачи моделирования отличаются большим разнообразием и трудно поддаются формализации.

5. В ЭВМ с помощью математических моделей ВС имитируется процесс ввода, обработки и вывода данных в соответствии с назначением рассматриваемой ВС. Процедуры получения математической модели объекта и имитации его функционирования на этой модели выполняются методом имитационного моделирования.

6. На системном уровне моделирования в меньшей мере, чем на других уровнях, удается автоматизировать процедуру составления математических моделей ВС на основе моделей компонентов.

7. Соответствие ВС своему назначению невозможно проверить на примере одной задачи, решаемой в ВС. Необходимо обобщение для всего класса решаемых задач. Такое обобщение достигается при учете статистических закономерностей поступления и обработки данных, т. е. используются вероятностные методы.

8. Наиболее часто ВС моделируются с позиций систем массового обслуживания (СМО), с помощью сетей Петри, моделей систем автоматического регулирования, а также с помощью дифференциальных уравнений.

9. Возможна постоянная детализация моделей ВС. Число уровней детализации может быть очень большим.

Методы, алгоритмы и модели системного уровня проектирования ЭВА подробно рассматриваются в учебном курсе «Моделирование», а также в учебном пособии [17].

3.2. Задачи и особенности функционального проектирования

На функциональном уровне проектирования в качестве основных элементов системы рассматриваются такие устройства, как процессоры, каналы связи, запоминающие и другие устройства, блоки и узлы ЭВА.

Основная задача при функциональном моделировании (ФЛМ) – проверка работоспособности функциональных и принципиальных

схем устройств при заданных входных воздействиях. Иногда при моделировании выделяют подуровни: логический (вентильный, logical, gate level) и регистровых передач (RTL – register transfer level).

На *логическом уровне* в качестве компонентов используются простейшие логические элементы – вентили (реализующие простейшие булевы функции И, ИЛИ, И-НЕ) и триггеры различных типов [3].

На *регистровом уровне* выполняется моделирование устройств (узлов, блоков), где в качестве компонентов (элементов, из которых состоит данное устройство) используются интегральные микросхемы средней степени интеграции (регистры, счетчики, дешифраторы, логические преобразователи, цепи межрегистровых передач и т. п.).

Но так как модели этих подуровней очень схожи и используются примерно одинаковые методы решения основных уравнений, то обычно их рассматривают вместе, к тому же в программном обеспечении современных систем моделирования они реализуются совместно.

Проектирование на функциональном уровне выполняется с позиций преобразования сигналов (обозначаются просто x или y) без конкретизации (физической сущности) того, что понимают под сигналом (ток или напряжение). Напомним, что на системном уровне был рассмотрен процесс преобразования информации.

Основная особенность моделирования на функциональном уровне (ФУ) состоит в том, что оно выполняется как многоступенчатое. Первоначально проверяют схему на соответствие заданным функциям без учета задержек сигналов и ограничений элементной базы. Такая проверка проводится очень быстро даже для очень сложных схем и позволяет выявить ошибки в схеме. На последующих этапах выполняется более подробный анализ с учетом реальных задержек сигналов и реальных технологий изготовления.

Другая особенность ФУ – математические модели и методы моделирования для аналоговой и цифровой частей ЭВА сильно различаются. Для аналоговых схем используют методы макро моделирования, функционального моделирования и теории автоматического управления. Для цифровых схем – методы математической логики и теории конечных автоматов. В последние годы разработаны эффективные подходы к совместному аналого-цифровому (смешанному, mixed signal) моделированию.

3.3. Моделирование цифровых функционально-логических схем

Задачи проектирования цифровых функционально-логических (ФЛ) схем:

1. Синтез функциональных и принципиальных схем отдельных узлов и блоков.
2. Проверка работоспособности синтезированных блоков с учетом задержек сигналов и ограничений выбранной элементной базы.
3. Синтез контролирующих и диагностических тестов.
4. Формирование ТЗ для схемотехнического уровня.

Особенности:

1. Задачи синтеза решаются только для конкретных технологий, в общем случае в автоматическом режиме решаются только для небольших размерностей.

2. Анализ – основная операция проектирования для оценки и отбора вариантов построения схем. Оценка многоступенчатая. На первых этапах проверяют схему на соответствие заданным функциям без учета задержек сигналов и ограничений элементной базы. На последующих этапах выполняется более подробный анализ с учетом реальных задержек сигналов.

3. Переменные модели представляются в дискретной форме (цифровая информация). Чаще всего это булевы переменные (0 и 1). Но в ряде случаев используют и многозначную логику (трех-пятизначные и более).

4. Анализ выполняется в дискретном времени. Ось времени разбивается на такты. Часто используется относительное время, номер такта. События происходят мгновенно.

Математические модели компонентов (элементов). Математическая модель элемента (ММЭ) или компонента (ММК) может быть представлена в виде таблиц истинности и состояний или конечным автоматом. Таблицы истинности и состояний очень удобны при моделировании, но для сложных компонентов могут потребовать слишком много памяти для хранения. Автомат Мура в общем случае может задаваться следующей системой уравнений:

$$\left. \begin{aligned} Y &= \varphi(X, A), \\ A' &= \psi(X, A), \end{aligned} \right\} \quad (3.1)$$

где Y – вектор выходных для элемента переменных; X – вектор входных переменных; A – вектор, характеризующий внутренние состояния. Элементы векторов X и A относятся к относительному моменту времени t , а элементы векторов A' и Y – к моменту времени $t + t_3$, где t_3 – задержка, которая может быть различной для элементов этих векторов.

Пример. Рассмотрим модель простейшего двухвходового элемента И-НЕ. Традиционно в алгебре логики элемент И-НЕ представляется соотношением

$$a = \overline{b \wedge c}.$$

Для математического моделирования соотношение преобразуется к виду

$$Y(t) = \overline{X_1(t) \wedge X_2(t)}$$

или

$$Y(t + t_3) = \overline{X_1(t) \wedge X_2(t)}$$

при учете задержек. Таким образом, для элементов комбинационных схем достаточно воспользоваться для описания только первым уравнением системы (3.1):

$$Y = \varphi(X),$$

а для последовательностных элементов схем (т. е. схем с элементами обратных связей и памяти), например триггеров, появятся составляющие вектора A (внутренние состояния). Например, модель T -триггера [18] можно представить в виде

$$A(t + t_3) = \overline{X(t) \wedge A(t)} \vee \overline{X(t) \wedge A(t)}.$$

Математическая модель схемы. Математическая модель схемы (ММС) для ФЛ моделирования – это совокупность ММЭ, в которой отождествляются переменные, относящиеся к соединенным выводам элементов [2]. Рассмотрим схему (рис. 3.1), которая содержит комбинационные элементы И-НЕ, с моделями, в которых задержка не учитывается (в данном случае параметр t – время – может быть опущен).

Тогда математической моделью схемы будет совокупность следующих уравнений (отождествление переменных осуществляется простым обозначением всех шин схемы своими именами):

$$e = \overline{a \wedge b};$$

$$f = \overline{e \wedge c};$$

$$h = \overline{g \wedge f};$$

$$g = \overline{d \wedge h}.$$

Последовательность записи уравнений произвольная в общем случае, если не оговорены особые условия упорядочивания. С системой логических уравнений такого вида неудобно работать, поэтому для упрощения записи и обобщения введем обозначения:

U – вектор значений, входных для всей схемы шин, в частном случае схемы (рис. 3.1) $U = (a, b, c, d)$;

V – вектор значений промежуточных (внутренних) и выходных шин схемы, для данной схемы $V = (e, g, h, f)$. Тогда уравнение ММС примет вид

$$V = F(V, U). \quad (3.2)$$

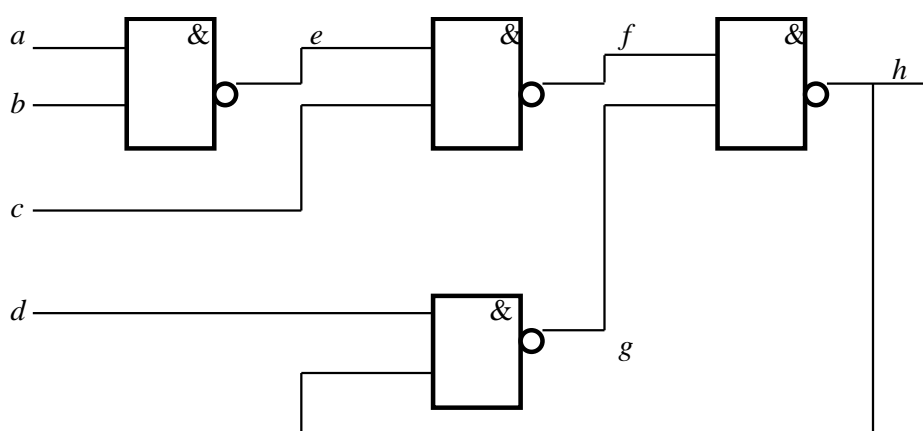


Рис. 3.1. Пример комбинационной схемы

Если для модели схемы учесть задержки и обозначить как V' вектор промежуточных и выходных значений в момент времени $t + t_3$, то получим

$$V' = F(V, U). \quad (3.3)$$

Модель (3.3) называется асинхронной моделью функциональной схемы, а модель (3.2) – синхронной.

Способы задания задержек. При моделировании электронных схем в программном обеспечении задержка может быть задана как фиксированная постоянная для каждого элемента величина, как случайная величина, как функция выходных характеристик элемента (нагрузочная способность, число элементов в цепи) или как точная величина, возвращаемая программой размещения и трассировки данной схемы в конкретном технологическом базисе (в зарубежных программах – back annotation).

Синхронные и асинхронные модели. Как уже отмечалось, модель функциональной схемы может быть синхронной или асинхронной.

Синхронная модель (3.2) наиболее экономична. В ней не учитывается задержка, поэтому время в уравнениях схемы не фигурирует. В то же время названная модель является частным случаем модели (3.3) и, по сути, представляет собой систему логических уравнений. Синхронная модель в первую очередь предназначена для моделирования синхронных схем, в которых передача информации между регистрами разрешается только при наличии специальных синхронизирующих импульсов; период синхроимпульсов T_c обычно выбирается из условия, что все переходные процессы межрегистровых передач в схеме за этот период заканчиваются. Если это условие выполняется, то работоспособность схемы может быть проверена лишь с помощью анализа в установившемся режиме. Синхронная модель выступает дискретным аналогом непрерывной статической модели в виде конечных уравнений (как будет рассмотрено далее). Системы логических уравнений обычно решают итерационными методами. Результат решения – установившиеся значения вектора V на данном периоде синхроимпульсов [2; 3].

Синхронная модель применяется:

- для проверки корректности соединений элементов в функциональной схеме;
- выявления рисков сбоя;
- синтеза тестов.

В синхронных моделях используются как булевы (двузначные, двоичные переменные), так и многозначные переменные. В некоторых программах моделирования возможна оценка до 128 состояний модели. В качестве примеров применения многозначной логики рассмотрим задачу определения рисков сбоя. *Риск сбоя* – это возможность появления ложных сигналов. Различают статический и динамический риски сбоев [2; 19].

Статический риск сбоя – это возможность изменения переменной на выходе какого-либо элемента в условиях, когда при правильном функционировании такого не должно быть.

Пример. Проиллюстрируем статический риск сбоя на примере двухвходового элемента И-НЕ (рис. 3.2, а) и временной диаграммы (рис. 3.2, б).

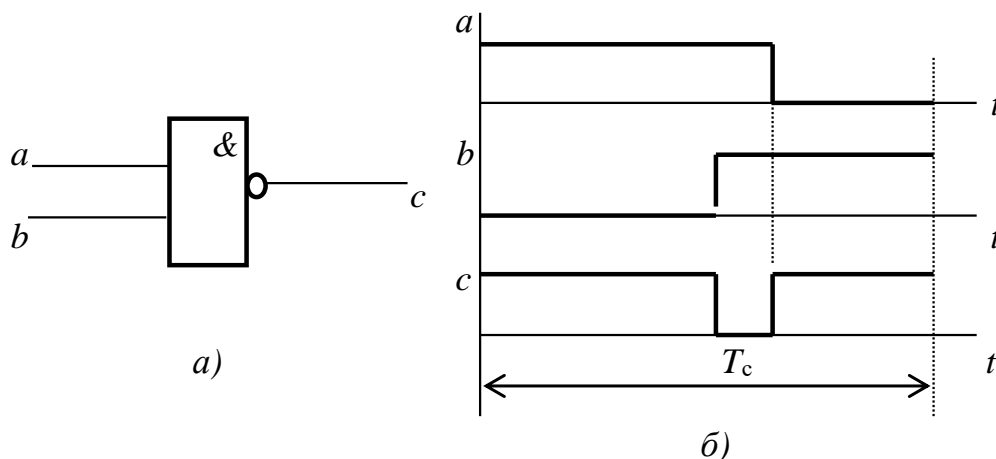


Рис. 3.2. Статический риск сбоя: а – схематический элемент; б – диаграмма

При одновременном переключении сигнала на шинах a и b изменений в сигнале по шине c не будет. Но так как сигналы по шинам проходят через разное число элементов с задержками, возможно неодновременное изменение – тогда появится ложный сигнал. Выявить его с помощью двузначной модели невозможно, так как нет такого промежуточного состояния, которое бы характеризовало это неопределенное состояние. Отразить статический риск сбоя можно с помощью трехзначных (троичных) моделей [2].

Введем новое состояние в схеме (которое задает неопределенное состояние соответствующего входа или выхода элемента) и обозначим как 2 (часто эти дополнительные состояния обозначаются буквами), что соответствует переходу $0 - 1$ или $1 - 0$ и является выражением статического риска сбоя. Правила выполнения основных операций (таблица истинности) в трехзначной логике приведены в табл. 2.

При моделировании многозначной логики требуется два расчета (анализа) переменных модели. Сначала вычисляются промежуточные значения, а затем – окончательные. При анализе промежуточное значение для входной переменной задается равным двум тогда, когда исходное и окончательное значения не одинаковы, в остальных случаях действует правило операций трехзначной логики по табл. 2 [Там же].

Таблица 2

Таблица истинности для трехзначной логики

<i>a</i>	0	1	2	0	1	2	0	1	2
<i>b</i>	0	0	0	1	1	1	2	2	2
И	0	0	0	0	1	2	0	2	2
ИЛИ	0	1	2	1	1	1	2	1	2
НЕ	1	0	2	–	–	–	–	–	–

Пример. Пусть задан фрагмент схемы, представленной на рис. 3.3. Модель фрагмента определяется следующими уравнениями:

$$e = a \wedge b, \quad g = \overline{e} \vee c.$$

Статический риск сбоя в этом фрагменте иллюстрирует табл. 3. Значение сигнала 2 по шине *g* (должно быть $g = 0$) во время переходного процесса (промежуточные значения) говорит о том, что возможен риск статического сбоя (появление ложного сигнала) [2].

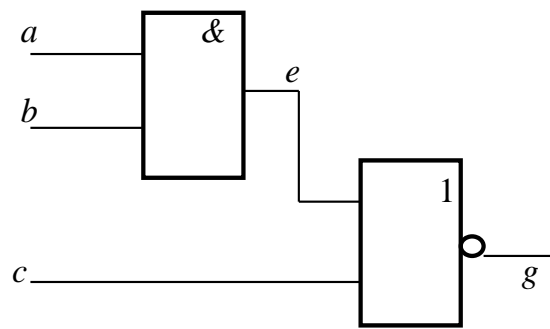


Рис. 3.3. Фрагмент схемы

Таблица 3

Результаты анализа схемы рис. 3.3

Значения переменных	<i>a</i>	<i>b</i>	<i>c</i>	<i>e</i>	<i>g</i>
Исходные	1	0	1	0	0
Промежуточные	1	2	0	2	2
Окончательные	1	1	2	1	0

Динамический риск сбоя – это опасность многократных изменений выходной переменной вместо правильного однократного изменения. Для схемы, показанной на рис. 3.3, возможна ситуация, изображенная на рис. 3.4. Если воспользоваться трехзначной моделью для схемы рис. 3.3, то она даст результаты, приведенные в табл. 4, из которой видно, что динамический риск сбоя по шине *g* такая модель не отражает, хотя статический риск сбоя по шине *e* модель указывает.

Таблица 4

Анализ схемы с трехзначной моделью

Значения переменных	a	b	c	e	g
Исходные	1	0	1	0	0
Промежуточные	2	2	2	2	2
Окончательные	0	1	0	0	1

Выявить динамический риск сбоя позволяет пятизначная логика. В пятизначной логике дополнительно вводятся состояние 3, соответствующее переходу из 1 в 0, и 4 – для перехода из 0 в 1. Состояние 2

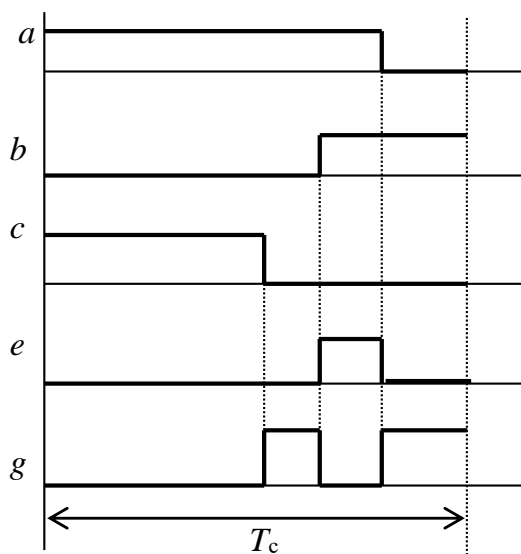


Рис. 3.4. Динамический риск сбоя

будет соответствовать статическому риску сбоя. Табл. 5 представляет собой таблицу истинности для пятизначной логики.

Пример. Для фрагмента схемы рис. 3.3 в ситуации, заданной диаграммой рис. 3.4, результаты расчета с использованием пятизначных моделей представлены в табл. 6.

Эти результаты указывают на динамический риск сбоя по шине g (значение 2 в промежуточных состояниях, где при правильном функционировании должно быть 4), а по шине e – на статический риск сбоя.

Эти результаты указывают на динамический риск сбоя по шине g (значение 2 в промежуточных состояниях, где при правильном функционировании должно быть 4), а по шине e – на статический риск сбоя.

Таблица 5

Таблица истинности для пятизначной логики

a	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
b	0	0	0	0	0	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4
И	0	0	0	0	0	0	1	2	3	4	0	2	2	2	2	0	3	2	3	2	0	4	2	2	4
ИЛИ	0	1	2	3	4	1	1	1	1	1	2	1	2	2	2	3	1	2	3	2	4	1	2	2	4
НЕ	1	0	2	4	3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Анализ схемы с пятизначной моделью

Значения переменных	<i>a</i>	<i>b</i>	<i>c</i>	<i>e</i>	<i>g</i>
Исходные	1	0	1	0	0
Промежуточные	3	4	3	2	2
Окончательные	0	1	0	0	1

Асинхронная модель (3.3) учитывает задержки элементов и представляет собой совокупность рекуррентных уравнений, позволяющих при известном векторе V в предыдущих временных точках и заданном законе изменения входных переменных U вычислять значения вектора V во всем интересующем пользователя диапазоне времени, т. е. построить временные диаграммы работы схемы. Асинхронная модель наиболее универсальна, может применяться как для синхронных, так и для асинхронных схем, но требует больших временных затрат. Это дискретный аналог динамической модели с непрерывными переменными.

Обнаружение рисков сбоя с помощью синхронной модели не гарантирует, что сбой действительно будет. Результаты синхронного моделирования аналогичны результатам расчета в наихудшем случае. Более точные результаты можно получить с помощью асинхронных моделей (анализа во временной области – *timing analysis*). Эта модель позволяет воспроизвести все временные последовательности событий с построением временных диаграмм для любых последовательностей на входных шинах. По временным диаграммам можно проследить все критические состояния, учесть задержки в каждом логическом блоке, выявить «состязания» сигналов.

В асинхронных моделях задержки элементов $t_3 = \Delta t$ учитываются в виде относительных задержек K_3 , равных округленным до целого значения отношения $\Delta t/T$, где T – длительность такта моделирования. Так выбирается из условия, что он не превосходит минимальной задержки в схеме. Модель позволяет получить полную характеристику работы схемы во времени с большой точностью, но за счет увеличения трудоемкости.

В асинхронных моделях так же, как и в синхронных, может использоваться многозначная логика. Рассмотрим на примере учета разброса задержек использование трехзначной логики [2]. Если задержка

элемента имеет минимальное K_{\min} и максимальное K_{\max} значения, а входной сигнал поступает в момент времени t , то выходная переменная будет иметь значения 1 или 0 до момента времени $t + K_{\min}$ и после момента $t + K_{\max}$, а между этими моментами состояние элемента считается неопределенным и отображается состоянием 2. В данном примере многозначная логика в асинхронной модели выполняет функции, аналогичные расчету наихудшего случая. Правила выполнения логических операций в асинхронных моделях те же, что и в синхронных. Анализ выполняется дважды, но по сравнению с методом статистических испытаний Монте-Карло получается значительный выигрыш по времени.

Обобщенная схема алгоритма программ логического моделирования. Обобщенная схема алгоритма приведена на рис. 3.5. Она отражает основные операции, которые выполняются в программах логического моделирования. Рассмотрим назначение основных блоков и отметим некоторые особенности.

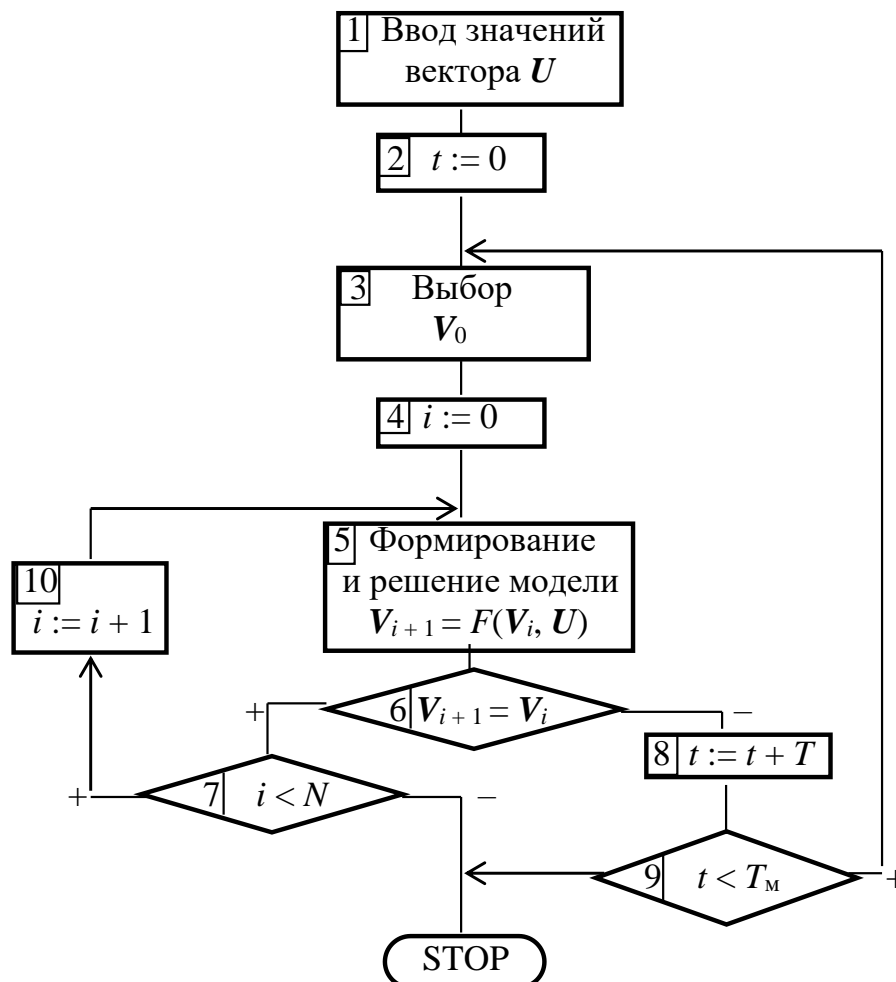


Рис. 3.5. Схема алгоритма программ логического моделирования

Блок 1 предназначен для ввода значений входных для схемы сигналов. Эта процедура в программном обеспечении может быть выполнена различными способами: с помощью графических редакторов, когда на экране задается форма колебаний на входах схемы, с помощью имеющихся библиотек моделей генераторов сигналов и с помощью ручного ввода по точкам графика колебания в виде списка. В результате работы данного блока вектор значений входных шин U считается известным.

Блок 2 – организация цикла по времени. Моделирование может начинаться с нулевого момента времени или с момента, заданного пользователем.

Блок 3 предназначен для задания начальных значений вектора V . Логические уравнения решаются итерационными методами, которые требуют обязательного задания начального приближения. В качестве начального приближения V_0 обычно берутся значения 0 или 1, которые задаются в программе по умолчанию или изменяются пользователем.

Блок 4 – организация цикла по итерациям, где i – номер итерации при решении логических уравнений.

Блок 5 – формирование и решение уравнений модели схемы. Здесь выполняется обращение к библиотекам элементов, формируются уравнения модели и реализуется один из алгоритмов решения уравнений.

Блок 6 – проверка сходимости итерационного процесса. Если равенство значений на данной итерации (V_{i+1}) и предыдущей (V_i) не достигнуто, то итерационный процесс продолжается (переход к блоку 7), иначе – переход к блоку 8.

Блок 7 служит для ограничения итераций в случае расходимости процесса. Если число итераций достигнет значения N , то процесс моделирования останавливается. Такая ситуация возможна при ошибках в схеме. Число N зависит от метода решения уравнений.

Блок 8 осуществляет изменение модельного времени на величину такта моделирования T .

В блоке 9 выполняется проверка окончания моделирования. Если модельное время t достигнет заданного времени окончания моделирования T_m , то процесс моделирования заканчивается, иначе – управление передается блоку 3.

Блок 10 осуществляет изменение номера итераций.

Следует отметить, что данный алгоритм включает в себя как

асинхронное моделирование, так и синхронное. Синхронная модель является частным случаем асинхронной модели. На каждой временной точке решается статическая задача анализа синхронной модели. Поэтому если требуется выполнить только анализ синхронной модели, то цикл по времени не организуется и блоки 2, 8 и 9 не используются.

Методы решения логических уравнений. Общий объем вычислений при логическом моделировании определяется количеством временных тактов (шагов), количеством решаемых уравнений и используемым методом решения уравнений. При моделировании современных сложнейших электронных схем, содержащих сотни тысяч логических элементов, этот объем может оказаться чрезмерно большим. Поэтому важно выбрать наиболее эффективные подходы к организации процесса моделирования и методы решения логических уравнений [20].

Как уже отмечалось, анализ синхронных моделей есть решение системы логических уравнений (3.2), анализ асинхронных моделей также сводится к решению уравнений (3.2) на каждом такте моделирования. Для решения систем логических уравнений применяют итерационные методы: простой итерации, Зейделя без ранжирования и с ранжированием. Рассмотрим эти методы и сравним их между собой.

Метод простой итерации заключается в выполнении итераций по формуле

$$V_{i+1} = F(V_i, U),$$

где V_i – значение вектора V на i -й итерации.

При известном векторе U и принятом начальном приближении V_0 новое значение получается путем прямой подстановки:

$$V_1 = F(V_0, U).$$

Далее, беря результат в качестве исходного, получаем

$$V_2 = F(V_1, U).$$

Этот процесс продолжается до тех пор, пока не будет достигнуто равенство $V_{i+1} = V_i$. Метод отличается простотой реализации в программном обеспечении (необходимо лишь организовать внешний цикл, дополнительно хранить один массив).

Основные недостатки – высокие вычислительные затраты, связанные с медленной сходимостью и необходимостью обращаться ко всем моделям элементов при решении уравнений.

Пример. Сравним методы на несложном примере схемы (рис. 3.6), но результаты сравнения попытаемся распространить на более общий случай. Этой схеме соответствует следующая система уравнений:

$$\left. \begin{array}{ll} 1. e = a \wedge c, & 2. g = \overline{e \vee f}, \\ 3. p = \overline{g \wedge q}, & 4. f = b \wedge c, \\ 5. h = \overline{g \wedge d \wedge c}, & 6. q = \overline{p \wedge h}. \end{array} \right\} \quad (3.4)$$

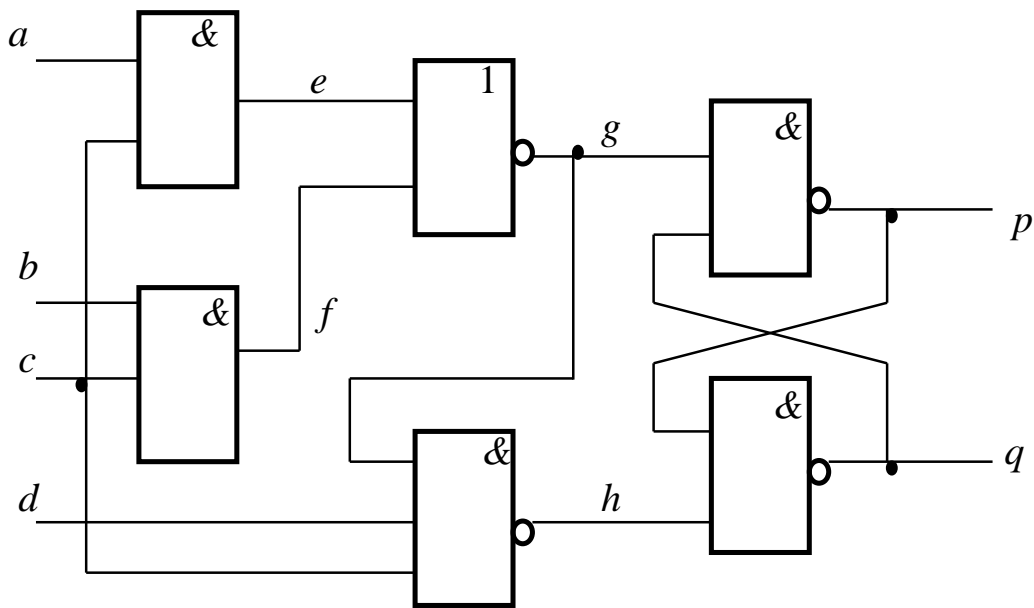


Рис. 3.6. Схема для сравнения методов анализа логических схем

Пусть вектор входных значений $U = (a, b, c, d) = (0110)$. Пусть также для вектора $V = (e, g, p, f, h, q)$ известно начальное приближение, записанное в первой строке табл. 7. Вычислительный процесс по методу простой итерации отражен в последующих строках табл. 7, в которой каждая строка – состояние вектора V на каждой итерации, получаемое в результате вычислений по соответствующим уравнениям модели при значениях аргументов, полученных на предыдущей итерации. Из табл. 7 видно, что потребовалось шесть обращений к каждому уравнению на шести итерациях. Всего 36 обращений.

Таблица 7

Результаты исследования методов решения логических уравнений

Метод	Номер итерации	Начальное приближение	Значения вектора V						Трудо- емкость метода	Изменя- ющиеся переменные (уравнения)
			e	g	p	f	h	q		
			0	0	1	1	1	0		
Простой итерации	1		0	0	1	0	1	0	36	—
	2		0	1	1	0	1	0		
	3		0	1	1	0	0	0		
	4		0	1	1	0	0	1		
	5		0	1	0	0	0	1		
	6		0	1	0	0	0	1		
Зейделя (без ранжирования)	1		0	0	1	0	1	0	24	—
	2		0	1	1	0	0	1		
	3		0	1	0	0	0	1		
	4		0	1	0	0	0	1		
Зейделя (с ранжированием)	1		0	1	0	0	0	1	12	—
	2		0	1	0	0	0	1		
Событийный с простой итерацией	0		0	0	1	1	1	0	7	$b, d(4,5)$ $f(2)$ $g(5)$ $h(6)$ $q(3)$ $p(6)$
	1					0	1			
	2			1						
	3						0			
	4							1		
	5				0					
	6							1		

Ранее уже была отмечена трудоемкость логического моделирования. Реальные схемы могут содержать сотни тысяч логических элементов. Общие затраты определяются соотношением $m \cdot r$, где m – количество тактов моделирования во времени; r – количество обращений к моделям, определяемое произведением числа итераций n на число элементов в схеме g . Если взять средние величины для типовых расчетов схем ($m = 1000$, $g = 10\,000$, $n = 100$ (для метода простой итерации число n может достигать 200 – 300 итераций)), то трудоемкость метода простых итераций составит $1000 \cdot 10\,000 \cdot 100 = 10^9$ обращений к модели.

Метод Зейделя. Метод позволяет уменьшить объем вычислений. Отличие метода Зейделя от метода простой итерации заключается в том, что при вычислении очередного элемента вектора V_i в правую часть следует подставлять не предыдущие значения вектора V_{i-1} , а вновь вычисленные на этой итерации значения V_i , где это возможно. Например, при вычислении по уравнению 3 из системы (3.4) в методе простой итерации используется значение g от предыдущей итерации, а в методе Зейделя – значение g , вычисленное по уравнению 2 на данной итерации. Обобщенная формула для метода Зейделя может быть представлена в следующем виде:

$$V_k^{i+1} = f_k(V_1^{i+1}, \dots, V_{k-1}^{i+1}, V_k^i, \dots, V_N^i),$$

где верхний индекс обозначает номер итерации, а нижний – номер решаемого уравнения; N – число уравнений в модели.

Количество итераций в методе Зейделя существенно зависит от порядка, в котором решаются уравнения модели. Для произвольного порядка записи уравнений (метод называется *методом Зейделя без ранжирования*) пример (3.4) даст результаты, которые приведены в табл. 7. В данном подходе потребовалось уже 4 итерации, трудоемкость метода – 24 обращения к моделям элементов.

Метод Зейделя с ранжированием. В предыдущем случае уравнения были расположены в произвольном порядке, но интуитивно можно предположить, что сходимость будет лучше, если уравнения будут расположены так, что решение предыдущего будет необходимо для всех последующих, другими словами, надо упорядочить уравнения. В технике это называется ранжированием. Физический смысл ранжирования (упорядочивания) – расположение элементов в порядке прохождения сигналов. Тогда для схем без обратных связей потребуется всего лишь одна итерация. В схемах с обратными связями необходимо несколько больше итераций, но все равно их количество будет значительно меньше, чем в методе простой итерации. Очевидно, что метод Зейделя без ранжирования по трудоемкости будет занимать промежуточное положение между методом простой итерации и методом Зейделя с ранжированием, так как зависит от удачного расположения уравнений модели.

Алгоритм ранжирования. Уравнение модели (элемент схемы) получает ранг j , если все аргументы этого уравнения (все входы элемента) ранжированы и максимальный среди рангов аргумент (вход) равен $(j - 1)$. Переменная модели (шина схемы) получает ранг j , если она является левой частью уравнения (выходом элемента), имеющего ранг j . Исполнение алгоритма начинается с присвоения всем входным переменным ранга $j = 0$. Затем определяются уравнения первого ранга, переменные первого ранга, элементы второго ранга и т. д. В итоге уравнения располагаются в порядке возрастания рангов. В случае если в схеме есть обратные связи, то одна из цепей каждого контура должна быть предварительно разорвана и части разорванных цепей, подключенные к входам элементов, получают ранг $j = 0$. Далее действует ранее описанный алгоритм.

В нашем примере ранжирование начинается с присвоения ранга 0 входным переменным a, b, c и d . Далее просматриваются уравнения (3.4) и устанавливается, что в уравнениях 1 и 4 все аргументы ранжированы и имеют ранг 0. Тогда уравнения 1 и 4 получают ранг 1 и так далее по алгоритму. После ранжирования получим следующую последовательность уравнений:

$$\left. \begin{array}{l} 1(1). e = a \wedge c, \quad 2(4). f = b \wedge c, \\ 3(2). g = \overline{e \vee f}, \quad 4(5). h = \overline{g \wedge d \wedge c}, \\ 5(6). q = \overline{p \wedge h}, \quad 6(3). p = \overline{g \wedge q}. \end{array} \right\} \quad (3.5)$$

Результаты решения методом Зейделя с ранжированием приведены в табл. 7. Решение получается за одну итерацию, вторая служит для проверки сходимости. Всего получилось 12 обращений к модели. Обобщение на случай схемы средних размерностей дает трудоемкость метода Зейделя с ранжированием: $m \cdot g \cdot n = 1000 \cdot 10\,000 \cdot 1 = 10^7$ обращений к модели (при $m = 1000, g = 10\,000, n = 1$). Выигрыш по сравнению с методом простой итерации составляет примерно два порядка.

Событийный метод. Дальнейшее сокращение вычислительных затрат достигается с помощью событийного метода. Если метод Зейделя по сравнению с методом простой итерации сокращает число итераций (по сути – число строк в табл. 7), то событийный метод сокращает трудоемкость каждой итерации (по сути – число столбцов в табл. 7), что

особенно важно для современных схем (число столбцов определяется размерностью схемы).

Основная идея событийного метода заключается в выполнении вычислений по уравнениям только активизированных элементов, т. е. элементов, у которых хотя бы на одном входе произошло событие (что-то изменилось). В алгоритмах событийного метода на каждом очередном шаге устанавливается, какие элементы являются активизированными, и выполняется обращение только к этим элементам. Обычно в цифровых устройствах только до 10 % элементов схемы меняют свое состояние на такте моделирования, поэтому событийный алгоритм позволяет дать значительный выигрыш по времени расчета. Но надо отметить, что программное обеспечение несколько усложняется. Событийный метод может быть использован в комбинации с любым итерационным методом.

Для примера, описанного на с. 67 учтем, что входной вектор $U = (a, b, c, d) = (0110)$ на очередном такте моделирования будет менять свое значение на (0011) . Таким образом, входной сигнал меняется на шинах b и d (с 1 на 0 и с 0 на 1). Это активизирует элементы 4 и 5, они изменяют f и активизируют 2 и т. д. В итоге получим семь обращений к модели (см. результаты в табл. 7). Без событийного алгоритма, напомним, было 36 обращений к моделям. Для случая схемы средних размерностей трудоемкость событийного метода в комбинации с методом Зейделя с ранжированием даст $m \cdot g \cdot n = 1000 \cdot 10\,000 \cdot 0,1 \cdot 1 = 10^6$ обращений к модели (при $m = 1000$, $g = 10\,000$, $n = 1$ и 10 % активизированных элементов).

3.4. Моделирование аналоговых функциональных схем

Как было отмечено, существует два фундаментальных подхода к моделированию аналоговых функциональных схем. Первый подход основан на использовании техники макро моделирования, т. е. представлении схем функциональных блоков (ФБ) в виде макромоделей. Это упрощенные описания моделей ФБ в виде эквивалентных схем замещения на основе электрического описания объекта, где далее используются все приемы и подходы схемотехнического моделирования.

Второй подход связан с методами теории автоматического управления, где отдельные ФБ используются в качестве функциональных

преобразователей, выполняющих определенную операцию над входным сигналом, используя теорию автоматического регулирования. Если первый подход должен рассматриваться в рамках теории схемотехнического моделирования (гл. 4), то второй подход мы рассмотрим далее.

Задачи и особенности моделирования. Основные задачи при функциональном моделировании – отработка структуры схемы (состава и связей функциональных элементов), исследование процессов преобразования сигналов при прохождении через тракты обработки и преобразования информации.

Основная особенность функционального моделирования состоит в том, что в качестве переменных используются переменные типа сигнал, не отождествляемые с видом сигнала (ток или напряжение) и передаваемой информацией. При моделировании на функциональном уровне ставится задача моделирования с очень высокой скоростью, так как необходимо исследовать множество вариантов с целью отбора самого лучшего.

Функциональная схема (ФС) представляется в виде структуры, где каждый элемент выполняет какое-либо функциональное преобразование над сигналом (усиление, фильтрацию, ограничение, кодирование, декодирование, задержку и т. п.).

При моделировании решается задача расчета формы сигнала или его параметров в различных точках функциональной схемы. Под формой сигнала (waveform) понимается зависимость сигнала от времени $x(t)$ или эквивалентное представление сигнала по Лапласу в области комплексной переменной p – $x(p)$, или зависимость от частоты $x(j\omega)$ в частотной области.

При моделировании функциональных схем учитываются следующие допущения [10]:

- развязка отдельных ФБ, т. е. независимость характеристик ФБ от режимов работы других ФБ. Это эквивалентно условию, что у ФБ входное сопротивление $R_{вх} = \infty$, а выходное $R_{вых} = 0$. Таким образом, описывается характеристика вход/выход ФБ и не учитывается их взаимное влияние;

- однонаправленность элементов, т. е. сигнал на выходе ФБ не влияет на сигнал на его входе. Это означает передачу сигнала только в

одном направлении от входа к выходу (нет внутренних обратных связей).

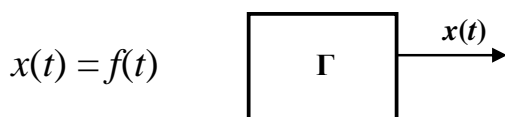
Функциональное моделирование применяют для исследований:

- систем автоматического регулирования, где функциональная схема представляется набором типовых звеньев (дифференцирующего, интегрирующего, задержек, стандартных преобразователей и т. п.);
- временных диаграмм работы аналого-цифровых преобразователей (АЦП) и цифро-аналоговых преобразователей (ЦАП);
- трактов передачи и обработки сигналов (сигнальные процессоры);
- радиотрактов.

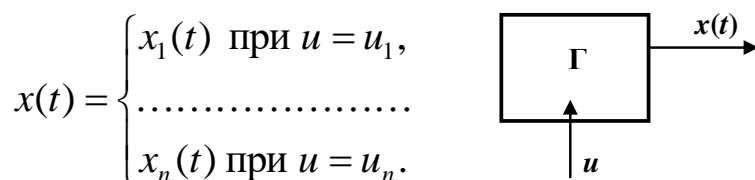
Модели базовых элементов функциональных схем. Несмотря на многообразие ФС, все они строятся из небольшого числа базовых элементов (БЭ), которые можно объединить в четыре группы:

- генераторы сигналов;
- безынерционные элементы;
- линейные инерционные элементы;
- нелинейные инерционные элементы.

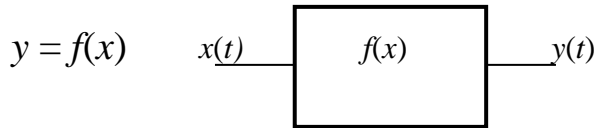
Генераторы сигналов подразделяют на независимые генераторы, генерирующие на выходе сигнал $x(t)$ по заданному закону изменения (в качестве таких законов могут быть использованы различные тригонометрические функции и их комбинации),



и на управляемые генераторы, формирующие ту или иную форму сигнала на выходе $x(t)$ в зависимости от управляющего воздействия u :



Безынерционные элементы могут быть как линейными, так и нелинейными. Функция преобразования таких элементов представляется в виде функции f , связывающей входной сигнал x с выходным u :

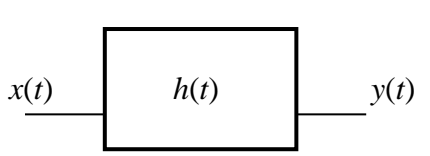


Функция времени в данном выражении отсутствует (формально), поэтому безынерционные элементы широко используются для преобразования формы сигнала, например для ограничения амплитуды входного сигнала, формирования импульсов другой формы и т. п. Важным частным случаем является безынерционный элемент с памятью, например триггер. В этом случае функция преобразования зависит не только от значения x , но и от состояния элемента A :

$$y = f(x, A).$$

Некоторые примеры безынерционных элементов приведены в табл. 8.

Линейный инерционный элемент (линейный элемент с памятью) может характеризоваться следующими функциями преобразования: переходной (импульсной) характеристикой во временной области $h(t)$, коэффициентом передачи в частотной $K(j\omega)$ области или в области комплексной переменной $K(p)$. Связь между входом и выходом будет соответственно описываться соотношениями



$$\begin{aligned}
 y(t) &= \int_0^t x(\tau) h(t - \tau) d\tau; \\
 y(j\omega) &= K(j\omega) x(j\omega); \\
 y(p) &= K(p) x(p).
 \end{aligned}
 \tag{3.6}$$

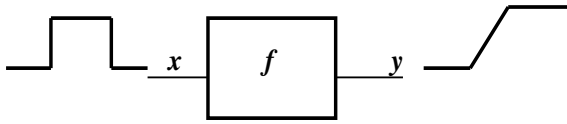
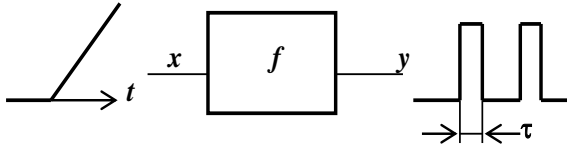
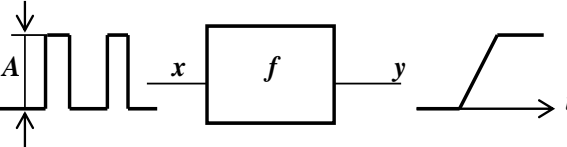
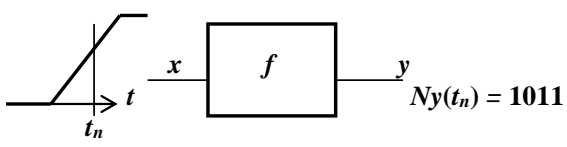
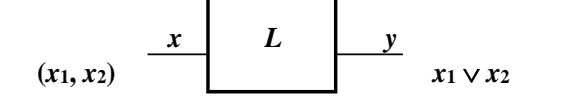
Линейные инерционные элементы могут быть описаны и дифференциальными уравнениями, которые будут рассмотрены ниже. К линейным инерционным элементам относят широкий класс устройств, например различные типы фильтров, усилителей, устройства, выполняющие типовые линейные операции интегрирования

$$y(t) = \int_0^t x(\tau) d\tau$$

и дифференцирования

$$y(t) = dx(t) / dt.$$

Примеры безынерционных элементов

Тип элемента	Выполняемое преобразование	Математическое соотношение
Аналого-аналоговый		$y = f(x)$
Аналого-импульсный		$\tau = f(x(t))$ – широтно-импульсная модуляция
Импульсно-аналоговый		$y(t) = k(A)t$
Аналого-цифровой		$Ny(t) = kx(t)$
Логический		$y = L(x_1, x_2)$, L – логическая функция

Нелинейные инерционные элементы считаются наиболее сложными для моделирования. Функцией преобразования здесь является нелинейный оператор $A(x)$, т. е. каждой реализации $x(t)$ ставится в соответствие своя реализация $y(t)$. Такие элементы и системы часто называют динамическими элементами с памятью.

Наиболее общей формой описания подобных элементов считается дифференциальное уравнение (ДУ)

$$\frac{dy(t)}{dt} + g[y(t)] = \frac{dx(t)}{dt} + f[x(t)],$$

где $x(t)$, $y(t)$ – входные и выходные сигналы элемента.

Форма представления модели в виде ДУ удобна для моделирования в программном обеспечении на ЭВМ, легко вписывается в аппарат численных методов, но часто получить данные о модели в виде ДУ затруднительно, пользователи программ чаще оперируют передаточными

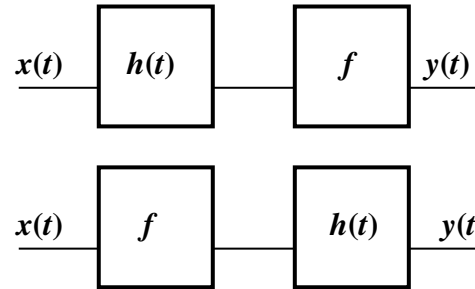
характеристиками, которые легче получить либо теоретически, либо экспериментально путем измерений. Поэтому для описания нелинейных инерционных элементов применяют ряд приближенных методов.

Первый из таких чаще всего применяемых приемов – разделение нелинейных и инерционных свойств объекта и представление его последовательным соединением линейного инерционного и нелинейного безынерционного элементов. В этом случае уравнение модели представляется зависимостью

$$y(t) = f\left(\int_0^t x(\tau)h(t - \tau)d\tau\right)$$

или

$$y(t) = \int_0^t f(x(\tau))h(t - \tau)d\tau$$



Отметим, что выделить нелинейные или инерционные свойства не всегда удастся без потери точности, но на системном (функциональном) уровне часто этого достаточно.

Другой способ приближенного представления нелинейного инерционного элемента, широко распространенный в последние годы, – применение аппарата функциональных рядов Вольтерра. Ряд Вольтерра является обобщением линейных инерционных элементов на случай нелинейных динамических (инерционных) систем и в общем случае представляется в виде

$$y(t) = \sum_{i=1}^n y_i(t),$$

где $y_i(t) = \int \dots \int h_i(\tau_1, \dots, \tau_i) \prod x(t - \tau_i) d\tau_1 \dots d\tau_i$; $h_1(\tau_1)$ – импульсная линейная характеристика (ядро первого порядка); $h_2(\tau_1, \tau_2)$ – ядро второго порядка и т. д. Ряд Вольтерра можно также охарактеризовать как степенной ряд Тейлора, где коэффициенты ряда выступают комплексными величинами. Отсюда и ограничения метода – требуется достаточно большое число членов ряда для описания элементов с сильными нелинейными свойствами. Трудоемкость расчета ядер высоких порядков очень велика, поэтому на практике обычно ограничиваются тремя членами ряда, что значительно сужает область применения данного аппарата для слабо нелинейных систем.

Алгоритмы моделирования базовых элементов. Рассмотрим основные приемы, используемые при реализации на ЭВМ алгоритмов расчета отклика выходной величины y по заданному входному значению x для базовых элементов.

Генераторы сигналов моделируются путем вычисления заданной функции $x(t)$ в известные моменты времени t_n , т. е. непрерывная функция $x(t)$ заменяется дискретной $x_n = x(t_n)$. Для экономии вычислительных затрат применяют рекуррентные соотношения, когда при вычислении новых значений x_n используют известные старые значения x_{n-1} , x_{n-2} ,

Пример. Необходимо смоделировать сигнал $x(t) = ke^{at}$. Используется следующее рекуррентное соотношение:

$$x_n = x(t_n) = ke^{at_n} = ke^{a(t_{n-1} + \Delta t)} = ke^{a\Delta t} e^{at_{n-1}} = e^{a\Delta t} x_{n-1}.$$

Если сигнал моделируется через равные промежутки времени Δt , то, вычислив один раз $A = e^{a\Delta t}$, все последующие значения x можно рассчитать по простой рекуррентной формуле

$$x_n = Ax_{n-1}, \quad x_0 = k.$$

Безынерционные элементы моделируются путем подстановки дискретных значений входных сигналов $x_n = x(t_n)$ в функцию, которая описывает линейное или нелинейное преобразование:

$$y_n = f(x_n).$$

Линейные инерционные элементы моделируются по-разному в зависимости от областей представления сигналов и самих элементов. Самые простые случаи – когда область представления сигнала совпадает с областью представления элемента. Например, в частотной области, или области комплексной переменной, расчет отклика на входное воздействие может быть выполнен по простым соотношениям

$$\begin{aligned} y(p) &= H(p) \cdot x(p), \\ y(j\omega_i) &= H(j\omega_i) \cdot x(j\omega_i), \end{aligned}$$

где ω_i – значение i -й частоты расчета, $\omega_i = 2\pi f_i$.

Во временной области линейные инерционные элементы моделируются на основе соотношений *дискретной свертки* (метод скользящего суммирования). Суть метода состоит в разбиении интервала интегрирования $(0, t)$ для

$$y(t) = \int_0^t x(\tau) h(t - \tau) d\tau$$

и интервала текущего времени интегрирования $(0, \tau)$ на отрезки Δt :
 $t = n\Delta t$, $\tau = k\Delta t$, $t - \tau = (n - k)\Delta t$.

Используя метод прямоугольников при численном интегрировании, получим при $d\tau \approx \Delta t$ вместо интеграла его дискретную аппроксимацию в виде суммы

$$y_n = \sum_{k=0}^n x(k\Delta t) h((n - k)\Delta t) \Delta t = \Delta t \sum_{k=0}^n x_k h_{n-k}. \quad (3.7)$$

Выражение (3.7) называют дискретной сверткой, хотя, применяя другие формулы численного интегрирования, можно получить более точные формулы, и соотношение будет немного отличаться.

Все остальные подходы к моделированию линейных инерционных элементов, когда области представления сигналов и элементов различаются, будут несколько сложнее, так как для преобразования сигналов будет необходимо применять методы Фурье- и Лапласа-преобразования [10].

Нелинейные инерционные элементы моделируются в зависимости от принятого метода моделирования. Если принят метод ДУ, то алгоритм решения задачи моделирования очень похож на решение задач схемотехнического моделирования и будет рассмотрен в гл. 4. Если принят метод разделения нелинейных и инерционных свойств, т. е. представление модели в виде нелинейного безынерционного элемента и линейного инерционного элемента, то алгоритмы будут те же, что были рассмотрены в соответствующих пунктах. Алгоритмы моделирование на основе аппарата рядов Вольтерра очень сложны в изложении и поэтому в пособии не рассматриваются. Читателю можно порекомендовать работу [21].

Построение и моделирование функциональных схем. Построение модели сложных функциональных схем обычно выполняется в два этапа. На первом этапе каждый реальный элемент устройства (системы) представляется соответствующей моделью базового элемента. Если выполняемые функции ФС не очень сложны, то на этом построение модели ФС может быть закончено. Если же выполняемые функции элементов ФС сложны, то обычно необходим второй этап – этап детализации модели ФС. Этап детализации, в свою очередь, может быть

также многоступенчатым, когда детализируемый элемент разделяется на более мелкие, дополнительно характеризующие его преобразования.

Пример. Рассмотрим на примере схемы приемника методику построения модели ФС. Пусть структурная схема имеет вид, изображенный на рис. 3.7, а. После первого этапа, предположив, что усилитель высокой частоты (УВЧ), усилитель промежуточной частоты (УПЧ) и усилитель низкой частоты (УНЧ) выполняют чисто линейную операцию (линейные инерционные элементы), идеальный смеситель выполняет операцию вычитания сигналов, а детектор – идеальное нелинейное безынерционное преобразование, получим модель ФС, изображенную на рис. 3.7, б. Если первое приближение не дает при моделировании удовлетворительной точности, то отдельные наиболее сложные элементы можно детализировать. Так, например, дальнейшая детализация смесителя приведет к более сложной модели (рис. 3.8), где вместо идеальной операции сложения учитываются нелинейные (блок нелинейного безынерционного элемента) и инерционные свойства смесителя.

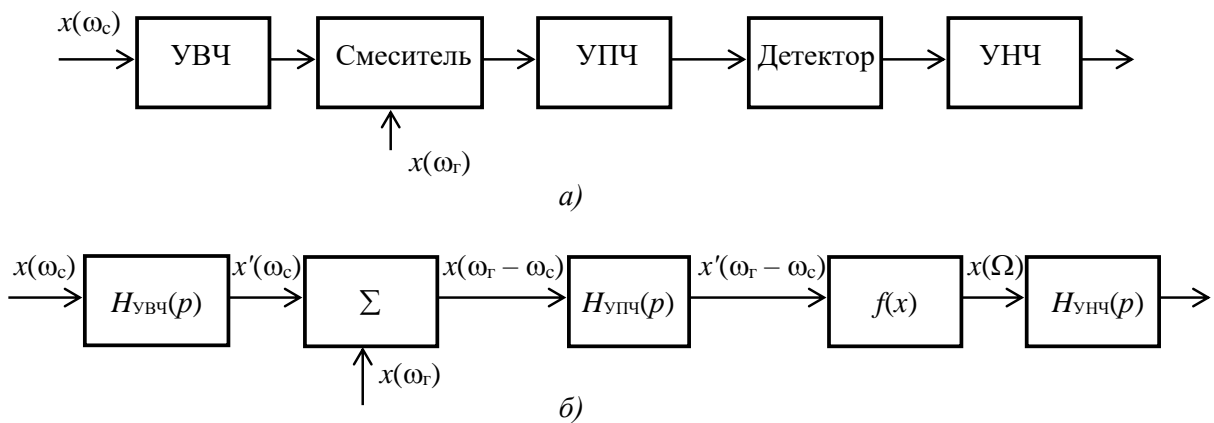


Рис. 3.7. Схема приемника: а – структурная; б – в виде функциональной модели

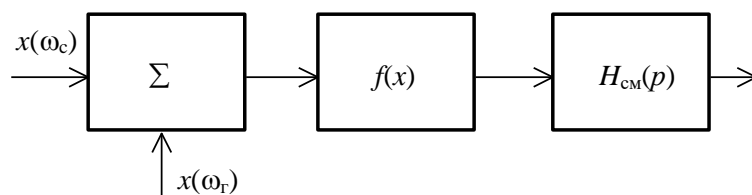


Рис. 3.8. Функциональная модель смесителя

Аналогично могут быть детализированы блоки усилителей (необходимо учесть нелинейные свойства) и детектора (следует учесть инерционные свойства).

Моделирование полученной модели ФС заключается в последовательном преобразовании сигнала каждым элементом с помощью алгоритмов, которые были рассмотрены ранее. Такое моделирование называется *имитационным*, или *причинно-следственным*. Моделирование ФС при воздействии сложных модулированных сигналов может быть выполнено на основе метода несущей или огибающей сигнала.

Пример. Необходимо выполнить моделирование схемы при воздействии амплитудно-модулированного сигнала, изображенного на рис. 3.9.

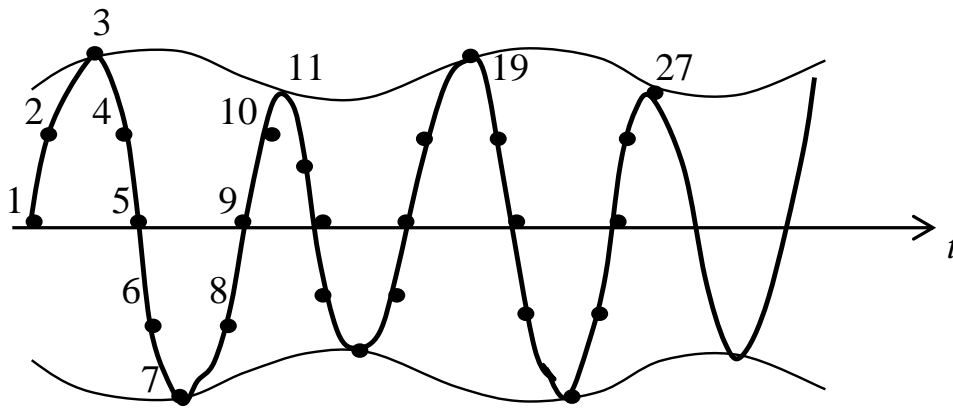


Рис. 3.9. Амплитудно-модулированный сигнал

При моделировании ФС чаще всего интересует процесс преобразования информации, которая заключается в огибающей исходного (несущего) сигнала. Если выполнять моделирование сигнала (см. рис. 3.9) на основе исходных уравнений (по несущей частоте) в точках 1, 2, 3, 4, ..., то это может привести к значительным затратам времени. Если же выполнять моделирование только в точках 3, 11, 19, ..., т. е. по огибающей сигнала, то это может значительно сократить вычислительные затраты. Если круговая частота исходного (несущая) сигнала $\omega_0 = 10^6$ Гц, а частота модуляции (огибающая) $\omega_M = 10^3$ Гц и для достоверного представления сигнала берутся хотя бы десять точек на периоде $N = 10$, то для моделирования одного периода модуляционного колебания потребуется $(\omega_0/\omega_M) N = (10^6/10^3)10 = 10^4$. Это очень много.

В то же время моделирование только огибающей сигнала потребует $N = 10$ расчетов. Выигрыш по времени в этом случае составит три порядка.

Но моделирование на основе огибающей требует преобразования исходных моделей ФБ, которые в методе огибающей должны отражать преобразования огибающей сигнала (их можно найти в работах [10; 22]).

Вопросы для самопроверки

1. Задачи и особенности функционально-логического моделирования.
2. Особенности моделирования цифровых схем.
3. Общее выражение для представления математической модели элементов цифровых схем на логическом уровне.
4. Уравнения синхронной и асинхронной моделей схемы.
5. Моделирование рисков сбоя.
6. Сравнение итерационных методов решения системы логических уравнений.
7. Алгоритм ранжирования переменных в методе Зейделя.
8. Задачи и особенности моделирования аналоговых функциональных схем.
9. Модели базовых элементов функционального моделирования.
10. Алгоритмы моделирования базовых элементов.
11. Подходы к моделированию функциональных схем.

Глава 4. АВТОМАТИЗАЦИЯ СХЕМОТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Вопросам схемотехнического моделирования и проектирования посвящено большое число учебников, пособий и монографий [1 – 11; 23 – 43], поэтому в данной главе отметим лишь основные положения и рассмотрим некоторые современные тенденции схемотехнического моделирования.

4.1. Задачи схемотехнического проектирования

Основная задача схемотехнического проектирования – разработка принципиальных электрических схем устройств. Основной особенностью схемотехнического уровня проектирования является исследование процессов в элементах схемы путем расчёта токов и напряжений в заданных точках схемы [2]. При этом каждый элемент схемы должен быть представлен своей моделью. Элементами могут служить дискретные компоненты: резисторы, конденсаторы, индуктивности, диоды, транзисторы, трансформаторы и т. п., интегральные микросхемы, отдельные участки полупроводникового кристалла, фрагменты БИС.

Исходными данными для схемотехнического проектирования выступают результаты функционального и функционально-логического проектирования (структурная или функциональная схема, требования к выходным параметрам и к исполнению). Основная цель автоматизации схемотехнического проектирования (АСхП) состоит в автоматическом формировании математической модели электронной схемы по исходному ее описанию и в определении формы и параметров токов и напряжений в различных точках схемы. Основная задача АСхП – анализ. Задачи параметрической оптимизации хорошо формализованы и имеют развитые программные средства, но ограничиваются либо малой размерностью последних (малой размерностью схемы), либо большими вычислительными затратами.

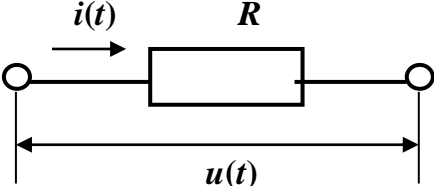
4.2. Математические модели элементов

При схемотехническом проектировании математической моделью называют математическое описание связей между токами и напряжениями в различных режимах [3]. В частности, математическими мо-

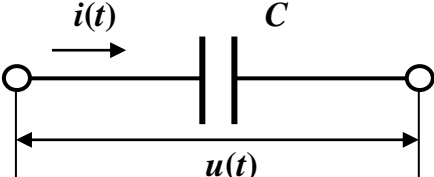
делями элементов (ММЭ) могут быть уравнения вольт-амперных характеристик или дифференциальные уравнения, математическими моделями схем (ММС) – системы уравнений, описывающие различные режимы работы схем. Математические модели электронных схем – транзисторов, диодов, резисторов, конденсаторов, катушек индуктивностей, источников питания и внешних сигналов – в значительной степени определяют эффективность использования САПР [10]. От них зависят как адекватность результатов моделирования, так и затраты машинного времени. Поэтому задача выбора, а в ряде случаев и корректировки моделей, должна решаться каждый раз при постановке задачи моделирования. В настоящее время практически все системы АСхП содержат встроенные библиотеки ММЭ электронных схем, включая модели типовых элементов. Такими моделями можно пользоваться, автоматически включая их в общую ММС. Однако при этом необходимо представлять особенности и ограничения используемых моделей.

Математические модели элементов. Для ММЭ (иногда используют определение «математическая модель компонента», ММК) уравнения должны отражать связь между внутренними параметрами элемента (электрическими или конструктивно-технологическими) и внешними, в качестве которых служат токи и напряжения. При построении ММЭ используется небольшое число базовых компонент, которые отражают все основные свойства электрических цепей и из которых строятся модели типовых элементов схем. В качестве таких базовых компонент используют, например:

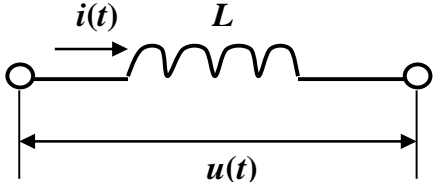
– модель резистора R , которая учитывает потери (сопротивление) и описывается известным соотношением

$$i(t) = u(t)/R$$

(4.1)

– модель конденсатора C , учитывающую емкость (накопление электрической энергии)

$$i(t) = C \cdot du(t)/dt$$

(4.2)

– модель катушки индуктивности L , учитывающую индуктивность (накопление магнитной энергии)

$$u(t) = L \cdot di(t)/dt$$

(4.3)

Эти базовые элементы отражают только свойства цепей, модель же элементов представляется в виде эквивалентных схем замещения, где используются базовые элементы. Например, простейшей моделью типового схемного элемента – резистора (идеального) – будет соотношение (4.1), но более полная модель резистора, учитывающая потери (сопротивление), индуктивности выводов и емкость корпуса, будет уже представляться эквивалентной схемой, содержащей базовые элементы, отражающие описываемые свойства (рис. 4.1).

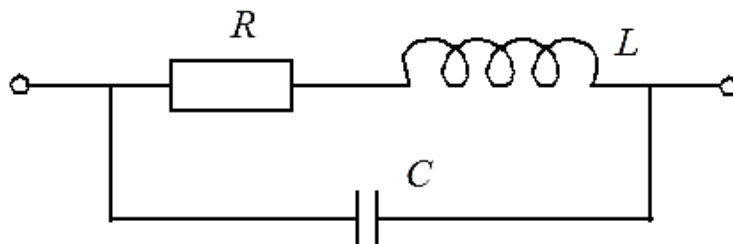


Рис. 4.1. Модель резистора

Кроме трех базовых элементов (4.1 – 4.3) при построении моделей типовых схемных элементов применяют модели:

- нелинейного двухполюсного резистивного элемента, в общем виде записываемого в виде различных аппроксимаций: $i(t) = f[u(t), x]$, где x – вектор параметров аппроксимации;
- нелинейных энергоемких элементов (C, L);
- различные виды управляемых источников;
- взаимной индуктивности [8; 32].

ММЭ более сложных схемных элементов – активных приборов (диодов, транзисторов), типовых интегральных микросхем – обычно представляются в виде сложных эквивалентных схем, содержащих простейшие базовые компоненты и нелинейные зависимости. Модели компонентов могут учитывать воздействия внешних условий, таких как температура, давление, влажность, радиация, эффекты старения.

Модели полупроводниковых диодов. Полупроводниковые диоды применяют как в дискретных, так и в интегральных цифровых и аналоговых схемах. В интегральных схемах (ИС) в качестве диодов обычно используют транзисторы в одном из возможных диодных включений. Эквивалентная схема модели диода [8] показана на рис. 4.2, а. Она содержит управляемый источник тока I_d , сопротивление диода R_d , складывающееся из сопротивлений областей базы диода, омических контактов и выводов и емкости диода C_d .

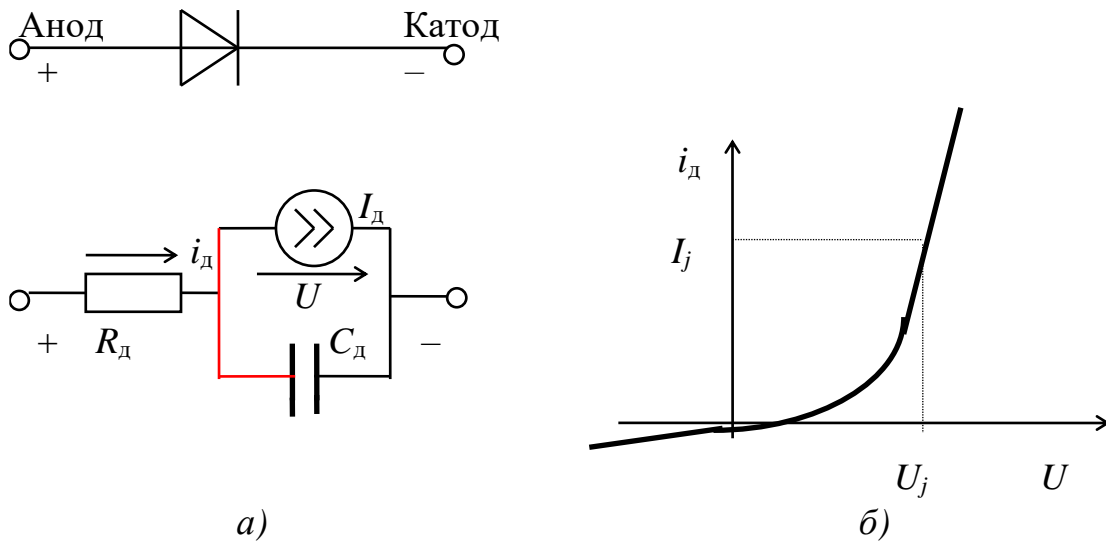


Рис. 4.2. Модель диода: а – эквивалентная схема; б – зависимость тока от напряжения

Наиболее распространена нелинейная модель диода, которая базируется на уравнениях Эберса – Молла для управляемого источника тока и учитывает зависимости емкостей от режима:

$$I_d = I_0 \left[\exp\left(\frac{U}{mV_T}\right) - 1 \right],$$

$$C_d = C_{бар} + C_{диф},$$

$$C_{бар} = C_0 (\phi - U)^{-1/2},$$

$$C_{диф} = \frac{\partial I_d}{\partial U} \tau = \frac{\tau}{mV_T} \left[I_0 \exp\left(\frac{U}{mV_T}\right) \right].$$

где I_0 – тепловой ток; U – напряжение на $p-n$ переходе; m – коэффициент не идеальности вольт-амперной характеристики $p-n$ перехода;

V_T – тепловой потенциал перехода, $V_T \frac{kT}{q}$, где k – постоянная Больцмана, T – температура окружающей среды, q – заряд электрона; ϕ – контактная разность потенциалов; C_d , $C_{бар}$, $C_{диф}$ – суммарная, барьерная и диффузионная емкости соответственно; τ – постоянная времени, учитывает предельную частоту работы диода.

Параметры модели диода определяются из физических соображений: для нормальной температуры $V_T = 0,026$ В, $\phi = 0,5...0,7$ В. Параметры I_0 , m , R_d вычисляются из условия аппроксимации статической характеристики диода (рис. 4.2, б) с помощью выражения

$$I_j = I_0 \left(\exp \frac{U_j - I_0 R_d}{m V_T} \right), \quad j = 1, 2, \dots, N,$$

где I_j , U_j – соответствующие j -й экспериментальной точке характеристики диода значения тока и напряжения на диоде; N – число известных экспериментальных точек.

Барьерная емкость $C_{бар}$ моделирует приращение пространственного заряда при изменении напряжения на p - n переходе. Диффузионная емкость $C_{диф}$ отражает влияние перераспределения подвижных носителей. Следует помнить, что диффузионная емкость является доминирующей при прямом смещении, а барьерная – при обратном.

Модели биполярных транзисторов. Промышленные предприятия выпускают большое число разнообразных типов транзисторов различного назначения. Кроме того, транзисторы – один из основных элементов ИС, причем их относительное число в схемах непрерывно растет. Следует отметить, что практически все программы АСхП имеют встроенные модели биполярных транзисторов.

В качестве нелинейной модели биполярного транзистора наибольшее распространение получили различные модификации модели Эберса – Молла, которая была предложена для дискретного транзистора в 1954 году, т. е. в самом начале развития транзисторной техники. Рассмотрим один из вариантов такой модели, в основе которой лежит представление транзистора как системы двух взаимодействующих диодов [7; 44]. Эквивалентная схема модели приведена на рис. 4.3. В ней используются следующие обозначения: $I_э$, $I_к$ – источники внутренних токов, управляемые напряжениями на p - n переходах; $r_э$, $r_б$, $r_к$ – сопротивления объемных областей эмиттера, базы и коллектора соответственно; $C_э$, $C_к$ – емкости эмиттерного и коллекторного переходов.

Для расчетов токов $I_э$ ($U_э, U_к$) и $I_к$ ($U_э, U_к$) используются модифицированные уравнения Эберса – Молла:

$$I_э = I_{э0} \left(\exp \frac{U_э}{mV_T} - 1 \right) - \frac{B_I}{B_I + 1} I_{к0} \left(\exp \frac{U_к}{mV_T} - 1 \right),$$

$$I_к = I_{к0} \left(\exp \frac{U_к}{mV_T} - 1 \right) - \frac{B_N}{B_N + 1} I_{э0} \left(\exp \frac{U_э}{mV_T} - 1 \right),$$

где $U_э$ и $U_к$ – напряжения на эмиттерном и коллекторном переходах; B_N и B_I – статические коэффициенты передачи по току в схеме с общим эмиттером; $I_{к0}$ и $I_{э0}$ – обратные токи переходов. В ряде случаев в модель вводят также зависимость коэффициентов передачи от напряжения на переходах, задающую наклон коллекторных характеристик транзистора. Для расчета емкостей модели используют соотношения, аналогичные модели диода (см. рис. 4.2).

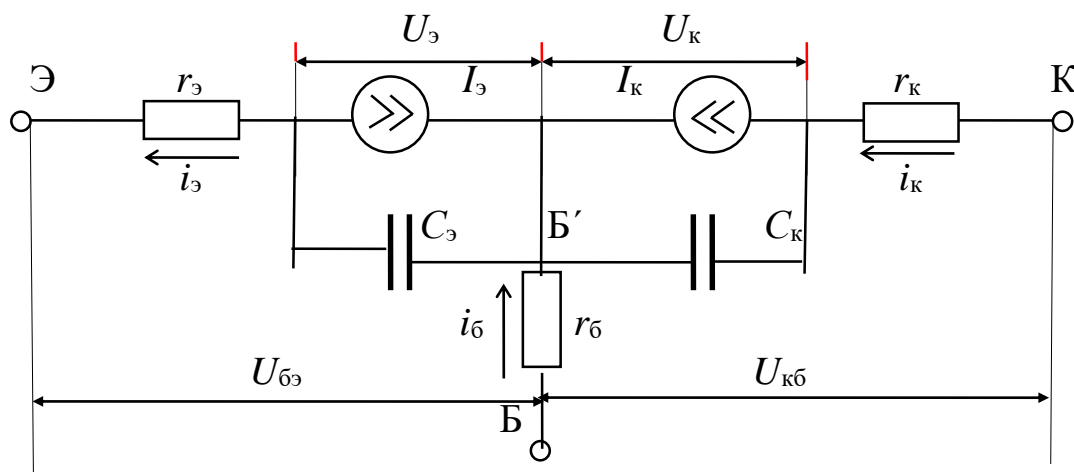


Рис. 4.3. Модель Эберса – Молла для биполярного транзистора

Данная нелинейная модель транзистора, которая называется также *инжекционной моделью*, имеет 24 параметра. Она может быть модифицирована как в сторону упрощения, так и в сторону усложнения. Например, можно исключить зависимости параметров от режима, считать емкости постоянными, а в ряде схем вообще пренебречь ими при анализе на низких частотах, для маломощных транзисторов не учитывать сопротивления эмиттера и коллектора и т. п. При моделировании высокочастотных схем в модель вносят индуктивности выводов, для повышения точности необходимо учитывать зависимость сопротивления базы от режима, частотные зависимости коэффициентов передачи и др.

Другой часто используемый [45] в программах АСхП вариант модели биполярного транзистора известен как *передаточная Т-образная* (рис. 4.4, а) и *П-образная* (рис. 4.4, б) модель, или как *модель Гуммеля – Пуна*.

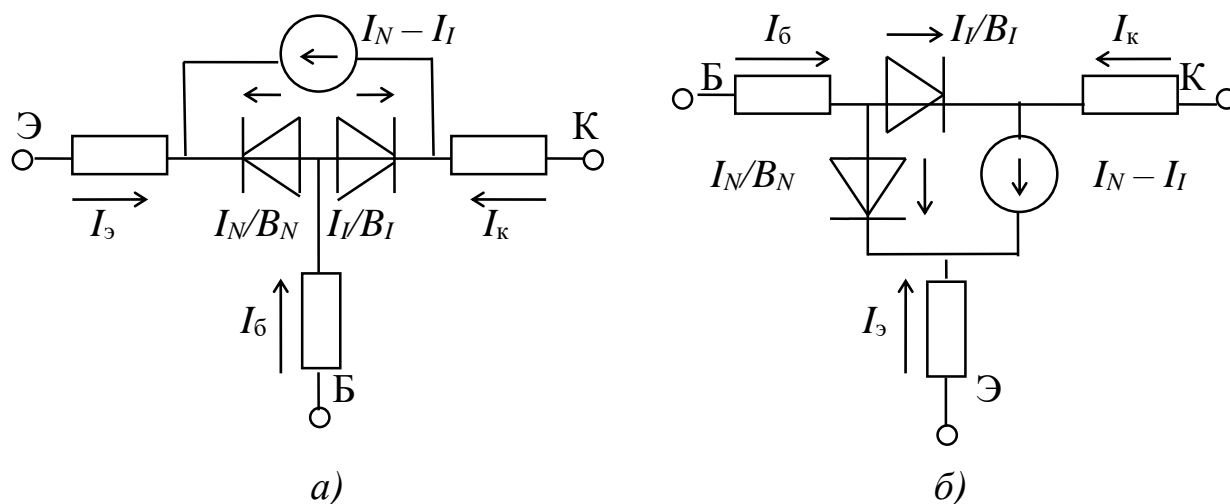


Рис. 4.4. Модель биполярного транзистора: а – Т-образная; б – П-образная

Если в инжекционной модели в качестве токов управляемых источников приняты токи, инжектируемые *p-n* переходами, то в передаточной модели приняты токи, собираемые *p-n* переходами:

$$I_N = I_S[\exp(U_э/V_T) - 1];$$

$$I_I = I_S[\exp(U_к/V_T) - 1],$$

где I_S – обратный ток перехода, называемый током *отсечки*. Этот параметр определяется по характеристикам транзистора. Токи в цепи выводов базы, эмиттера и коллектора имеют вид

$$I_6 = I_N/B_N + I_I/B_I;$$

$$I_э = I_I - (1 + 1/B_N)I_N;$$

$$I_к = I_N - (1 + 1/B_I)I_I.$$

Зависимости емкостей от напряжения в передаточной модели описываются так же, как и в инжекционной модели. Для диффузионных емкостей модели Гуммеля – Пуна используется следующее представление:

$$C_{э. \text{ диф}} = (\tau_N/V_T) (I_N + I_S), \quad C_{к. \text{ диф}} = (\tau_I/V_T) (I_I + I_S),$$

где τ_N, τ_I – постоянные времени, характеризующие инерционность процессов передачи носителей заряда от одного перехода к другому. Экспериментальные исследования показали, что эти постоянные времени

слабее зависят от режима, чем соответствующие параметры инжекционной модели, а также имеют меньший технологический разброс. Однако в целом точность обеих моделей одного порядка определяется используемыми параметрами.

Модели МДП-транзисторов. Эти приборы широко распространены, в особенности в качестве элементов ИС и БИС. МДП-транзисторы относятся к униполярным приборам, модели которых проще, чем биполярных транзисторов. Основным элементом моделей выступает источник тока канала, зависимость которого от напряжений на выводах характеризует активное действие униполярного транзистора, а различные аппроксимации этой зависимости являются основой разных видов моделей.

Одна из моделей МДП-транзистора для режима большого сигнала показана на рис. 4.5.

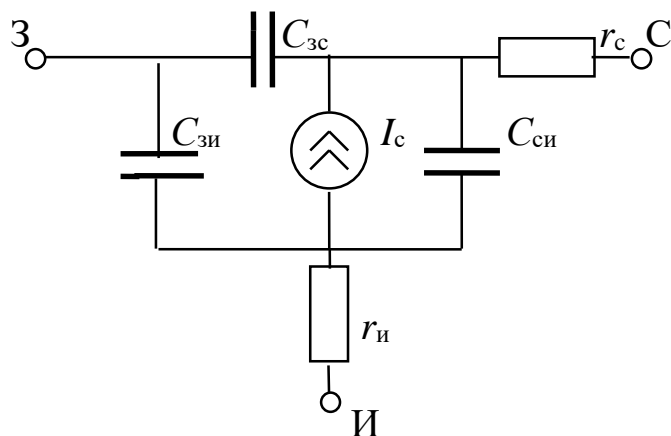


Рис. 4.5. Модель МДП-транзистора

Основной элемент модели – источник тока I_c , зависимость которого от напряжений на выводах и определяет основные характеристики работы модели. При напряжениях на затворе, меньших порогового напряжения ($U_{зи} \leq U_0$), ток стока $I_c = 0$. При напряжениях на затворе, больших U_0 , ток стока описывается выражением

$$I_c = S(U_{зи} - U_0)^2 \left(1 - \exp\left(-\frac{3U_{ск}}{V_{зи} - U_0}\right) \right) + \lambda(U_{зи} - U_0)U_{си},$$

где S – удельная крутизна сток-затворной характеристики; λ – параметр модуляции длины канала. Емкости затвор – сток $C_{зс}$, затвор – исток $C_{зи}$, сток – исток $C_{си}$, как правило, считаются постоянными.

Модели распределенных элементов. Законы Кирхгофа справедливы для любых электронных цепей, если скорость изменения токов и напряжений в схеме была много меньше скорости распространения этих изменений или геометрические размеры элементов цепи были много меньше длины волны. Если эти условия не выполняются, то параметры носят распределенный характер и вместо законов Кирхгофа следует использовать уравнения Максвелла. Распределенный характер цепей обычно учитывают (в зависимости от размеров элементов) на частотах 0,3...1 ГГц и выше, что характерно для работы многих высокоскоростных микропроцессорных и для большинства аналоговых устройств.

В настоящее время существует несколько сильно различающихся подходов к моделированию распределенных структур и устройств [46; 47]. Так, для специализированных программ анализа линейных устройств СВЧ используют аппарат матриц рассеяния (S -матриц). В то же время для программ анализа нелинейных аналоговых ВЧ и СВЧ устройств удобнее оказались модели линейных компонентов в виде Y -матрицы, в которой элементы матрицы зависят от частоты. Требование представления моделей распределенных элементов в виде частотно-зависимых S - или Y -матриц накладывает одно существенное ограничение на методы анализа нелинейных схем. Это ограничение сводится к тому, что для моделирования, по крайней мере линейной части, необходимо использовать только методы анализа в частотной области. В программах моделирования цифровых схем во временной области находят применение модели в виде эквивалентных схем замещения (содержат сосредоточенные элементы) и в виде интегро-дифференциальных уравнений, решение которых выполняется совместно с решением дифференциальных уравнений модели остальной части схемы. Но часто эти модели становятся неадекватными с повышением частоты анализа.

В программах моделирования на основе Y -матриц нередко используют два уровня моделей для распределенных элементов. Первый уровень – это модели на основе теории длинных линий, например модель отрезка микрополосковой линии (МПЛ), второй уровень – учет нерегулярностей в МПЛ (модели неоднородностей). Отрезок регулярной МПЛ представляется Π -образной схемой замещения (рис. 4.6), в которой параметры схемы замещения определяются по выражениям

$$X_L = Z_B \cdot \sin(2\pi l \sqrt{\epsilon_{\text{эф}}} / \lambda);$$

$$X_C = Z_B \cdot \text{ctg}(\pi l \sqrt{\epsilon_{\text{эф}}} / \lambda),$$

где Z_B – волновое сопротивление МПЛ; l – длина отрезка линии; λ – рабочая длина волны; $\epsilon_{\text{эф}}$ – эффективная диэлектрическая проницаемость материала подложки [46].

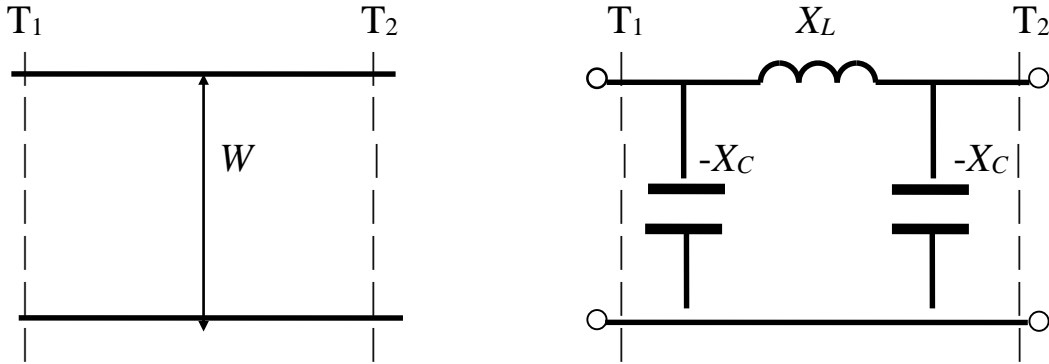


Рис. 4.6. Эквивалентная схема для МПЛ

Модели неоднородностей в МПЛ представляются также в виде схем замещения, полученных на основе модифицированного метода Олинера в рамках одномодового приближения [46]. В этих моделях эффективная диэлектрическая проницаемость $\epsilon_{\text{эф}}$ зависит от частоты, модели обеспечивают высокую точность до частот 10...18 ГГц. Некоторые эквивалентные схемы неоднородностей приведены в тексте: скачок ширины полоски (рис. 4.7) и открытый конец МПЛ, который моделируется в виде эквивалентного удлинения полоски (рис. 4.8) на величину Δl .

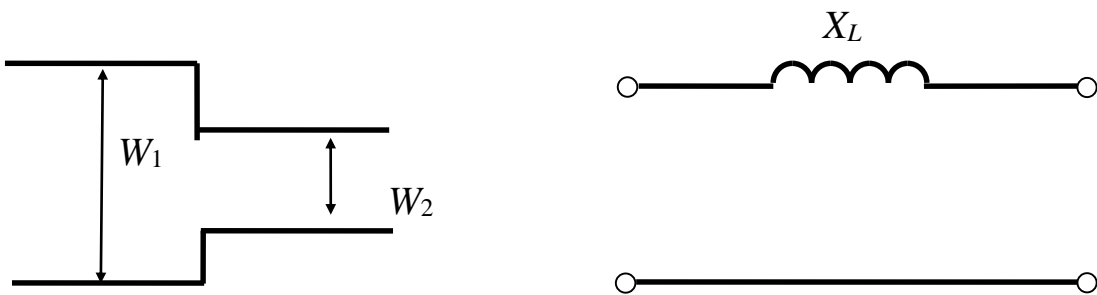


Рис. 4.7. Модель скачка ширины МПЛ

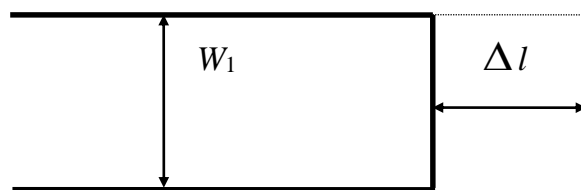


Рис. 4.8. Модель открытого конца МПЛ

4.3. Математическая модель схемы

Математическая модель схемы формируется на основе ММЭ и законов Кирхгофа. В большинстве программ АСхП используется базис узловых потенциалов (может быть использован базис контурных токов или переменных состояний, но они не нашли широкого распространения). В этом случае ММС формируется на основе первого закона Кирхгофа

$$\sum_{k=1}^n i_k = 0,$$

где алгебраическая сумма токов в любом узле k равна нулю.

С использованием ММЭ и закона токов Кирхгофа в общем случае ММС может быть представлена системой нелинейных дифференциальных уравнений (СНДУ), которую обычно записывают в виде

$$F(U', U, t) = 0 \quad (4.4)$$

или нормальной формы Коши

$$U' = f(U, t) = 0,$$

где $U' = dU/dt$; U – вектор переменных, в частности потенциалов узлов схемы.

Важно отметить, что по исходному описанию схемы на входном языке программ моделирования ММС формируется не в виде уравнения (4.4) и даже не в виде формул решения этих уравнений, а в виде векторов и матриц, входящих в состав реализованного в программах численного алгоритма решения уравнений [2].

Для решения СНДУ в программах моделирования наиболее распространены неявные методы численного интегрирования, где при решении уравнения (4.4) заменяют производные их конечно-разностными аппроксимациями, соответствующими различным формулам численного интегрирования. Например, для простейшей неявной формулы Эйлера первого порядка аппроксимация будет задаваться соотношением

$$U' = dU/dt \approx (U_{n+1} - U_n)/h,$$

где $U_{n+1} = U(t_{n+1})$, $U_n = U(t_n)$, $h = t_{n+1} - t_n$ – шаг интегрирования.

Подстановка вышеприведенных выражений в выражение (4.4) позволяет преобразовать СНДУ в систему нелинейных алгебраических (конечных) уравнений (СНАУ)

$$F(U_{n+1}, U_n, \dots, U_{n-k}) = 0,$$

или

$$F(U) = 0. \quad (4.5)$$

В записи (4.5) рассмотрен общий случай аппроксимации неявными формулами численного интегрирования $k + 1$ порядка.

Подстановка дискретных компонентных аппроксимаций и получение уравнения (4.5), не содержащего производных, называется *алгебраизацией* исходных уравнений.

Решение системы (4.5) относительно U_{n+1} чаще всего реализуется с помощью итерационного метода Ньютона, основанного на *линеаризации* компонентных уравнений на каждой итерации:

$$F^{i+1} = F^i + dF/dU(U^{i+1} - U^i) + \dots = 0.$$

В этом случае решение нелинейных уравнений сводится к решению на каждой итерации системы линейных алгебраических уравнений (СЛАУ), обычно представляемых в методе узловых потенциалов в виде

$$YU = I, \quad (4.6)$$

где Y – матрица узловых проводимостей (якобиан) схемы, $Y = dF/dU$; U – вектор неизвестных узловых потенциалов, $U = U^{i+1} - U^i$; I – вектор задающих токов, $I = -Fi$; i – номер итерации.

Таким образом, для решения ММС в общем виде, т. е. в виде СНДУ, необходимо на каждой временной точке t_n решать СНАУ (рис. 4.9). В свою очередь, для решения СНАУ на каждой итерации метода Ньютона необходимо решать СЛАУ. Поэтому основой при формировании ММС для задач АСхП является матрица узловых проводимостей Y и вектор тока I системы линейных уравнений (4.6). Те же Y и I с помощью несложных эквивалентных преобразований теории цепей используются и при решении уравнений (4.4) и (4.5) [10; 43].

В то же время, как показано на рис. 4.9, решение трёх основных состояний модели схемы в виде СНДУ, СНАУ и СЛАУ – основа трёх базовых алгоритмов анализа АСхП: переходного, статического и линейного режимов, которые присутствуют во всех программах АСхП.



Рис. 4.9. Связь состояний математической модели схемы с базовыми видами анализа

4.4. Анализ линейного режима

Анализ линейного режима (может называться также малосигнальным анализом, анализом частотных характеристик, small signal analysis, alternative current analysis, AC analysis) выполняется для расчета характеристик схемы, в которой ее компоненты заменяются линейными моделями, справедливыми, конечно, только при малых изменениях сигналов. Типичный пример этого анализа – расчет амплитудно-частотных и фазо-частотных характеристик (АЧХ и ФЧХ).

Основная система уравнений, которая решается в этом анализе, – это система линейных алгебраических уравнений типа (4.6):

$$YV = I,$$

где Y – матрица узловых проводимостей схемы; V – вектор неизвестных узловых потенциалов; I – вектор задающих токов.

Задача формирования ММС в виде уравнения (4.6) сводится в этом случае к заполнению матрицы Y и вектора I , что может быть легко выполнено в программах непосредственно по исходному описанию схемы.

Рассмотрим простейший пример схемы (рис. 4.10) и сформируем уравнения для нее. Для этого получим следующие уравнения:

- для узла 0: $i_1 + i_5 - i_4 = 0$;
- для узла 1: $I_1 - i_1 + i_2 = 0$;
- для узла 2: $I_2 + I_3 - i_2 - i_5 = 0$;
- для узла 3: $i_3 - I_3 = 0$;
- для узла 4: $i_4 - I_1 - I_2 - i_3 = 0$.

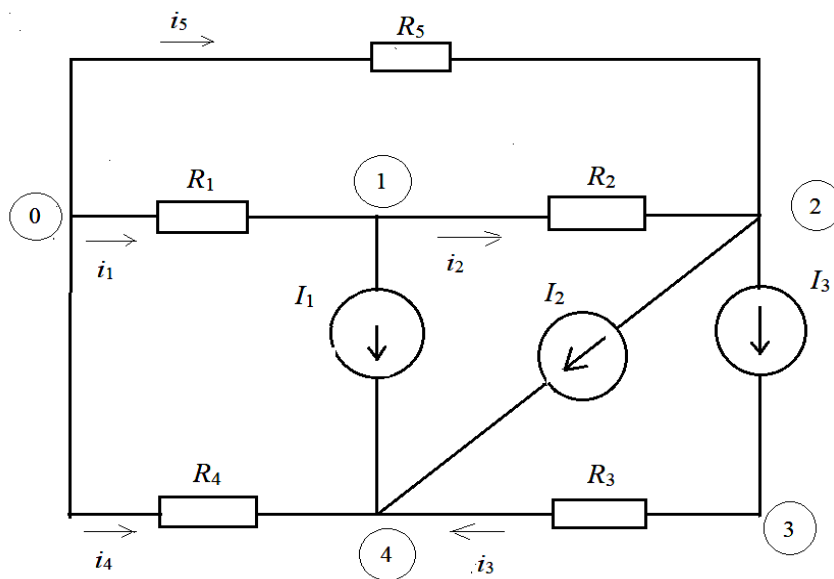


Рис. 4.10. Пример электрической схемы

Выразим токи в схеме через элементы схемы и напряжения:

$$i_1 = \frac{1}{R_1} V_1, \quad i_2 = \frac{1}{R_2} (V_1 - V_2), \quad i_3 = \frac{1}{R_3} (V_3 - V_4), \quad i_4 = \frac{1}{R_4} V_4.$$

Сформируем систему узловых уравнений в матричном виде (4.6), где каждый ряд и столбец соответствуют номеру узла.

В матричном виде

$$\begin{bmatrix} \frac{1}{R_1} + \frac{1}{R_2} & -\frac{1}{R_2} & & & \\ -\frac{1}{R_2} & \frac{1}{R_2} + \frac{1}{R_5} & & & \\ & & \frac{1}{R_3} & -\frac{1}{R_3} & \\ & & -\frac{1}{R_3} & \frac{1}{R_3} + \frac{1}{R_4} & \\ & & & & \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} -I_1 \\ -I_2 - I_3 \\ I_3 \\ I_1 + I_2 \end{bmatrix}$$

Алгоритм формирования (заполнения) матрицы узловых проводимостей можно представить в следующем виде (n_1 и n_2 – номера узлов подключения элемента):

if $(n_1 > 0) \& (n_2 > 0)$

$$Y(n_1, n_1) = Y(n_1, n_1) + 1/R, \quad Y(n_2, n_1) = Y(n_2, n_1) - 1/R;$$

$$Y(n_1, n_2) = Y(n_1, n_2) - 1/R, \quad Y(n_2, n_2) = Y(n_2, n_2) + 1/R;$$

else if $(n_1 > 0)$

$$Y(n_1, n_1) = Y(n_1, n_1) + 1/R;$$

else

$$Y(n_2, n_2) = Y(n_2, n_2) + 1/R.$$

Вектор токов I , как видно из примера, заполняется по сходным правилам.

Таким образом, при соответствии номеров узлов схемы номерам строк и столбцов матрицы Y проводимость любого двухполюсного элемента, включенного между узлами i и j , будет записываться в матрицу Y в качестве слагаемого в четыре позиции: со знаком «+» в главную диагональ на позиции ii и jj и со знаком «-» на позиции ij и ji , расположенные симметрично относительно диагонали (рис. 4.11). Если один из узлов двухполюсника – опорный («земля»), то проводимость двухполюсника будет учитываться лишь в одном элементе матрицы Y – в позиции ii , где i – незаземленный узел.

Аналогично вкладом нелинейного двухполюсника с уравнением $i = f(u)$ будет проводимость $y = \partial i / \partial u$. Однако многие элементы схем описываются уравнениями отличного от $i = f(u)$ вида. Такие элементы считаются неудобными для базиса узловых потенциалов. К ним относятся идеальные источники напряжения E , индуктивности и управляемые источники вида $i = f(i)$, $u = f(i)$, $u = f(u)$. Для неудобных элементов можно использовать ряд приемов, позволяющих представить их уравнение в виде $i = f(u)$ [10] или расширенный координатный базис (см. п. 6.3).

Для компонентов, содержащих модели источников тока, управляемого напряжением $i_{ij} = f(u_{kl})$, матрица Y будет несимметричной, так как крутизна этого источника $g = \partial i / \partial u$ будет записываться в позиции, показанной на рис. 4.12.

Формирование матрицы узловых проводимостей для многополюсников требует некоторых пояснений. Пусть в схеме имеется

n -полюсник, каждый полюсный ток i_k которого непосредственно зависит от потенциалов всех остальных полюсов, т. е. уравнение тока имеет вид $i_k = f(u_1, \dots, u_n)$.

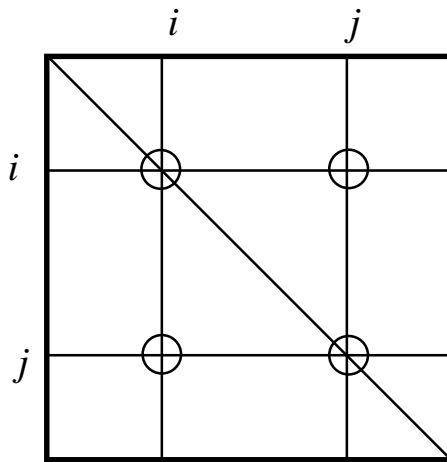


Рис. 4.11. Заполнение матрицы Y для симметричных элементов

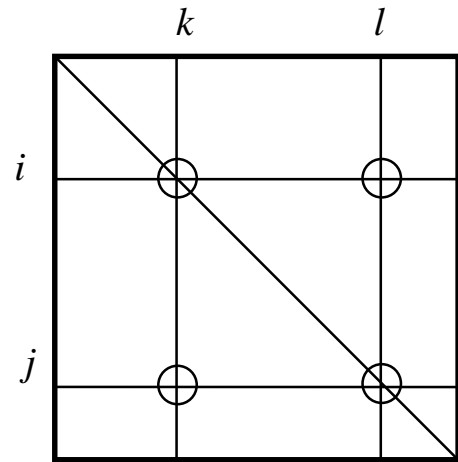


Рис. 4.12. Заполнение матрицы Y для несимметричных элементов

Пусть k, l, \dots, j – номера полюсов многополюсника в схеме. Тогда матрица полюсных проводимостей многополюсника будет иметь вид

$$Y_{\text{MH}} = \begin{array}{c} \begin{array}{cc} k & l & \dots & j \end{array} \\ \begin{array}{c} k \\ l \\ \cdot \\ \cdot \\ j \end{array} \end{array} \begin{array}{cccc} \hline y_{kk} & -y_{kl} & \dots & -y_{kj} \\ -y_{lk} & y_{ll} & \dots & -y_{lj} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ -y_{jk} & -y_{jl} & \dots & y_{jj} \end{array}$$

Здесь y_{kk}, y_{ll}, y_{jj} – собственные проводимости полюсов k, l, \dots, j , равные суммам проводимостей n -полюсника, подсоединенного к полюсам k, l, \dots, j соответственно; y_{kl}, y_{kj}, \dots и т. д. – взаимные проводимости между полюсами n -полюсника. Например, любой биполярный транзистор можно рассматривать как трехполюсник с матрицей проводимостей [9]

$$Y_{\text{тр}} = \begin{array}{c} \begin{array}{ccc} \text{э} & \text{б} & \text{к} \end{array} \\ \begin{array}{c} \text{э} \\ \text{б} \\ \text{к} \end{array} \end{array} \begin{array}{ccc} \hline y_{\text{ээ}} & -y_{\text{эб}} & -y_{\text{эк}} \\ -y_{\text{бэ}} & y_{\text{бб}} & -y_{\text{бк}} \\ -y_{\text{кэ}} & -y_{\text{кб}} & y_{\text{кк}} \end{array}$$

Формулы расчетов для элементов этой матрицы различны и зависят от выбранной схемы замещения модели транзистора (см. п. 4.2).

Необходимо учитывать, что большинство моделей транзисторов содержат элементы источников тока, управляемых напряжением. В этом случае матрица проводимостей будет несимметричной.

Основное назначение линейного анализа – расчет частотных характеристик линейных или линеаризованных в заданных или рассчитанных в статическом режиме рабочих точках схем и построение графиков частотных зависимостей напряжений на элементах в выходных узлах схемы или коэффициентов передачи. Алгоритмы анализа в линейном режиме требуют очень малых вычислительных затрат и поэтому позволяют достаточно быстро оценить частотные свойства, полосу пропускания устройств, неравномерность частотной характеристики и все характеристики устройства как эквивалентного четырехполюсника.

Коэффициент передачи определяется из соотношения

$$K(j\omega) = U_{\text{вых}}(j\omega)/U_{\text{вх}}(j\omega),$$

где $U_{\text{вых}}$ – разность потенциалов узлов выходной цепи; $U_{\text{вх}}$ – заданная амплитуда входного воздействия; ω – круговая частота, на которой выполняется расчет, $\omega = 2\pi f$. Вектор узловых потенциалов определяется из решения системы линейных уравнений вида

$$Y(j\omega)U(j\omega) = I(j\omega), \quad (4.7)$$

где $Y(j\omega)$ – матрица узловых проводимостей, элементами которой являются проводимости всех линейных компонентов схемы на частоте ω , а также элементы эквивалентных схем активных элементов, линеаризованных в рабочей точке. Элементы матрицы проводимости для реактивных ветвей схемы определяются соотношениями

$$Y_C = j\omega C,$$

$$Y_L = 1/(-j\omega L).$$

При формировании матричных уравнений постоянные источники напряжения схемы закорачиваются, а постоянные источники тока замыкаются. При моделировании схем в линейном режиме нет особой необходимости в расчете при реальной амплитуде входного воздействия, поэтому в большинстве программ в качестве входного сигнала используется источник напряжения $U_{\text{вх}}(j\omega)$ с единичной амплитудой и нулевой фазой. В этом случае вектор $U(j\omega)$ будет непосредственно содержать коэффициенты передачи узлов схемы относительно входа,

вектор тока $I(j\omega)$ при решении уравнения (4.7) будет содержать везде нулевые значения, кроме единичного – во входной ветви.

Пример. Сформируем матрицу узловых проводимостей для схемы на рис. 4.13.

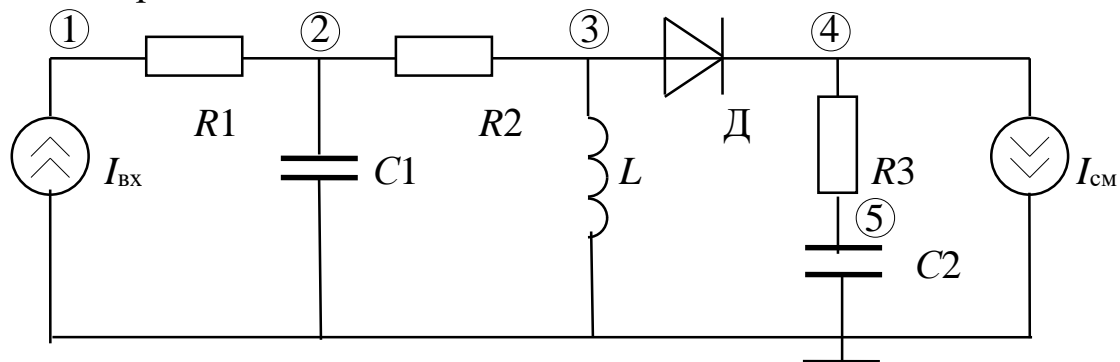


Рис. 4.13. Электрическая схема

В качестве модели диода воспользуемся моделью рис. 4.2 и пренебрежем для упрощения сопротивлением R_d , а емкость диода будем считать постоянной. Нелинейная зависимость тока диода в малосигнальном режиме должна быть линеаризована, например в рабочей точке U_0 :

$$g = \frac{\partial I_D}{\partial U} \Big|_{U_0} = I_0 / mV_T \exp\left(\frac{U_0}{mV_T}\right).$$

Тогда матрица проводимостей будет иметь следующий вид (рис. 4.14).

$1/R1$	$-1/R1$	0	0	0
$-1/R1$	$1/R1 + j\omega C1 + 1/R2$	$-1/R2$	0	0
0	$-1/R2$	$1/R2 + g + 1/j\omega L + j\omega C_d$	$-g - j\omega C_d$	0
0	0	$-g - j\omega C_d$	$1/R3 + g + j\omega C_d$	$-1/R3$
0	0	0	$-1/R3$	$1/R3 + j\omega C2$

Рис. 4.14. Матрица проводимостей для схемы рис. 4.13

Решение системы (4.7) дает информацию о всех потенциалах (коэффициентах передачи) узлов схемы. Если же интерес представляет частотная характеристика в каком-нибудь одном узле схемы, например k , то на каждой частоте ω_i будет выбираться из вектора $V(j\omega_i)$ комплексное значение $V_k(j\omega_i)$ и вычисляться модуль и фаза по соотношениям

$$A_k(\omega_i) = \sqrt{\operatorname{Re}[V_k(j\omega_i)]^2 + \operatorname{Im}[V_k(j\omega_i)]^2},$$

$$\varphi_k(\omega_i) = \operatorname{arctg} \frac{\operatorname{Im}[V_k(j\omega_i)]}{\operatorname{Re}[V_k(j\omega_i)]}.$$

Решение системы линейных уравнений (4.6) – (4.7) может быть выполнено с использованием стандартных программ, имеющихся в обеспечении любых ЭВМ. Особенность этих алгоритмов в АСхП состоит в том, что решение систем вида (4.6) не только является самостоятельной задачей при расчете частотных характеристик в режиме малого сигнала, но (как было показано ранее) служит одной из наиболее часто повторяющихся процедур при решении других более сложных задач (например, задач расчета статического режима, переходного процесса) и, как будет показано позже, при расчете шумов, коэффициентов чувствительности и нелинейных искажений схем. В связи с этим вычислительные затраты всей системы моделирования будут во многом определяться вычислительными затратами на решение уравнения (4.6).

Наибольшее распространение в программах АСхП получил метод исключения Гаусса и его разновидность – метод LU -разложения [2; 3; 10]. Метод LU -разложения имеет преимущества перед методом Гаусса, если система линейных уравнений должна решаться много раз с одной и той же матрицей Y , но с разными векторами правых частей I .

В методе решение системы линейных уравнений в общем виде $AX = B$ реализуется в три этапа:

– этап разложения матрицы A на треугольные матрицы L и U :

$$LU \times X = B,$$

где L – нижнетреугольная матрица; U – верхнетреугольная матрица;

– этап прямой подстановки

$$L \times Y = B;$$

– этап обратной подстановки

$$U \times X = Y.$$

Так как разложение на L и U рекуррентное, то элементы матриц L и U хранятся на месте элементов матрицы A .

Сложность и специфика программной реализации этих методов в АСхП заключается в том, что матрицы узловых проводимостей Y имеют очень большую размерность: так, например, при числе узлов n , равном 100, размерность матрицы будет n^2 , т. е. 10 000 ячеек памяти ЭВМ. Кубическая зависимость объема вычислений от размерности задач дополнительно ограничивает сложность анализируемых схем. Проектируемые узлы ЭВА могут содержать десятки тысяч и более узлов, что приводит к значительным вычислительным затратам.

Здесь важно отметить, что матрицы проводимостей электронных схем являются слабо заполненными, или разреженными, что позволяет значительно (примерно на порядок) сократить объем требуемой оперативной памяти, хотя приводит к усложнению алгоритмов решения, так как программы решения систем линейных уравнений будут уже оперировать не матрицами, а векторами индексов только заполненных (ненулевых) ячеек [3; 10; 43].

4.5. Анализ статического режима

Анализ статического режима (режима по постоянному току, статике, *direct current analysis*, DC) может иметь самостоятельное значение, например определение постоянных напряжений и токов в различных точках схемы, рабочих точек активных элементов, потребляемых и рассеиваемых мощностей в различных точках и элементах схемы. Однако чаще всего анализ статического режима – основа других, более сложных задач, например при определении начальных условий и анализе переходного режима численными методами интегрирования.

Решается модель схемы в виде СНАУ

$$F(V) = 0. \quad (4.8)$$

Для решения системы (4.8) можно использовать *метод простой итерации*

$$V^{k+1} = V^k - pF(V^k),$$

где k – номер итерации; p – множитель, регулирующий сходимость. Главный недостаток этого алгоритма – медленная сходимость. Поэтому для решения системы (4.8) чаще всего используют итерационный *метод Ньютона*, где новое значение вектора неизвестных потенциалов определяется по формуле

$$V^{k+1} = V^k - J(V^k)^{-1} \times F(V^k), \quad (4.9)$$

где $J(V^k) = dF(V)/dV|_{V^k}$ – матрица Якоби (якобиан), определяемая как

$$J = \begin{bmatrix} dF_1/dV_1 & dF_1/dV_2 & \dots & dF_1/dV_n \\ dF_2/dV_1 & dF_2/dV_2 & \dots & dF_2/dV_n \\ \dots & \dots & \dots & \dots \\ dF_n/dV_1 & dF_n/dV_2 & \dots & dF_n/dV_n \end{bmatrix}.$$

Алгоритм нахождения нового значения по уравнению (4.9) наиболее экономично определять из решения системы линейных уравнений вида

$$J(V^k) \times \Delta V^{k+1} = -F(V^k), \quad (4.10)$$

где ΔV^{k+1} – вектор поправок, $\Delta V^{k+1} = V^{k+1} - V^k$.

В отличие от линейно сходящегося метода простых итераций алгоритм Ньютона сходится квадратично и обычно получается 5...10 ньютоновских итераций, тогда как методом простых итераций требуется несколько десятков или даже сотен итераций. К недостаткам метода Ньютона можно отнести малый размер области сходимости (обычно меньше, чем для простых итераций) и необходимость задания точки начального приближения, достаточно близкого к решению (в противном случае скорость сходимости заметно снижается).

Для уточнения алгоритма анализа в статическом режиме, формирования и решения уравнений (4.10) рассмотрим компонентное нелинейное уравнение на примере резистивной нелинейности [33]:

$$i = f(u). \quad (4.11)$$

Разлагая уравнение (4.11) в ряд Тейлора относительно текущей итерации и учитывая лишь линейный член ряда (линеаризация), получим

$$i^{k+1} = i^k + (u^{k+1} - u^k) di/du + \dots$$

или

$$i^{k+1} = i^k + (u^{k+1} - u^k)g^k = g^k u^{k+1} + I_{\text{ЭКВ}}^k, \quad (4.12)$$

где g^k – крутизна (производная) нелинейности, $g^k = \left. \frac{di}{du} \right|_{u^k}$; $I_{\text{ЭКВ}}^k = i^k - g^k u^k$.

Уравнению (4.12) соответствует эквивалентная схема замещения (рис. 4.15), называемая итерационной и используемая при формировании модели схемы в статическом режиме. В этом случае для нахождения нового значения вектора V^{k+1} необходимо решать СЛАУ вида

$$J(V^k)V^{k+1} = I_{\text{ЭКВ}}^k, \quad (4.13)$$

где J – матрица узловых проводимостей схемы Y в режиме по постоянному току (т. е. только активные сопротивления, емкостные элементы будут в режиме холостого хода, а индуктивные – короткого замыкания) с добавлением нелинейных элементов, замененных их производными при предыдущем значении V^k . Вектор $I_{\text{ЭКВ}}^k$ будет содержать составляющие постоянных источников схемы (источники питания, смещения и др.) и эквивалентных источников тока $I_{\text{ЭКВ}}^k$ нелинейных элементов.

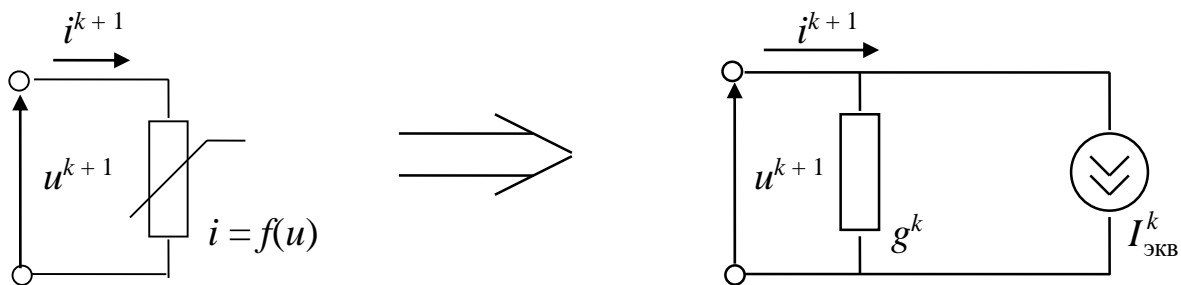


Рис. 4.15. Преобразование модели нелинейного элемента в итерационную эквивалентную схему замещения

Итерационный процесс решения уравнения (4.13) заканчивается при выполнении условия

$$|V^{k+1} - V^k| \leq \varepsilon,$$

где ε – заданная погрешность расчета.

По найденным в результате расчетов токам и напряжениям в программах АСХП вычисляются различные мощности (как потребляемые, так и рассеиваемые), напряжения и токи на переходах активных приборов (рабочие точки). Алгоритм анализа можно представить в виде последовательности шагов.

Шаг 1. Определение начального приближения для потенциалов узлов схемы по постоянному току: $V_j^i, i = 0, j = 1, \dots, n$ (n – число узлов схемы). Начальное приближение может быть задано либо нулевым, либо конкретными значениями в исходных данных из физических соображений.

Шаг 2. Организация итераций для решения уравнений методом Ньютона, $i = i + 1$.

Шаг 3. Заполнение матрицы Якоби $J(V^{i-1})$. Резистивные элементы заносятся по исходному описанию схемы, параметры нелинейных приборов ($g^{i-1}, I_{\text{экв}}^{i-1}$) – путем обращения к подпрограммам моделей.

Шаг 4. Решение системы линейных уравнений (4.13).

Шаг 5. Если сходимость достигнута $|V^i - V^{i-1}| < \varepsilon$, то переходят к шагу 6, иначе – к шагу 2.

Шаг 6. Обработка и сохранение результатов расчета V^i .

Пример. Сформируем якобиан для схемы рис. 4.13. В режиме постоянного тока емкости схемы не учитываются, а индуктивности заменяются проводимостями $g_L = 1$; нелинейная зависимость тока диода будет учитываться проводимостью $g^k = I_0/mV_T \exp(U^k/mV_T)$. Якобиан будет иметь следующий вид (рис. 4.16).

1/R1	-1/R1	0	0	0
-1/R1	1/R1 + 1/R2	-1/R2	0	0
0	-1/R2	1/R2 + $g_L + g^k$	- g^k	0
0	0	- g^k	1/R3 + g^k	-1/R3
0	0	0	-1/R3	1/R3

Рис. 4.16. Матрица проводимостей по постоянному току для схемы рис. 4.13

4.6. Анализ переходного режима

Анализ переходного режима (называемый также анализом в динамическом режиме, анализом во временной области, transient или time domain analysis) предназначен для получения формы отклика на выходе схемы при заданном воздействии на входе (рис. 4.17).

Модель схемы в виде СНДУ

$$F(V', V, t) = 0, \quad (4.14)$$

где $V' = \frac{dV}{dt}$, решается методами численного интегрирования с определением $V(t)$, при заданных начальных условиях $V_0 = V(t_0)$. Искомые

функции выходного напряжения $V(t)$ определяются в отдельных точках интервала t_1, t_2, \dots, t_k в виде значений V_1, V_2, \dots, V_k , приближенно равных значениям $V(t_1), V(t_2), \dots, V(t_k)$ точного решения $V(t)$. Расстояние $\Delta t = t_{k+1} - t_k = h$ называется шагом интегрирования и может быть задано перед началом вычислений (интегрирование с постоянным шагом) либо определено в процессе вычислений в зависимости от погрешности расчетов (с автоматическим выбором шага). Начальное условие V_0 может быть задано в исходных данных или вычислено с использованием анализа статического режима при учете постоянных составляющих источников воздействий и питания.

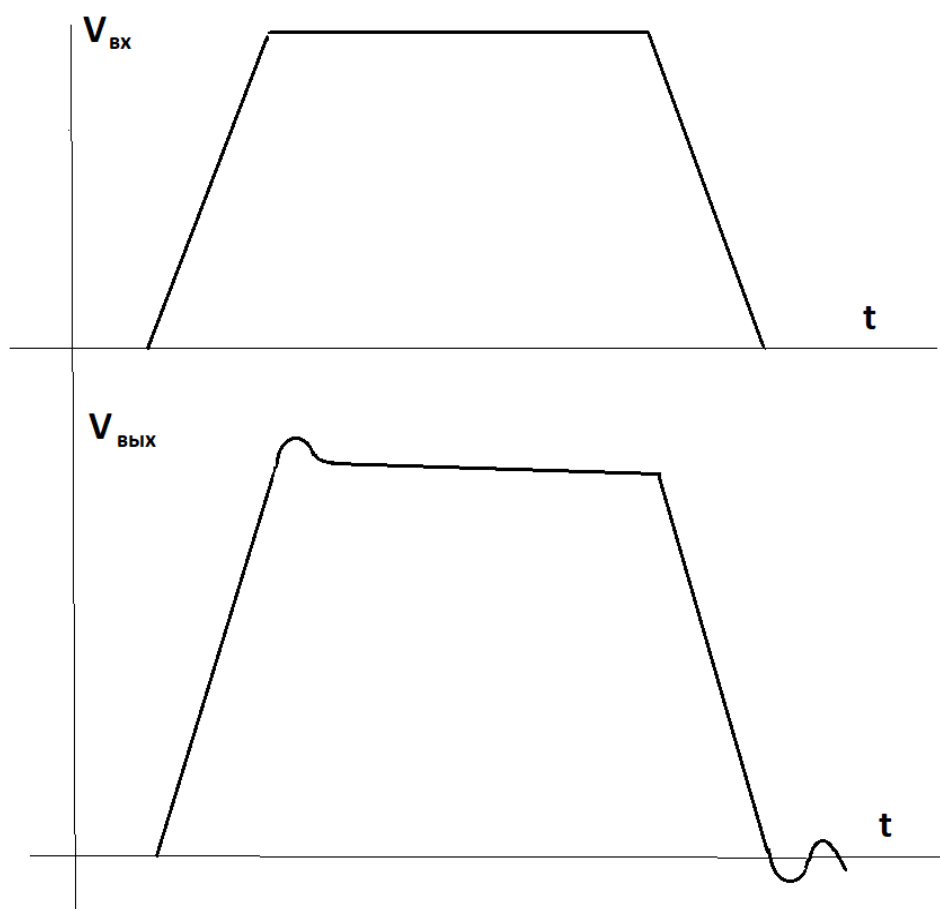


Рис. 4.17. Входное воздействие и отклик в анализе переходного режима

Решение СНДУ (4.14) или уравнения $V' = f(V, t)$ методами численного интегрирования можно рассмотреть на примере простейших методов первого порядка, называемых *явными* и *неявными методами Эйлера*.

Для двух соседних точек справедливо соотношение

$$V_{k+1} = V_k + \int_{t_k}^{t_{k+1}} f(V, t) dt.$$

Решение интеграла численным методом сводится к вычислению площади подынтегральной кривой. Пусть V_k и V_k' – значения переменных в момент времени t_k при численном решении уравнения (4.14). В методе Эйлера (формула прямоугольника) для V_k' применяется одно из следующих аппроксимирующих выражений [2]:

$$V_{k+1} \approx V_k + hf(V_k, t_k),$$

$$V_{k+1} \approx V_k + hf(V_{k+1}, t_{k+1}),$$

где h_k – величина шага с номером k , $h_k = t_k - t_{k-1}$. Эти формулы приводят к виду, соответствующему явному и неявному методам Эйлера.

Для иллюстрации приведенных соотношений рассмотрим рис. 4.18, где площадь прямоугольника может быть вычислена двумя способами, различающимися только тем, какое значение функции выбрано в качестве высоты прямоугольника.

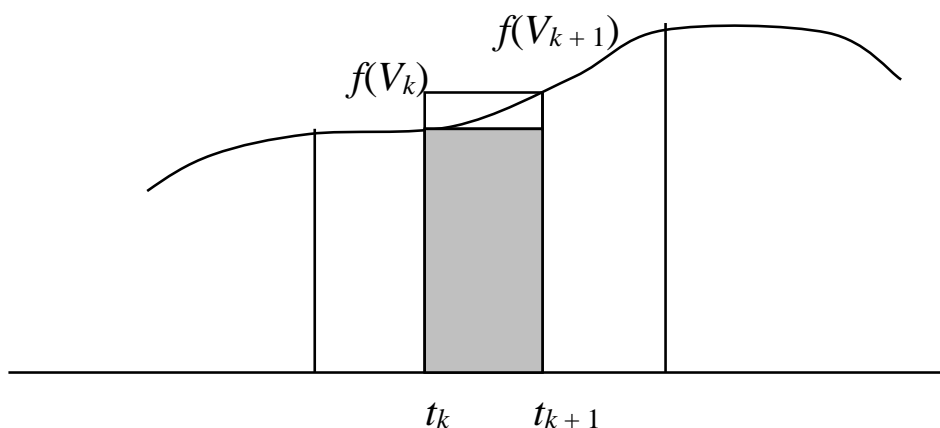


Рис. 4.18. Иллюстрация метода Эйлера

На первый взгляд, явная формула имеет значительные преимущества, так как новое значение функции вычисляется по известным значениям (явно) V_k , h и $f(V_k)$ по простой формуле.

В неявном методе новое значение находится и в левой, и в правой частях уравнения V_{k+1} , т. е. связано неявно. Для нахождения V_{k+1} необходимо применять какой-либо итерационный метод, что увеличивает вычислительные затраты. Поэтому многие годы в программах моделирования использовались явные методы численного интегрирования.

Явные формулы наиболее удобны в случаях, когда математическую модель схемы удастся представить в виде

$$V' = f(V, t)$$

с известными начальными условиями V_0 . Тогда при использовании метода Эйлера на первом шаге интегрирования получим $V_1 = V_0 + h_1 f(V_0, t_0)$, на втором шаге $V_2 = V_1 + h_2 f(V_1, t_1)$ и так далее, на k -м шаге

$$V_k = V_{k-1} + h_k f(V_{k-1}, t_{k-1}).$$

Из вышеприведенного выражения видно, что, применяя явное интегрирование к нормальной форме, можно вообще избежать решения системы конечных уравнений, которое в случае неявного интегрирования необходимо осуществлять на каждом шаге. Трудоемкость выполнения одного шага мала – в этом состоит преимущество явных методов интегрирования.

Если система уравнений задана в общей форме (4.14), то трудоемкость одного шага явного интегрирования увеличивается, так как приходится вычислять V_k путем решения системы конечных уравнений. Однако и в этом случае указанная трудоемкость остается меньше трудоемкости шага неявного интегрирования. Но большая простота выполнения одного шага еще не означает меньших вычислительных затрат на всем отрезке интегрирования $[0, T]$, так как затраты определяются также количеством шагов. В подавляющем большинстве случаев анализа электронных схем применение явных методов характеризуется большим количеством шагов, чем применение неявных методов. Это связано с тем, что для явного интегрирования характерна ограниченная устойчивость вычислений. В этом недостаток явных методов. Кроме того, явные методы имеют следующие сильные ограничения:

1. В современных высокоскоростных электронных схемах имеется сильный разброс постоянных времени, что приводит к так называемым «жестким» дифференциальным уравнениям, когда для обеспечения достаточной точности вычислений шаг интегрирования приходится выбирать сверхмалым, что значительно увеличивает вычислительные затраты и сводит преимущества явных методов на нет.

2. В явных методах имеют место накопление ошибки от шага к шагу и численная неустойчивость. Для уменьшения ошибки применяют формулы интегрирования очень высоких порядков, что приводит дополнительно к увеличению вычислительных затрат.

В неявные формулы искомый на k -м шаге вектор V_k входит неявно и может быть найден путем решения системы конечных уравне-

ний. Несмотря на большую трудоемкость выполнения одного шага, неявные методы получили наибольшее распространение в программах АСхП в связи с устойчивостью порождаемого ими вычислительного процесса.

Явление неустойчивости вычислительного процесса при использовании явных формул связано с тем, что погрешность, допущенная на некотором шаге интегрирования, порождает увеличенную погрешность на последующих шагах. Неустойчивость связана с таким катастрофическим ростом погрешностей, что получаемые результаты численного решения ни в коей мере не отражают действительного характера интегральных кривых. Явление неустойчивости вычислений недопустимо при решении практических задач. Поэтому в САПР должны применяться устойчивые методы.

Среди методов второго порядка точности наиболее известен *метод трапеций*, в котором аппроксимация для V_k' получается из формулы трапеций:

$$V_k = V_{k-1} + h_k (V_k' + V_{k-1}') / 2. \quad (4.15)$$

Для методов высоких порядков можно записать обобщенную формулу

$$V_k = \sum_{i=1}^p a_i V_{k-1} + h \sum_{i=1}^p b_i V_{k+1-i}'$$

где a_i и b_i – коэффициенты аппроксимации; p зависит от порядка метода. Если $b_1 = 0$, то имеем явную формулу, если $b_1 \neq 0$ – неявную формулу.

Неявные методы первого и второго порядка являются устойчивыми и наиболее распространены в программах АСхП. В этом случае типичный алгоритм анализа переходного режима будет включать в себя следующие шаги.

Шаг 1. Нахождение начального приближения V_{k0} к исходному вектору V_k . Начальное приближение может быть получено из расчета статического режима.

Далее алгоритм включает шаги, повторяющиеся на каждой временной точке отрезка интегрирования.

Шаг 2. Определение величины шага h_k . В современных программах АСхП величина шага вычисляется в зависимости от скорости изменения процессов в схеме (специальный алгоритм). Для медленных участков изменения процесса величина шага увеличивается, для быстро изменяющихся процессов – уменьшается.

Шаг 3. Вычисление V_k путем решения системы конечных уравнений (4.15) методом Ньютона. В современных программах делается только одна итерация Ньютона (что значительно сокращает вычислительные затраты), так как на каждой последующей временной точке решение опять будет уточняться.

Шаг 4. Переход к шагу 2, если временная точка расчета переходного режима не превысила конечного времени.

В современных алгоритмах практически отсутствует само интегрирование. Метод решения СНДУ будет определяться формулой дискретизации (4.15). Так, использование формулы

$$dV/dt = V'_k \approx (V_k - V_{k-1})/h_k$$

соответствует решению СНДУ методом Эйлера, а использование формулы

$$dV/dt = V'_k \approx 2(V_k - V_{k-1})/h_k - V'_{k-1} - \text{решению методом трапеций.}$$

Более того, формулы дискретизации можно даже не связывать с каким-либо конкретным численным методом решения СНДУ, а рассматривать только как соотношения, связывающие предыдущие известные значения $V_{k-n}, V_{k-n+1}, \dots, V_k$ с неизвестным последующим значением V_{k+1} . При этом каждый реактивный элемент L или C схемы может быть описан своей формулой дискретизации и при расчете схемы в целом одновременно будут реализовываться разные численные методы расчета переходных процессов, соответствующие этим формулам.

Формулы дискретизации можно применять и для любых других функций, встречающихся при описании схемы, например формула Эйлера для уравнений схемы, содержащей интегралы, имеет вид

$$\int_{t_k}^{t_{k+1}} V dt \approx h V_{k+1}.$$

Рассмотрим на примере компонентного уравнения линейной емкости особенности формирования и алгоритмов решения уравнений для неявной формулы Эйлера первого порядка

$$\begin{aligned} i_{C,n+1} &= C \cdot dU_C(t)/dt = C(U_{C,n+1} - U_{C,n})/\Delta t = \\ &= C/h \cdot U_{C,n+1} - C/h \cdot U_{C,n} = Y_\varepsilon \cdot U_{C,n+1} + I_{\text{ЭКВ},n}; \\ I_{\text{ЭКВ},n} &= -Y_\varepsilon \cdot U_{C,n}; \quad Y_\varepsilon = C/h. \end{aligned} \quad (4.16)$$

Процессу дискретизации компонентного уравнения (4.16) можно поставить в соответствие схему замещения (рис. 4.19), называемую

дискретной. Аналогично можно получить схемы замещения и для других компонентных уравнений [10; 43].

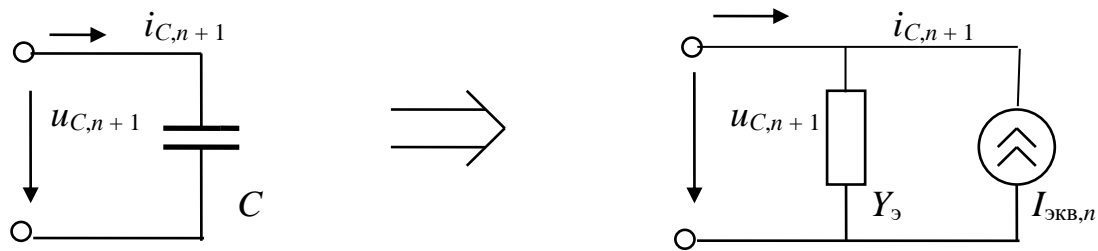


Рис. 4.19. Дискретная схема замещения

Дискретизация ММС для анализа переходного режима легко вписывается в общий алгоритм формирования и решения уравнений модели схемы в базисе узловых потенциалов и имеет вид

$$J(V_n^k) V_{n+1}^{k+1} = I_{\text{эКВ},n}^k.$$

Здесь на каждой временной точке (t_n) анализа переходного режима решается итерационно ($k = 1, 2, \dots$) методом Ньютона система линейных уравнений типа $YV = I$.

Вопросы для самопроверки

1. Задачи и особенности схемотехнического моделирования.
2. Математические модели простых и сложных элементов.
3. Математическая модель схемы и ее связь с базовыми видами анализа.
4. Основные приемы при моделировании в линейном режиме.
5. Проблемы решения систем линейных уравнений.
6. Методы решения уравнений модели в статическом режиме.
7. Назначение эквивалентной схемы замещения в статическом режиме.
8. Сравнение явных и неявных методов численного интегрирования при расчете переходного режима.
9. Алгоритм решения анализа в переходном режиме.
10. Эквивалентная схема замещения для переходного режима.

Глава 5. ДОПОЛНИТЕЛЬНЫЕ ВИДЫ АНАЛИЗА СХЕМОТЕХНИЧЕСКОГО МОДЕЛИРОВАНИЯ

5.1. Анализ чувствительности

В п. 2.2 были рассмотрены основные определения и примеры применения анализа чувствительности. Под *чувствительностью* при анализе схем понимают влияние изменения параметров компонентов и внешних условий работы схемы на ее выходные реакции или передаточную функцию [10; 43]. Количественная оценка такого влияния параметра какого-либо компонента X (сопротивление, крутизна, температура, геометрические параметры и т. п.) на изменение выходной функции схемы F (напряжения и токи, импедансы, различные коэффициенты передачи и т. п.) называется *функцией относительной чувствительности* (или просто коэффициентами чувствительности):

$$S_X^F = \frac{dF/F}{dX/X} = \frac{d(\ln F)}{d(\ln X)} = \frac{X dF}{F dX}.$$

Функция чувствительности S_X^F показывает отношение относительного изменения функции цепи F к относительному изменению параметра X при условии, что изменения достаточно малы. S_X^F часто называют *нормализованной чувствительностью*. Находят применение и полунормализованные чувствительности

$$X \frac{dF}{dX}, \quad \frac{1}{F} \frac{dF}{dX}.$$

Ненормализованная чувствительность (часто называемая коэффициентом влияния) определяется выражением

$$A = A_X^F = \frac{dF}{dX}.$$

Наиболее простым для реализации в программах моделирования и наиболее универсальным по отношению к различным применениям методом расчета чувствительности является *метод малых приращений*.

Основное достоинство метода малых приращений – универсальность, так как он используется для расчета коэффициентов чувствительности любых выходных параметров в любых режимах при изменении любого внутреннего параметра схемы X_i . Кроме того, реализация метода в программном обеспечении очень проста.

Однако указанный метод имеет и серьезные недостатки, ограничивающие его применение. Один из них – очень большие вычислительные затраты: $(n + 1)$ обращений к программе анализа. Более существенным недостатком считается его низкая точность вследствие замены частной производной малыми приращениями. Попытки уменьшения ΔX_i могут привести к возникновению очень малой величины ΔF , что из-за погрешности анализа (как разности очень близких значений), возможно, приведет к неверным результатам.

С целью уменьшения вычислительных затрат и повышения точности расчетов для анализа электронных схем был разработан специальный метод расчета коэффициентов чувствительности – метод присоединенной схемы [13; 27; 43], с помощью которого за однократный расчет двух схем (исходной и присоединенной) можно рассчитать коэффициенты чувствительности одной выходной функции F по всем внутренним параметрам схемы X_i .

Метод присоединенной схемы. В основе метода лежит теорема Теллегена, доказывающая, что алгебраическая сумма произведений токов и напряжений всех ветвей схемы равна нулю:

$$\sum_k i_k(t) u_k(t) = 0, \quad (5.1)$$

где суммирование выполняется по всем ветвям (k) -схемы.

Важное следствие теоремы Теллегена состоит в том, что соотношение (5.1) справедливо и в том случае, если напряжения ветвей или токи принадлежат разным схемам, названным *исходной* и *присоединенной*, но имеющим одинаковую топологию. Выражение (5.1) справедливо и для приращений токов или напряжений в исходной схеме, т. е.

$$\begin{aligned} \sum_k [u_k(t) + \Delta u_k(t)] \hat{i}_k(t) &= 0; \\ \sum_k [i_k(t) + \Delta i_k(t)] \hat{u}_k(t) &= 0, \end{aligned}$$

или

$$\sum_k \Delta u_k(t) \hat{i}_k(t) = 0; \quad \sum_k \Delta i_k(t) \hat{u}_k(t) = 0, \quad (5.2)$$

где i_k и u_k – ток и напряжение k -й ветви исходной схемы; \hat{i}_k и \hat{u}_k – ток и напряжение соответствующей ветви присоединенной схемы; Δi_k и Δu_k – приращения тока и напряжения ветви k в исходной схеме.

Из выражения (5.2) следует, что

$$\sum_k [\Delta u_k(t) \widehat{i}_k(t) - \Delta i_k(t) \widehat{u}_k(t)] = 0. \quad (5.3)$$

Соотношение (5.3) – основное для построения присоединенной схемы и получения выражений для расчета чувствительности. Так как теорема Теллегена предоставляет определенную свободу в выборе компонентов присоединенной схемы (единственное ограничение – топология исходной и присоединенной схем должна быть одинаковой), то с помощью их подбора можно добиться наиболее простых соотношений.

Для того чтобы получить алгоритм формирования присоединенной схемы и формулы для расчета коэффициентов чувствительности, необходимо подробно проанализировать отдельные составляющие выражения (5.3), соответствующие различным типам компонентов схемы. Рассмотрим построение присоединенной схемы в частотной области в режиме малого сигнала, при котором уравнение (5.3) для комплексных значений напряжений и токов ветвей будет иметь вид

$$\sum_k [\Delta u_k \widehat{i}_k - \Delta i_k \widehat{u}_k] = 0. \quad (5.4)$$

Проанализируем составляющую суммы (5.4), соответствующую ветви выхода схемы, при условии, что выходной переменной будет напряжение, т. е. $U_{\text{ВЫХ}} (I_{\text{ВЫХ}} = 0)$. В этом случае выходная цепь

$$N_{\text{ВЫХ}} = \Delta U_{\text{ВЫХ}} \widehat{I}_{\text{ВЫХ}} - \Delta I_{\text{ВЫХ}} \widehat{U}_{\text{ВЫХ}} = \Delta U_{\text{ВЫХ}} \widehat{I}_{\text{ВЫХ}}. \quad (5.5)$$

Если при построении присоединенной схемы заменить разомкнутую ветвь исходной схемы ($I_{\text{ВЫХ}} = 0$) источником тока ($\widehat{I}_{\text{ВЫХ}} = 1$), то выражение (5.5) примет следующий вид (рис. 5.1).

$$N_{\text{ВЫХ}} = \Delta U_{\text{ВЫХ}} \quad (5.6)$$

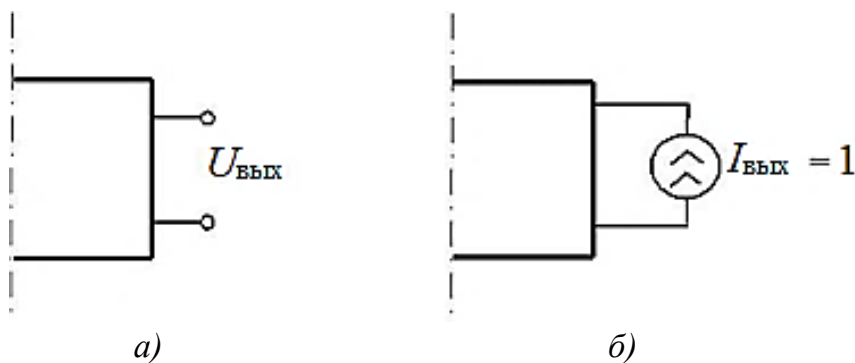


Рис. 5.1. Цепь выхода схемы: а – исходная; б – присоединенная

Рассмотрим другую произвольную ветвь, например, содержащую сопротивление, описываемое компонентным уравнением $U_R = RI_R$. Тогда соответствующая составляющая суммы (5.4) при подстановке приращения

$$\Delta U_R = R\Delta I_R + I_R\Delta R$$

будет иметь вид

$$\begin{aligned} N_R &= \Delta U_R \widehat{I}_R - \Delta I_R \widehat{U}_R = (R\Delta I_R + I_R\Delta R)\widehat{I}_R - \Delta I_R \widehat{U}_R = \\ &= \Delta I_R (R\widehat{I}_R - \widehat{U}_R) + I_R\Delta R \widehat{I}_R. \end{aligned}$$

Выражение упростится, если $(R\widehat{I}_R - \widehat{U}_R)$ приравнять нулю. Этого можно добиться, если в присоединенной схеме в данную ветвь ввести сопротивление, описываемое компонентным уравнением

$$\widehat{U}_R = R\widehat{I}_R,$$

что соответствует рис. 5.2.

$$N_R = I_R\Delta R \widehat{I}_R. \quad (5.7)$$

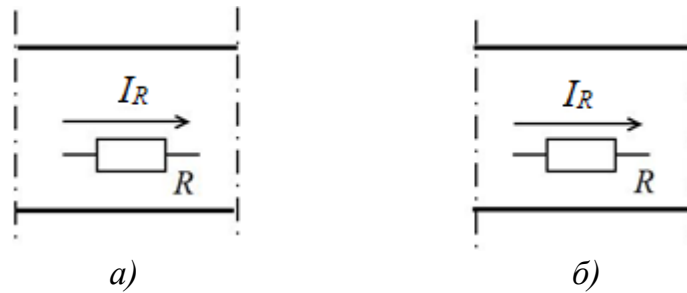


Рис. 5.2. Цепь резистора схемы: а – исходная схема; б – присоединенная схема

Аналогично для всех остальных типов компонентов исходной схемы можно определить уравнения соответствующих им компонентов присоединенной схемы. Так, компонентным уравнениям исходной схемы для проводимости, емкости, индуктивности и крутизны нелинейной зависимости

$$I_G = GU_G; \quad I_C = j\omega CU_C; \quad U_L = j\omega LI_L; \quad I_1 = g_m U_2 \quad (5.8)$$

будут соответствовать следующие уравнения присоединенной схемы:

$$\widehat{I}_G = G\widehat{U}_G; \quad \widehat{I}_C = j\omega C\widehat{U}_C; \quad \widehat{U}_L = j\omega L\widehat{I}_L; \quad \widehat{I}_2 = g_m \widehat{U}_1. \quad (5.9)$$

По выражениям (5.6) и (5.7) можно рассчитать чувствительность изменения выходного напряжения к изменению сопротивления ветви:

$$\begin{aligned} N_{\text{вых}} + N_R &= \Delta U_{\text{вых}} + I_R\Delta R \widehat{I}_R = 0 \quad \text{или} \\ \Delta U_{\text{вых}}/\Delta R &= -I_R \widehat{I}_R. \end{aligned} \quad (5.10)$$

При малых вариациях переменных легко получить выражения для расчета чувствительности сопротивления

$$\partial U_{\text{вых}}/\partial R = -I_R \hat{I}_R. \quad (5.11)$$

Таким образом, коэффициенты чувствительности напряжения на выходе схемы относительно любого сопротивления будут определяться произведением токов в исходной и присоединенной схемах на этих сопротивлениях. Аналогично можно получить выражения для расчета чувствительностей по остальным компонентам схемы:

$$\partial U_{\text{вых}}/\partial C = j\omega U_C \hat{U}_C; \quad \partial U_{\text{вых}}/\partial L = -j\omega I_L \hat{I}_L; \quad \partial U_{\text{вых}}/\partial g_m = U_2 \hat{U}_1. \quad (5.12)$$

Алгоритм расчета чувствительности. Для получения правил формирования присоединенной схемы необходимо сравнить компонентные уравнения основных типов элементов (5.8) и (5.9). Из этих выражений видно, что все пассивные компоненты исходной схемы не меняют своего положения в присоединенной схеме, меняются местами только узлы источников тока, управляемого напряжением. Эти изменения можно учесть транспонированием матричных уравнений исходной схемы. Таким образом, для формирования присоединенной схемы в общем виде по уравнениям исходной схемы

$$YV = I \quad (5.13)$$

необходимо транспонировать матрицу узловых проводимостей Y^T , обнулить вектор задающих токов I , а в выходные узлы занести единичный источник тока

$$Y^T \hat{U} = \hat{I}. \quad (5.14)$$

Решение уравнений (5.13) и (5.14) дает полную информацию для расчета чувствительности выходного напряжения (а при единичном источнике в (5.13) – коэффициента передачи) по всем элементам схемы (5.11) и (5.12).

При решении уравнений (5.13) и (5.14) методом LU -факторизации может быть получен дополнительный выигрыш по вычислительным затратам, так как векторы I и \hat{I} сильно разрежены, а транспонирование матрицы Y^T требует транспонирования только L и U подматриц и выполнения прямой и обратной подстановок.

Расчет чувствительности для линеаризованных схем. Для программ моделирования электронных схем приведенная выше методика расчета коэффициентов чувствительности может быть применена только для чисто линейных схем. Для линеаризованных схем в про-

граммах моделирования перед расчетом малосигнального режима в частотной области выполняется анализ статического режима нелинейной схемы, в котором определяются рабочие точки активных элементов (АЭ), по результатам которых выполняется линеаризация их зависимостей. Очевидно, что изменение сопротивлений схемы окажет влияние на режимы работы (рабочие точки) АЭ, что, в свою очередь, должно повлиять и на частотные характеристики малого сигнала. Поэтому задачу расчета чувствительности по параметрам сопротивлений схемы необходимо рассматривать как задачу определения полной производной сложной функции. Алгоритм расчета чувствительности в этом случае будет охватывать анализ статического и малосигнального режимов. Полная производная выходной функции, например напряжения, по параметрам элементов R_i будет определяться следующим выражением:

$$\frac{dU_{\text{ВЫХ}}}{dR_i} = \frac{\partial U_{\text{ВЫХ}}}{\partial R_i} + \sum_{\text{АЭ}} \sum_P \left(\frac{\partial U_{\text{ВЫХ}}}{\partial P} \frac{dP}{dR_i} \right). \quad (5.15)$$

Здесь суммирование выполняется по всем активным элементам схемы АЭ и малосигнальным параметрам P этих АЭ. Производные $\frac{\partial U_{\text{ВЫХ}}}{\partial R_i}$ и $\frac{\partial U_{\text{ВЫХ}}}{\partial P}$ – чувствительности схемы в режиме малого сигнала – определяются путем решения уравнений (5.13) и (5.14) по выражениям (5.11) и (5.12), в которых под параметрами P понимаются параметры эквивалентной схемы АЭ (R, C, g_m и др.).

Зависимость малосигнальных параметров P от режима работы АЭ и сопротивлений схемы можно также представить в виде производной сложной функции

$$\frac{dP}{dR_i} = \sum_n \frac{\partial P}{\partial V_0} \frac{\partial V_0}{\partial R_i},$$

где суммирование осуществляется по числу переходов АЭ (n); V_0 – напряжение по постоянному току на переходе АЭ. Значения $\frac{\partial P}{\partial V_0}$ легко вычисляются дифференцированием аналитических зависимостей параметров моделей АЭ по напряжению на переходе и программируются в виде непосредственных зависимостей производных от V_0 . Значения $\frac{\partial V_0}{\partial R_i}$ рассчитываются путем построения и решения присоединенной схемы в статическом режиме, обычно одновременно с расчетом режимов в АЭ.

Полный алгоритм расчета чувствительности выходной функции в режиме малого сигнала в системе моделирования электронных устройств можно представить в виде последовательности шагов:

Шаг 1. Расчет статического режима схемы. Определение рабочих точек и других характеристик по постоянному току.

Шаг 2. Если в списке варьируемых параметров при расчете чувствительности отсутствуют резистивные элементы, то переход к шагу 6.

Шаг 3. Формирование и решение уравнений присоединенной схемы по постоянному току для всех переходов АЭ. Вычисление значений $\frac{\partial V_0}{\partial R_i}$.

Шаг 4. Вычисление значений производных $\frac{\partial P}{\partial V_0}$ и $\frac{dP}{dR_i}$.

Шаг 5. Организация цикла по частоте малосигнального режима.

Шаг 6. Формирование и решение уравнений исходной схемы в режиме малого сигнала, расчет частотных и других характеристик.

Шаг 7. Формирование и решение уравнений присоединенной схемы в режиме малого сигнала. Расчет чувствительности с использованием выражений (5.11), (5.12) и (5.15).

5.2. Расчет собственных шумов схемы

Метод присоединенной схемы дает простой и эффективный алгоритм расчета функций чувствительности электронных схем. Но очень важные свойства присоединенной схемы могут быть использованы и для эффективного расчета ряда других характеристик электронных устройств. При этом более наглядна «физическая» интерпретация правил построения и применения присоединенной схемы. Так, если представить выражение (5.10) в несколько преобразованном виде

$$\Delta U_{\text{вых}} = (-\Delta R_i I_{Ri}) \widehat{I}_{Ri} = \Delta U_{Ri} \widehat{I}_{Ri}, \quad (5.16)$$

то небольшому изменению напряжения на резисторе ΔU_{Ri} (как некоторый генератор) будет соответствовать небольшое изменение в напряжении на выходе $\Delta U_{\text{вых}}$ через величину \widehat{I}_{Ri} . Но это полностью соответствует расчету изменения выходного напряжения через передаточную функцию

$$\Delta U_{\text{вых}} = \Delta U_{Ri} K_{i\text{-вых}}, \quad (5.17)$$

где частную передаточную функцию $K_{i-\text{ВЫХ}}$ определяют при выключенном входном источнике отношением выходной переменной $U_{\text{ВЫХ}}$ к параметру некоторого источника ΔU_{Ri} , введенному в ветвь компонента R_i . Из сравнения (5.16) и (5.17) можно определить, что \widehat{T}_{Ri} играет роль частного коэффициента передачи от i -го элемента к выходу схемы, что согласуется со свойством теории чувствительности для линейных схем, где коэффициенты чувствительности могут быть определены через произведение частных передаточных функций [38]

$$\frac{\partial U_{\text{ВЫХ}}}{\partial X_i} = K_{\text{ВХ-}i} K_{i-\text{ВЫХ}},$$

где $K_{\text{ВХ-}i}$ характеризует передачу сигнала от входа схемы к выделенному компоненту X_i , а $K_{i-\text{ВЫХ}}$ – передачу сигнала от выделенного компонента к выходу схемы (рис. 5.3).

Равенство $\widehat{T}_{Ri} = K_{i-\text{ВЫХ}}$ можно объяснить и тем, что во время транспонирования матрицы узловых проводимостей схему мы практически «переворачиваем», а физически меняем местами узлы в управляемых источниках. Подавая теперь уже на вход «перевернутой» схемы единичный сигнал (т. е. в выходные узлы исходной схемы), за один анализ можно определить напряжения и токи (коэффициенты передачи при единичном входном сигнале) на любом компоненте схемы путем решения системы линейных матричных уравнений.

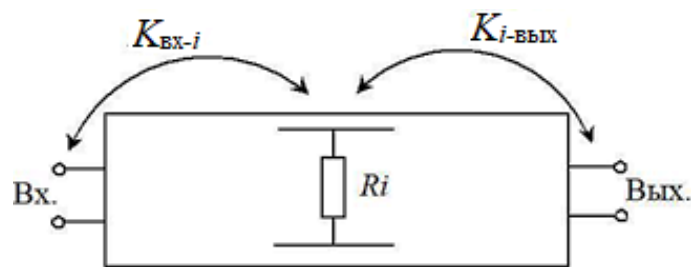


Рис. 5.3. Иллюстрация передаточных функций

Таким образом, решение уравнений присоединенной схемы (5.14) дает эффективный способ одновременного вычисления всех частных коэффициентов передачи и может быть использовано в различных приложениях, базирующихся на линейном представлении схем. Одним из таких приложений является анализ собственных шумов электронных схем [43].

Из-за очень малых амплитуд токов и напряжений, рассматриваемых при анализе шумов, этот анализ всегда опирается на линейные модели схемы. Для большинства обычных элементов, таких как резисторы, диоды и транзисторы, источники шума моделируются в виде независимых источников тока, каждый из которых включен параллельно с нешумящим резистором или переходом АЭ.

Дробовый, тепловой и фликкер-шум – это три наиболее важных составляющих шума в электронных цепях. Первые два моделируются с помощью физических параметров цепи, в то время как для третьего требуются эмпирические данные.

Источники дробового шума для резисторов схемы и для омических сопротивлений моделей АЭ вычисляются по известной формуле Найквиста [43]

$$I_{Ri}^2 = 4kT(1/Ri)\Delta f,$$

где k – постоянная Больцмана; T – температура, К; Ri – номинал сопротивления, Ом; Δf – ширина полосы частот, Гц.

Дробовый шум в моделях диода и биполярного транзистора представляется источником тока, описываемым зависимостью

$$I_{Ii}^2 = 2qIi\Delta f,$$

где q – заряд электрона; Ii – ток перехода АЭ; для полевых транзисторов зависимостью

$$I_{Ii}^2 = 2/3 \cdot 4kTIi\Delta f.$$

Цель анализа собственных шумов схемы – определение вкладов в выходной сигнал со стороны каждого источника шума:

$$U_{\text{ВЫХ}i}^2 = I_i^2 K_{i-\text{ВЫХ}}^2. \quad (5.18)$$

Из-за случайного характера источников шума обычно оценивается среднее квадратическое значение шума на выходе. Тогда суммарная амплитуда шума на выходе схемы для некоррелированных источников шума определяется следующим образом:

$$U_{\text{ш. ВЫХ}} = \sqrt{\sum_{i=1}^n U_{\text{ВЫХ}i}^2}.$$

Вычисление вкладов источников шума по выражению (5.18) требует вычисления передаточных функций (имеющих размерность полного сопротивления) от каждого источника шума на выход схемы. Используя свойства присоединенной схемы, все передаточные функции можно эффективно определить за один анализ присоединенной схемы. В этом случае выражение (5.18) может быть переписано в виде

$$U_{\text{вых}i}^2 = I_i^2 \widehat{U}_i^2,$$

где \widehat{U}_i – напряжение в присоединенной схеме на i -м элементе источника шума. В сложных цепях число источников шума может быть очень большим, поэтому использование свойств присоединенной схемы дает явный выигрыш в количестве вычислений.

5.3. Расчет нелинейных искажений в режиме малого сигнала

Основными причинами возникновения *нелинейных искажений* (НИ) в электронных схемах считаются нелинейность характеристик активных приборов и других нелинейных элементов, нестабильность источников питания, наличие мешающих воздействий и наводок. Как правило, появление НИ связано с общим свойством нелинейных цепей порождать новые колебания в спектрах проходящих через них сигналов и помех.

Рассмотрим некоторые основные виды НИ [48]:

1. *Гармонические*, выражающиеся в появлении кратных гармонических составляющих входного воздействия. Для расчета гармонических искажений часто используют следующие критерии:

– *коэффициент гармоник*, который при синусоидальном входном воздействии определяется отношением среднеквадратичной суммы амплитуд высших гармоник к амплитуде основной частоты на выходе схемы:

$$K_{\Gamma} = \frac{\sqrt{A_2^2 + A_3^2 + \dots}}{A_1} 100 \%,$$

где A_i – амплитуды i -й гармоники выходного сигнала;

– *коэффициенты гармоник по отдельным составляющим*

$$K_{\Gamma n} = (A_n/A_1)100 \%.$$

2. *Интермодуляционные* (взаимной модуляции), выражающиеся во взаимодействии в схеме двух и более сигналов или помех, вызывающих появление новых спектральных составляющих с частотами, равными сумме или разности частот входных сигналов или их гармоник. Для оценки интермодуляционных искажений чаще всего используют бигармоническое воздействие, содержащее колебания не кратных частот f_1, f_2 . В этом случае на выходе схемы возникают интермодуляционные колебания с частотами $f = mf_1 + nf_2$, где m и n – целые положи-

тельные и отрицательные числа. Под порядком продукта взаимной модуляции понимают абсолютное значение суммы коэффициентов $|m| + |n|$. Так, коэффициент интермодуляции второго порядка оценивается соотношением

$$K_{им2} = A_{11}/A_1 \cdot 100 \%,$$

а третьего порядка

$$K_{им3} = A_{21}/A_1 \cdot 100 \%,$$

где A_{11} и A_{21} – амплитуды интермодуляционных составляющих на частотах $(f_1 \pm f_2)$ и $(2f_1 \pm f_2)$ соответственно.

Для оценки нелинейных искажений в режиме малого сигнала наибольшее распространение в программах АСхП нашел метод на основе аппарата *функциональных рядов Вольтерра* [21; 43; 48]. Функциональный ряд Вольтерра (ФРВ) представляет собой обобщение интеграла Дюамеля (интеграл свертки для линейных устройств) на случай нелинейной инерционной цепи и связывает входной и выходной сигналы в виде

$$\begin{aligned} y(t) &= \sum_{n=1}^{\infty} y_n(t) = \int_{-\infty}^{\infty} h_1(\tau_1)x(t-\tau_1)d\tau_1 + \\ &+ \int \int h_2(\tau_1, \tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1d\tau_2 + \\ &+ \dots = \sum_{n=1}^{\infty} \int \dots \int h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t-\tau_i)d\tau_i, \end{aligned}$$

где $x(t)$ – входное воздействие; $h_n(\tau_1, \dots, \tau_n)$ – ядра Вольтерра n -го порядка. Если $n = 1$, то $h_1(\tau_1) = h(\tau)$ – обычная импульсная характеристика. Добавление членов ряда при $n > 1$ соответствует учету нелинейных свойств. Если амплитуда входного сигнала или нелинейности схемы не слишком велики, то достаточно учесть лишь несколько первых членов ряда.

Преобразование Фурье от $h_n(\tau_1, \dots, \tau_n)$

$$H_n(f_1, \dots, f_n) = \int \dots \int h_n(\tau_1, \dots, \tau_n) \exp[-j2\pi(f_1\tau_1 + \dots + f_n\tau_n)] d\tau_1 \dots d\tau_n$$

называют *передаточной функцией ряда Вольтерра n -го порядка*. Связь входного и выходного сигналов с помощью передаточной функции ряда Вольтерра будет иметь вид

$$y(f_1, \dots, f_n) = \sum_{n=1}^{\infty} H_n(f_1, \dots, f_n) \prod_{i=1}^n x(f_i). \quad (5.19)$$

Приведем некоторые составляющие выходного сигнала, определяемые через передаточные функции ряда Вольтерра, при гармоническом воздействии $x(t) = U_{\text{вх}} \cos(2\pi ft)$.

Так, постоянная составляющая напряжения на выходе устройства будет иметь вид

$$U_{\text{вых0}} = 0,5 U_{\text{вх}}^2 \operatorname{Re}[H_2(f, -f)],$$

составляющая на первой гармонике

$$U_{\text{вых1}} = U_{\text{вх}} |H_1(f)| \cos(2\pi ft);$$

составляющая на второй гармонике

$$U_{\text{вых2}} = 0,5 U_{\text{вх}}^2 |H_2(f, f)| \cos(2\pi ft).$$

При бигармоническом входном воздействии

$$x(t) = U_{\text{вх1}} \cos(2\pi f_1 t) + U_{\text{вх2}} \cos(2\pi f_2 t),$$

отклик на частоте $(f_1 \pm f_2)$ будет иметь вид

$$U_{\text{вых11}} = U_{\text{вх1}} U_{\text{вх2}} |H_2(f_1, \pm f_2)| \cos 2\pi(f_1 \pm f_2)t,$$

а на частоте $(2f_1 \pm f_2)$

$$U_{\text{вых11}} = U_{\text{вх1}}^2 U_{\text{вх2}} |H_3(f_1, f_1, \pm f_2)| \cos 2\pi(2f_1 \pm f_2)t.$$

По значениям амплитуд составляющих отклика нелинейной цепи можно определить все интересующие коэффициенты НИ, например:

$$K_{\Gamma 3} = 0,25 U_{\text{вх}}^2 \frac{|H_3(f, f, f)|}{|H_1(f)|},$$

$$K_{\Gamma} = \sqrt{K_{\Gamma 2}^2 + K_{\Gamma 3}^2 + \dots};$$

$$K_{\text{им2}} = U_{\text{вх2}} \frac{|H_2(f_1, \pm f_2)|}{|H_1(f_1)|},$$

$$K_{\text{им3}} = 0,75 U_{\text{вх1}} U_{\text{вх2}} \frac{|H_3(f_1, f_1, \pm f_2)|}{|H_1(f_1)|}.$$

Так как передаточная функция ряда Вольтерра первого порядка связывает отклик и воздействие в виде

$$y(f) = H_1(f)x(f)$$

и соответствует обычному линейному преобразованию, то $H_1(f)$ можно легко найти из решения системы линейных уравнений в режиме малого сигнала

$$YV = I,$$

если в векторе I примем единичным источник входного воздействия. В этом случае систему уравнений для определения вектора передаточных функций первого порядка можно представить в следующем виде:

$$Y(f)H_1(f) = I. \quad (5.20)$$

Степенной характер ряда Вольтерра (5.19) требует и степенного представления нелинейностей схемы. Так, зависимость, например, для резистивной нелинейности, может быть представлена в виде ряда Тейлора относительно рабочих точек прибора:

$$I_R = \left. \frac{\partial I_R}{\partial U_R} \right|_{U_0} U_R + \frac{1}{2} \left. \frac{\partial^2 I_R}{\partial U_R^2} \right|_{U_0} U_R^2 + \\ + \frac{1}{6} \left. \frac{\partial^3 I_R}{\partial U_R^3} \right|_{U_0} U_R^3 + \dots = g_R U_R + g'_R U_R^2 + g''_R U_R^3 + \dots \quad (5.21)$$

Тогда при учете только линейного члена ряда (5.21) и передаточной функции ряда Вольтерра первого порядка (5.19) получаем обычный малосигнальный режим, что и отражено уравнением (5.20), где матрица проводимостей $Y(f)$ содержит линейные элементы схемы и линейные члены разложения в ряд Тейлора.

Передаточную функцию второго порядка можно найти, если в уравнениях цепи все узловые потенциалы выразить через передаточные функции ряда Вольтерра второго порядка, а в нелинейных зависимостях учесть первые два члена ряда Тейлора. Опуская промежуточные выкладки, укажем, что матричные уравнения в этом случае будут иметь вид

$$Y(f_1 + f_2) H_2(f_1, f_2) = I_2, \quad (5.22)$$

где $Y(f_1 + f_2)$ – матрица узловых проводимостей на частоте $f_1 + f_2$, матрица Y содержит линейные элементы схемы и линейные члены разложения в ряд Тейлора; $H_2(f_1, f_2)$ – вектор передаточных функций Вольтерра второго порядка; I_2 – вектор так называемых нелинейных токов второго порядка. Составляющие вектора I_2 определяются для каждого нелинейного элемента схемы в виде источников тока, подключенных параллельно нелинейному элементу. Например, для резистивной нелинейности источник нелинейного тока второго порядка будет определяться соотношением

$$I_{2R} = g'_R H_{1R}(f_1) H_{1R}(f_2), \quad (5.23)$$

где $H_{1R}(f)$ – передаточная функция ряда Вольтерра первого порядка относительно узлов входа схемы и узлов подключения нелинейного элемента, значение $H_{1R}(f)$ может быть легко выбрано из вектора $H_1(f)$, полученного на предыдущем шаге; g'_R – коэффициент разложения нелинейной зависимости в ряд Тейлора (5.21).

Уравнения для определения передаточных функций третьего порядка получаются при учете трех членов рядов Тейлора и Вольтерра:

$$Y(f_1 + f_2 + f_3)H_3(f_1, f_2, f_3) = I_3, \quad (5.24)$$

где $Y(f_1 + f_2 + f_3)$ – матрица проводимостей на частоте $f_1 + f_2 + f_3$; $H_3(f_1, f_2, f_3)$ – вектор передаточных функций третьего порядка; I_3 – вектор нелинейного тока третьего порядка. Элементами вектора I_3 будут составляющие источников нелинейного тока каждой нелинейности схемы, например для резистивной нелинейности

$$I_{3R} = g_R'' H_{1R}(f_1) \cdot H_{1R}(f_2) \cdot H_{1R}(f_3) + 2g_R' \overline{H_{1R}(f_1) \cdot H_{2R}(f_2, f_3)}. \quad (5.25)$$

В формуле (5.25) черта сверху означает симметризацию передаточных функций относительно набора частот

$$\overline{H_{1R}(f_1) \cdot H_{2R}(f_2, f_3)} = 1/3 [H_{1R}(f_1) \cdot H_{2R}(f_2, f_3) + H_{1R}(f_2) \cdot H_{2R}(f_1, f_3) + H_{1R}(f_3) \cdot H_{2R}(f_1, f_2)].$$

Продолжая аналогичные рассуждения и построения, можно показать, что передаточные функции любого порядка будут определяться из решения уравнений вида

$$Y(f_1 + \dots + f_n)H_n(f_1, \dots, f_n) = I_n. \quad (5.26)$$

Таким образом, передаточные функции ряда Вольтерра любого порядка определяются из последовательного решения системы линейных уравнений одного и того же вида (5.26). Так как уравнения (5.26) полностью соответствуют уравнению малосигнального режима, то могут быть использованы известные и хорошо отработанные алгоритмы программ АСхП. Дополнительные задачи – определение коэффициентов разложения нелинейных зависимостей схемы в ряд Тейлора и формирование вектора нелинейных токов для каждой системы уравнений.

Алгоритм расчета НИ в программах АСхП можно представить в виде последовательности шагов [21]:

Шаг 1. Расчет статического режима схемы. Определение рабочих точек активных и нелинейных элементов. Расчет коэффициентов разложения нелинейных зависимостей в ряд Тейлора. Коэффициенты ряда Тейлора находятся по аналитическим выражениям.

Шаг 2. Формирование и решение системы линейных уравнений (5.20). Эти уравнения решаются на всех частотах, необходимых в дальнейшем для формирования векторов нелинейного тока. Например, для расчета только гармонических искажений это будет частота сигнала f_1 ,

в случае расчета интермодуляционных искажений частотами анализа будут f_1 и f_2 .

Шаг 3. По выделенным из вектора $H_1(f)$ передаточным функциям относительно нелинейных элементов и по коэффициентам ряда Тейлора формирование вектора нелинейного тока I_2 (используя выражения типа (5.23)).

Шаг 4. Формирование и решение системы линейных уравнений (5.22). Эти уравнения решаются на всех частотах ($f_1 \pm f_2$).

Шаг 5. Формирование вектора нелинейного тока третьего порядка I_3 .

Шаг 6. Решение уравнений (5.24).

Шаг 7. Расчет критериев оценки НИ.

5.4. Расчет спектра в установившемся режиме

В п. 5.3 рассматривался метод анализа НИ в режиме малого сигнала. Его основное ограничение – малый уровень входных воздействий, при которых данный подход справедлив. Другие подходы для анализа НИ электронных схем – методы моделирования при больших уровнях сигналов, один из которых – метод расчета спектра отклика на выходе устройства на заданное тестовое входное гармоническое воздействие в установившемся режиме во временной области. Отклик на тестовый сигнал может быть разложен в ряд Фурье с помощью алгоритмов дискретного или быстрого преобразования Фурье (ДПФ или БПФ) с целью определения основных спектральных составляющих (рис. 5.4). Подход очень прост и широко распространен, так как позволяет использовать для расчета хорошо отработанные алгоритмы программ АСхП, дополненные алгоритмами Фурье-преобразования. Знание спектрального состава входного воздействия и отклика позволяет вычислить все основные критерии оценки НИ (п. 5.3).

Дискретное преобразование Фурье описывается соотношением

$$F(\omega) = \sum_{k=0}^{N-1} x(t_k) e^{-j\omega t_k}, \quad (5.27)$$

где $t_k = k\Delta t$; $\omega = 2\pi p/T$; p – безразмерная частота; $T = N\Delta t$.

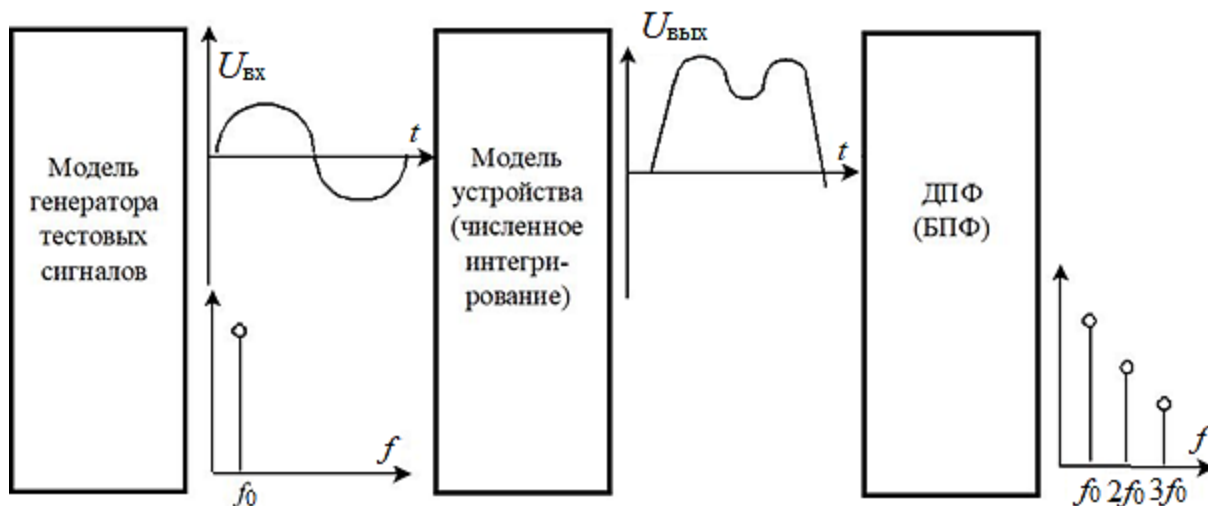


Рис. 5.4. Иллюстрация к расчету спектра

При расчете спектра с помощью ДПФ выбирают равноотстоящие точки t_k на интервале $(0, T)$ и по выражению (5.27) рассчитывают спектральные составляющие $F(0), F(1), \dots$. Непосредственное применение ДПФ требует больших затрат времени, если используется часто и для большого числа временных отсчетов. Поэтому при реализации на ЭВМ наибольшее применение получил специальный алгоритм, названный *быстрым преобразованием Фурье* (БПФ) [10].

5.5. Моделирование установившегося режима

При воздействии сложных входных сигналов наиболее общим методом анализа аналоговых схем считается анализ во временной области. Особенность работы большинства аналоговых схем – наличие в них резонансных свойств с высокой добротностью [49], что приводит к длительному затуханию переходных процессов в схеме. При этом время переходного процесса может значительно превышать длительность периода установившегося режима (рис. 5.5), а время установления стационарного режима в таких схемах может содержать тысячи периодов собственных колебаний. Для решения этой проблемы используются специальные алгоритмы поиска начальных условий для расчета установившегося режима высокодобротных схем [50].

Моделирование схем методом численного интегрирования при нулевых начальных условиях $V(t) = V(0)$ приводит к появлению в отклике переходной составляющей (штриховая линия на рис. 5.5).

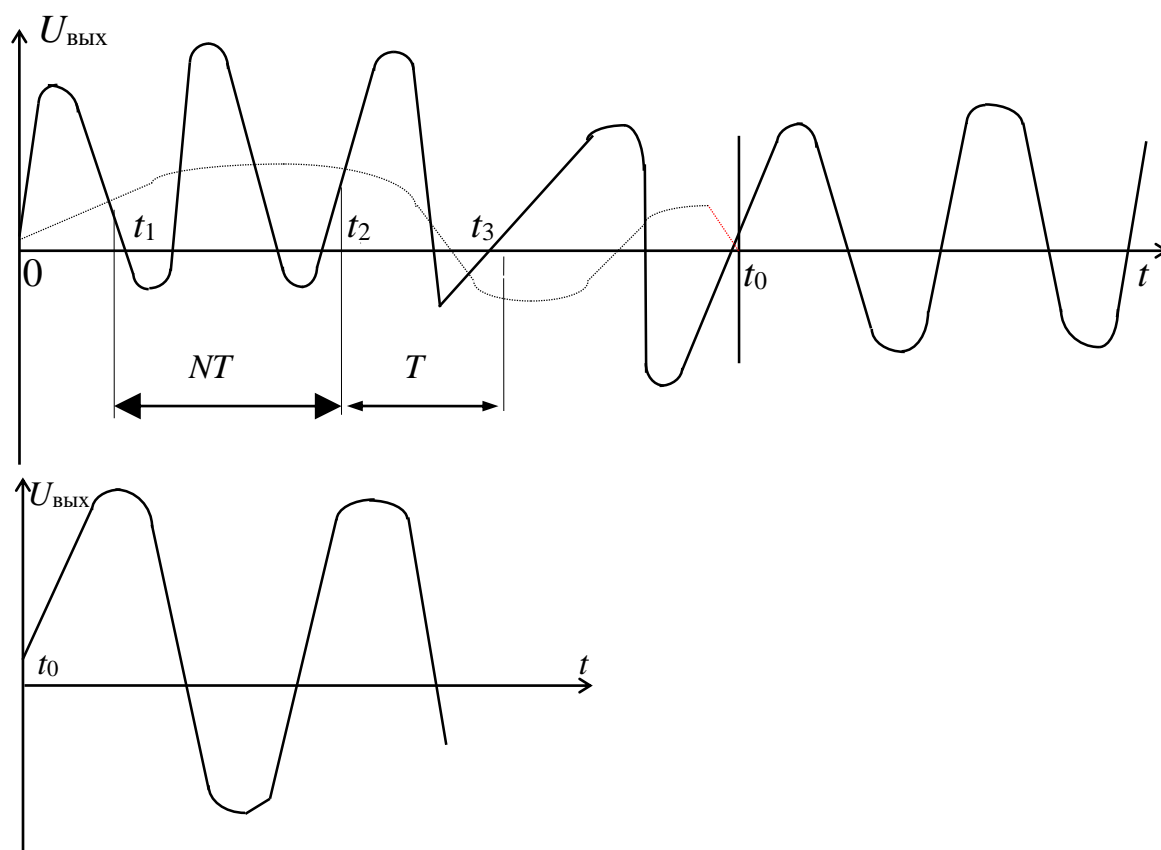


Рис. 5.5. Иллюстрации к методу установившегося режима

Предположим, что в момент времени t_0 переходная составляющая будет равна нулю. Если при интегрировании уравнений модели схемы взять за начальные условия вектор $V_0 = V(t_0)$, то теперь отклик будет содержать только основные колебания установившегося режима. Таким образом, задача поисковых методов – нахождение вектора начальных условий V_0 , при котором процесс $V(t)$ не будет иметь переходной составляющей. Для периодических сигналов это условие записывают в виде

$$V(t_0) = V(t_0 + T),$$

где T – период колебания.

Алгоритм нахождения установившегося режима поисковым методом можно представить в виде решения двух задач: интегрирования модели схемы в виде СНДУ, результатом чего является некоторая функция $F(V_0)$, и решения нелинейных алгебраических уравнений, т. е. нахождение V_0 из

$$F(V_0) = 0. \quad (5.28)$$

В программах АСхП широко применяется итерационный метод Ньютона, обладающий квадратичной сходимостью:

$$V_0^{n+1} = V_0^n - [I - F'(V_0^n)]^{-1} [V_0^n - F(V_0^n)], \quad (5.29)$$

где I – единичная матрица; $F'(V_0^n)$ – матрица Якоби, имеющая смысл матрицы чувствительности установившегося колебания к изменению начальных условий, $F'(V_0^n) = \frac{\partial F(V_0^n)}{\partial V_0^n}$; V_0 – вектор начальных условий; $F(V_0)$ – вектор результатов интегрирования за период T с начальным условием V_0 .

Алгоритм в этом случае сводится к выполнению следующих шагов:

Шаг 1. Задание или расчет начального условия V_0 .

Шаг 2. Обычное численное интегрирование СНДУ от $t = 0$ до $t = T$ и вычисление $F(V_0)$.

Шаг 3. Вычисление матрицы $F'(V_0^1)$.

Шаг 4. Вычисление нового вектора начальных условий V_0^1 по выражению (5.29) и проверка условия сходимости.

Шаг 5. Если условие сходимости не выполняется, то переход к шагу 2 с новым начальным условием V_0^2 и так далее до сходимости к решению V_0 .

Последняя итерация алгоритма позволяет найти не только V_0 , но и форму колебания в установившемся режиме на интервале $(t_0, t_0 + T)$.

Наиболее трудоемкой операцией в этом алгоритме является вычисление матрицы F' . Один из методов решения этой проблемы – метод моделей чувствительности.

Пусть заданы три момента времени t_1, t_2 и t_3 (см. рис. 5.2), причем $(t_3 - t_2) = T$, $(t_2 - t_1) = NT$, где N – целое число, значение $V(t_1)$ известно. Если N выбрать так, чтобы $V(t)$ изменялось мало на интервале $[t_1, t_3]$, то, выражая первые производные этих функций через конечные разности, получим

$$V' \approx \frac{[V(t_3) - V(t_2)]}{T} \approx [V(t_3) - V(t_1)] / (N + 1)T. \quad (5.30)$$

Учитывая выражение (5.30), можно получить

$$F(V(t_2)) = V(t_3) - V(t_2) - [V(t_3) - V(t_1)] / (N + 1) = 0. \quad (5.31)$$

Система (5.31) решается методом Ньютона относительно $V(t_2)$ аналогично решению системы (5.28), при этом для некоторого значения $V(t_2)$ необходимо численно проинтегрировать систему на интервале $[t_2, t_3]$.

5.6. Метод гармонического баланса

Метод гармонического баланса (ГБ) в настоящее время – один из самых популярных в программах АСхП для моделирования аналоговых устройств в частотной области [10]. Общим недостатком методов во временной области считаются большие вычислительные затраты на поиск установившегося режима избирательных схем, на расчет спектра при воздействии сигналов с сильно различающимися частотами или амплитудами. Частотные методы позволяют непосредственно анализировать схемы устройств в установившемся режиме, минуя переходный.

Суть метода ГБ такова: предполагается, что реакция нелинейной схемы на периодическое воздействие состоит из основной составляющей известной частоты и нескольких доминирующих гармоник. Это предполагаемое решение подставляется в дифференциальные уравнения, описывающие заданную схему. Затем, приравнивая коэффициенты при одинаковых номерах гармоник, получают систему конечных (алгебраических) уравнений. Решение алгебраических уравнений каким-либо известным методом дает искомые амплитуды и фазы гармонических составляющих (в том числе и постоянную составляющую) во всех интересующих исследователя точках схемы [51].

Одна из основных идей, нашедшая применение в большинстве модификаций метода ГБ, – предположение о разделении схемы устройства на линейную и нелинейную части. В этом случае в линейную часть (подсхему, многополюсник) включаются все пассивные компоненты схемы и линейные элементы моделей нелинейных приборов. Надо отметить, что разделение схемы на подсхемы не требует никаких предварительных операций и выполняется в большинстве программ автоматически на уровне составления матричных уравнений.

Разберем основные идеи и соотношения метода ГБ. Можно все нелинейности схемы считать элементами нелинейной подсхемы N , как показано на рис. 5.6, *а*, где элементы векторов $i = (i_1, i_2, \dots, i_n)$, $u = (u_1, u_2, \dots, u_n)$ – токи и напряжения относительно выводов многополюсников. Предполагая независимость элементов подсхемы L от токов или напряжений подсхемы N , представим их в виде двух отдельных схем, как показано на рис. 5.6, *б*. При этом подсхема L дополняется независимыми источниками тока с величинами i_1, i_2, \dots, i_n , а подсхема N –

источниками напряжения u_1, u_2, \dots, u_n . В установившемся режиме результирующая схема должна иметь единые решения для всех токов и напряжений в ветвях выводов подсхем. Например, для векторов тока

$$i_L = i_N.$$

Разделение анализируемой схемы на две подсхемы позволяет независимо решать задачи анализа линейной части методами линейного анализа и нелинейной части – методами анализа нелинейных цепей. Так, для линейной подсхемы при любом значении u_L достаточно просто определяется значение i_L на выводах подсхемы, а для нелинейной подсхемы при заданных зависимостях $i = f(u)$ всегда можно определить значение i_N при любом u_N . Имеющуюся разность (невязку) $i_L - i_N = f$ можно минимизировать одним из методов минимизации, выбирая такое значение $u = u_L = u_N$, при котором $f \rightarrow 0$, или находить корни уравнения $f(u) = 0$, используя любые итерационные методы.

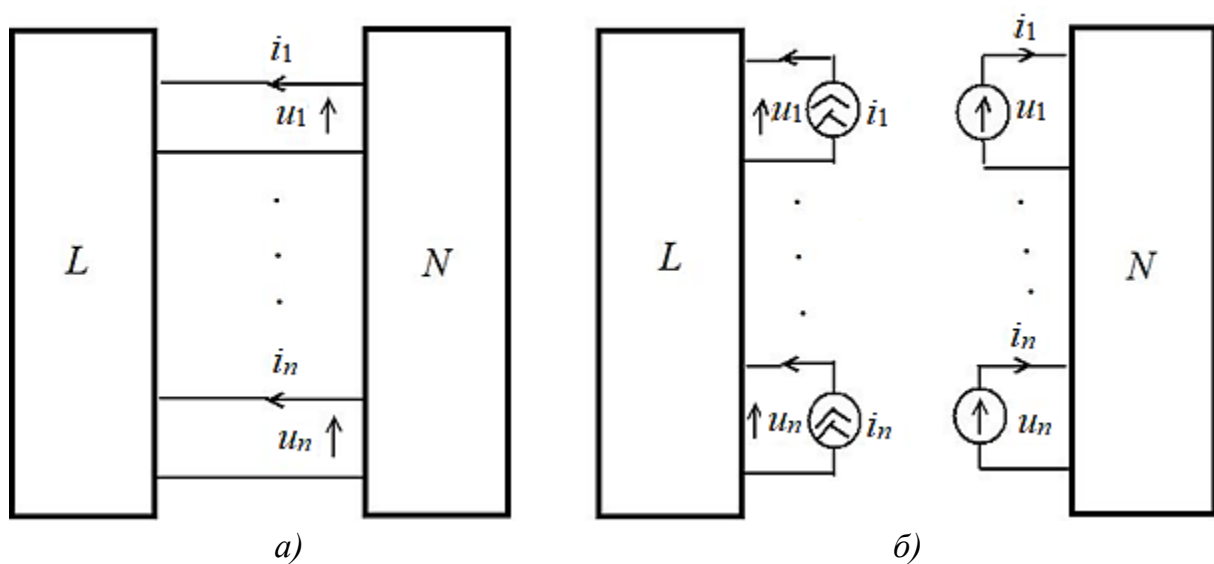


Рис. 5.6. Модель схемы в методе ГБ: а – линейная и нелинейная подсхемы; б – добавление эквивалентных источников

В методах ГБ искомое решение (вектор напряжений на выводах линейного многополюсника) традиционно описывается периодической функцией времени и аппроксимируется каким-либо тригонометрическим полиномом или отрезком ряда Фурье [52]

$$u(t) = V_0 + \sum_{k=1}^N (V_k^c \cos k\omega t + V_k^s \sin k\omega t), \quad (5.32)$$

где V_0 – вектор амплитуд постоянных составляющих, $V_0 = (V_{01}, V_{02}, \dots, V_{0n})$; V_k^c, V_k^s – векторы амплитуд гармоник (косинусные и синусные состав-

ляющие), $V_k^c = (V_{k1}^c, V_{k2}^c, \dots, V_{kn}^c)$, $V_k^s = (V_{k1}^s, V_{k2}^s, \dots, V_{kn}^s)$; n – число выводов линейного многополюсника (примерно равно числу нелинейностей схемы); N – число учитываемых гармоник при аппроксимации рядом Фурье; ω – частота внешнего возбуждения.

Для дальнейшего изложения метода амплитуды (5.32) удобно объединить в один вектор V размерностью $[n \times (2N + 1)]$:

$$V = (V_0, V_1^c, V_1^s, V_2^c, V_2^s, \dots, V_N^c, V_N^s). \quad (5.33)$$

При раздельном моделировании подсхем в методе ГБ можно применить для анализа линейной части численные алгоритмы и программы малосигнального анализа в частотной области. Так, используя алгоритмы решения СЛАУ на всех интересующих частотах $k\omega$

$$Y(k\omega) \times V_L(k\omega) = I_L(k\omega), \quad (5.34)$$

можно достаточно быстро определить токи и напряжения на выводах линейной подсхемы.

Для нелинейной подсхемы, где известными считаются нелинейные зависимости

$$i_N(t) = f[u_N(t)] \quad (5.35)$$

или система дифференциальных уравнений, описывающая нелинейный многополюсник, можно определить форму колебаний токов на выводах путем подстановки $u_N(t)$ в уравнение (5.35) или решением СНДУ.

Запишем уравнение ГБ для вектора невязки

$$F(V) = YV - I(V) = 0. \quad (5.36)$$

Здесь первое слагаемое описывает линейную часть схемы, второе является Фурье-преобразованием нелинейной зависимости (5.35):

$$\Gamma i_N(t) = f[\Gamma^{-1} u_N(t)], \quad (5.37)$$

где Γ и Γ^{-1} – прямое и обратное Фурье-преобразование.

Выражение (5.36) называют основным уравнением ГБ. Решение данной системы нелинейных алгебраических уравнений каким-либо итерационным методом дает все гармонические составляющие искомого вектора V . В программах АСхП часто применяется и метод Ньютона, где

$$V^{i+1} = V^i - J^{-1} \cdot F(V^i).$$

Для преобразования токов и напряжений из частотной области во временную и обратно на каждом шаге итерационного процесса решения используются прямое и обратное преобразования Фурье. Новое значение вектора V находят из решения системы линейных уравнений, что более экономично, чем инвертирование матриц:

$$J \Delta V^{i+1} = -F(V^i). \quad (5.38)$$

Таким образом, алгоритм расчета схем методом ГБ включает выбор начального приближения для вектора V^0 ; умножение матричных уравнений линейной части в частотной области уравнения (5.34) при выбранном значении вектора V ; обратное преобразование Фурье для перехода во временную область; подстановку напряжений в дискретные моменты времени в нелинейные уравнения во временной области уравнения (5.37); прямое преобразование Фурье для определения спектров токов нелинейной подсхемы и итерационного решения системы линейных уравнений ГБ (5.38).

Основными достоинствами метода ГБ считаются:

1. Возможность определения реакции схемы при большом уровне входных сигналов сразу в установившемся режиме, минуя переходный процесс. Это значительно сокращает время моделирования.

2. Независимость времени расчета в методе ГБ от размерности линейной части схемы.

К недостаткам и проблемам метода ГБ относятся:

1. Высокая размерность уравнений ГБ, зависящая от числа нелинейных элементов и числа учитываемых гармоник. Обычно для сокращения размерностей задачи ограничиваются рассмотрением лишь 3...5 гармоник.

2. Проблемы сходимости итерационных методов, особенно при воздействии сигналов с сильно различающимися амплитудами и частотами.

Вопросы для самопроверки

1. Виды коэффициентов чувствительности.
2. Основные соотношения метода присоединенной схемы для расчета чувствительности.
3. Особенности расчета чувствительности с учетом статики.
4. Основная идея расчета собственных шумов.
5. Оценка нелинейных искажений в режиме малого сигнала.
6. Расчет спектра с помощью преобразования Фурье.
7. Основная идея метода поиска установившегося режима.
8. Идея метода гармонического баланса.
9. Алгоритм расчета спектра методом гармонического баланса.

Глава 6. МНОГОУРОВНЕВОЕ МОДЕЛИРОВАНИЕ

6.1. Макромоделирование

Макромодели, используемые в программах АСхП, – это упрощенные представления функциональных узлов, которые с достаточной для конкретного применения точностью отражают статические и динамические характеристики на внешних выводах этих узлов [52].

Макромодель какого-либо функционального узла или подсхемы описывается системой уравнений значительно меньшего размера или существенно более простым алгоритмом расчета выходных переменных по сравнению с развернутым компонентным представлением той же подсхемы, что и обеспечивает уменьшение размера задачи моделирования. Вычислительные затраты при использовании макромоделей можно снизить с помощью некоторого уменьшения точности моделирования. Однако время моделирования при этом уменьшается в несколько раз и даже в несколько десятков раз, тогда как точность – всего лишь на 5...10 %, что в большинстве практических сфер применения вполне допустимо [53].

Наиболее целесообразно представление в виде макромоделей подсхем, топология и свойства которых многократно повторяются. Из таких макромоделей организуют библиотеки в САПР [3], для них решаются задачи подбора параметров и сопряжения с обычными схематехническими описаниями. Введем некоторые определения.

Полная математическая модель (ММ) – модель, в которой фигурируют переменные, характеризующие состояния всех имеющихся межэлементных связей (т. е. состояния всех элементов объекта).

Макромодель – математическая модель, в которой отображаются состояния значительно меньшего числа межэлементных связей, что соответствует более укрупненному описанию объекта. Понятия «полная модель» и «макромодель» относительны и обычно используются для различия двух моделей при разной степени детализации.

Различают:

1) физическое (электрическое) макромоделирование (замена точных, но сложных моделей отдельных фрагментов схем более простыми с отражением только важнейших свойств с помощью схематехнических элементов);

2) информационное (поведенческое) макро моделирование (отражение характеристик вход/выход в виде алгоритма, с помощью уравнений, в виде нейронных сетей и т. п.).

Физические макро модели проще реализовать в стандартных САПР АСХП в виде электрических схем. Информационные макро модели проще в построении (например, модель в виде нейронной сети, связывающая входные и выходные характеристики), но есть проблемы включения их в модели АСХП.

Для физических макро моделей существует множество плохо формализованных подходов, но есть общие принципы. Так, для простейшего описания макро моделей логических элементов [2] отображают только функционально-логическое назначение моделируемого

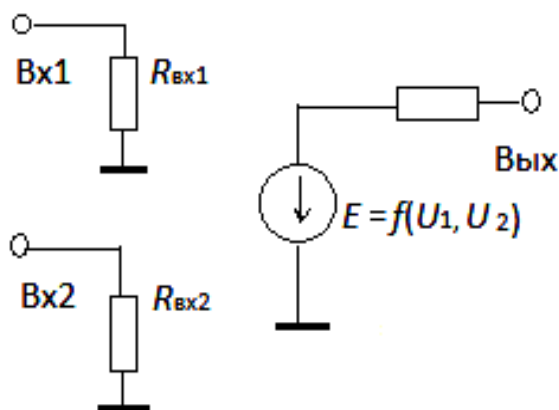


Рис. 6.1. Макро модель элемента И-НЕ

узла без учета его схемотехнических и технологических особенностей. Например, макро модель логического элемента И-НЕ (рис. 6.1) построена на основе идеального входного ($R_{вх} \rightarrow \infty$) и выходного ($R_{вых} \rightarrow 0$) сопротивлений и зависимого источника E , реализующего логическую функцию элемента.

Более сложные макро модели логических элементов можно строить с помощью более формализованного подхода, когда выделяют входные и выходные блоки, которые реализуют входные (блок 1) и выходные (блок 3) характеристики (рис. 6.2), и блок, реализующий основную передаточную функцию (блок 2).

Более сложные макро модели логических элементов можно строить с помощью более формализованного подхода, когда выделяют входные и выходные блоки, которые реализуют входные (блок 1) и выходные (блок 3) характеристики (рис. 6.2), и блок, реализующий основную передаточную функцию (блок 2).

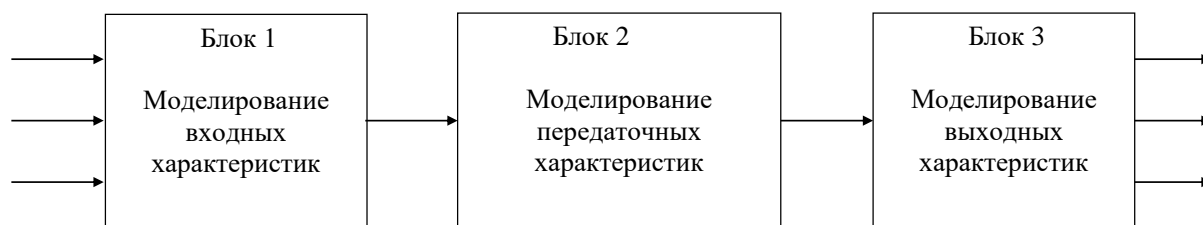


Рис. 6.2. Структурная схема макро моделей логических элементов

На рис. 6.3 представлена электрическая схема логического элемента И-НЕ, на рис. 6.4 – макромодель для данной схемы, полученной в виде трех блоков (см. рис. 6.2).

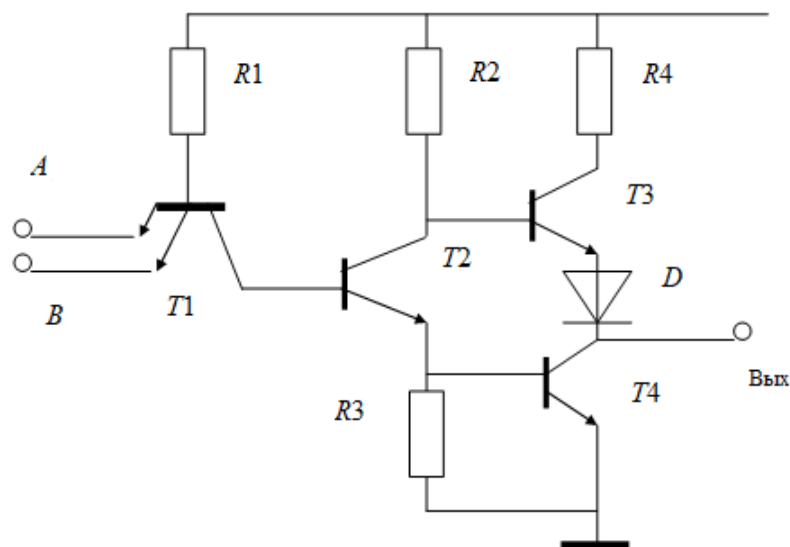


Рис. 6.3. Электрическая схема элемента И-НЕ

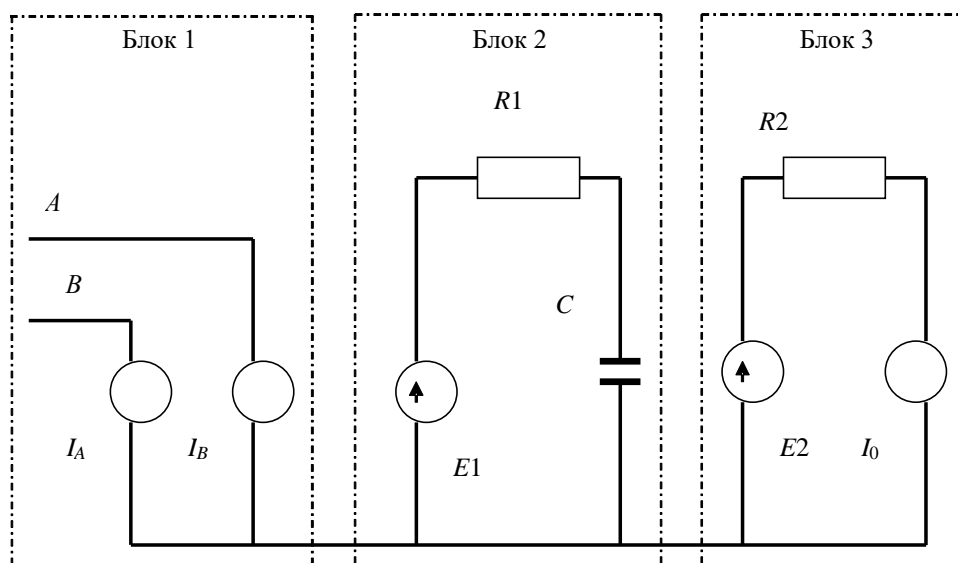


Рис. 6.4. Макромодель для схемы рис. 6.3

Рассмотрим подробнее макромодель, изображенную на рис. 6.4, где I_A , I_B , $E1$, $E2$ – управляемые источники тока и напряжения, I_0 – постоянный источник.

В блоке 1 управляемые источники $I_A(U_A)$ и $I_B(U_B)$ моделируют входные характеристики транзистора и могут быть реализованы в виде диодов со специальной вольт-амперной характеристикой или в табличной форме для функции $i = f(x)$.

В блоке 2 $E1 = f(U_A, U_B)$ – источник, управляемый напряжением, формируется на основе таблицы истинности для выполнения функции преобразования, инерционная цепь $R1C1$ задает задержку.

В блоке 3 $E2$ задает напряжение выхода, $R2$ – выходное сопротивление, I_0 – учитывает изменения выходной цепи.

Обычно блоки 1 и 3 выполняют сопряжение внешних и внутренних режимов, а также реализуют инерционные свойства входных и выходных цепей. Блок 2 выполняет основные функции преобразования сигналов и вносит суммарную инерционность модели.

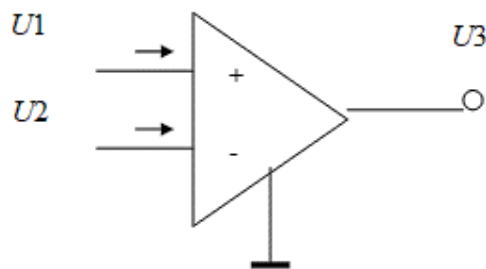


Рис. 6.5. Схема операционного усилителя

Один из самых распространенных аналоговых элементов – операционный усилитель (ОУ), который обладает очень высокими коэффициентом усиления и входным сопротивлением, а также низким значением выходного сопротивления (рис. 6.5).

Простейшая макромодель ОУ, учитывающая коэффициент усиления, входное $R_{вх}$ и выходное $R_{вых}$ сопротивления, представлена на рис. 6.6.

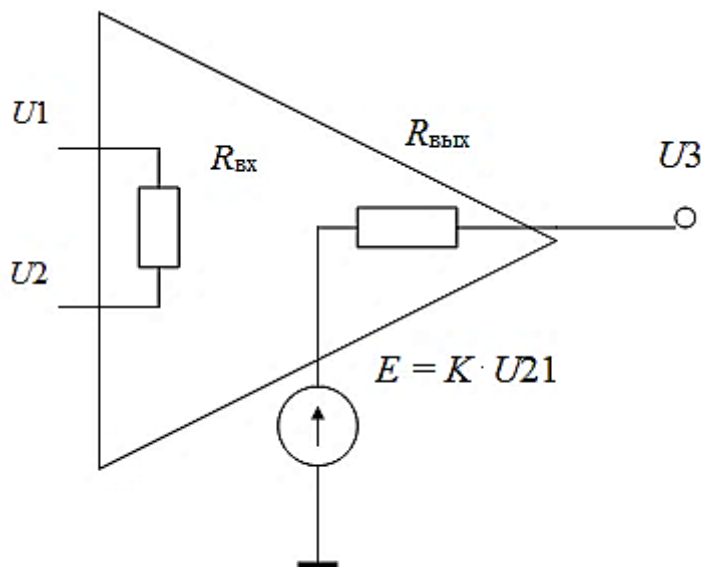


Рис. 6.6. Простейшая макромодель ОУ

Более сложная макромодель ОУ, которая дополнительно отображает характеристики быстродействия и т. п., приведена на рис. 6.7.

Здесь напряжения $E2$ и $E3$ являются функциями напряжений емкостных ветвей $C1$ и $C2$, напряжение $E4$ принимается равным напряжению на емкости $C3$, а напряжение $E5$ зависит от напряжений на емкостных ветвях $C4$ и $C5$. Емкости $C1$ и $C2$ – емкости входов, ветви $C3$ и $C4$ введены для более точного отображения переходных и частотных характеристик усилителя, емкость $C5$ – выходная емкость. Дрейф нуля отражается в макромодели с помощью источников напряжения $E1$ и тока $I1$ и $I2$. Ветвь $R1$ имитирует входное дифференциальное сопротивление для паразитных сигналов, а ветви $R2$ и $R3$ – для синфазных сигналов, сопротивление $R6$ есть активная составляющая выходного сопротивления. Данная макромодель отражает амплитудную, амплитудно-частотную, фазо-частотную характеристики, выходное и входное сопротивления, входное напряжение и входной ток, обуславливающие дрейф нуля.

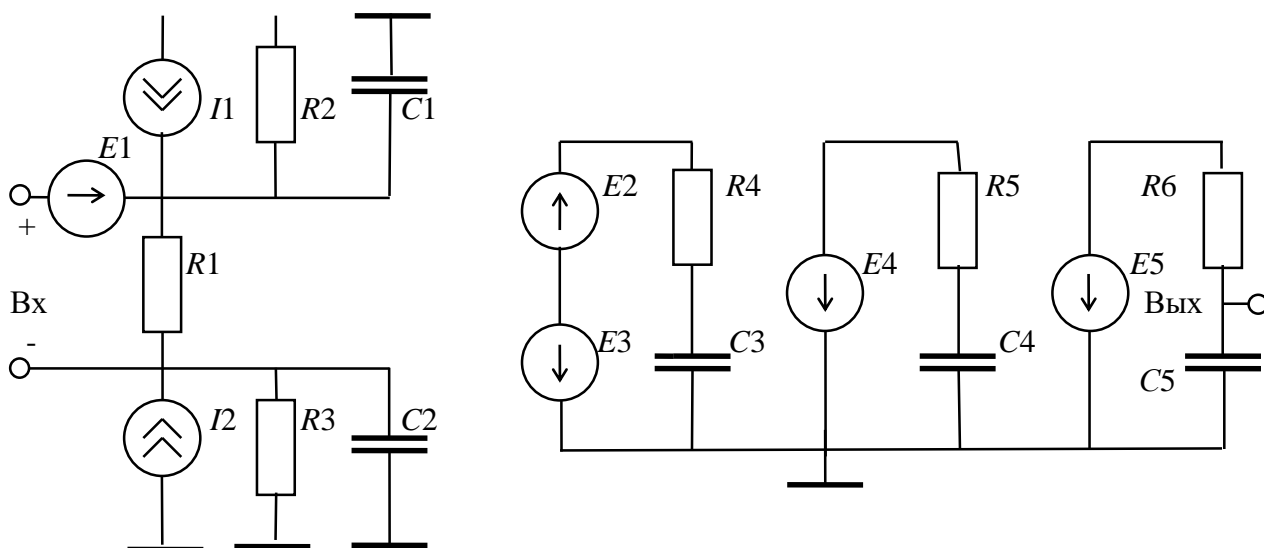


Рис. 6.7. Более сложная макромодель ОУ

Методика получения макромоделей (обобщенная). В общем случае процедура получения макромоделей совпадает с методикой построения математических моделей любых схмотехнических элементов и включает в себя следующие операции:

1. Отбор свойств объекта, которые подлежат отражению в модели (анализ возможных применений модели).
2. Сбор исходной информации о выбранных свойствах объекта. Источниками сведений могут быть опыт и знания инженера, научно-техническая справочная литература, описания прототипов, близкие по

своим свойствам имеющимся моделям, результаты экспериментальных измерений параметров и т. п.

3. Синтез структуры макромоделли (общий вид математических соотношений без конкретных числовых значений параметров, эквивалентные схемы, графы и т. п.). Это наиболее ответственная операция.

4. Расчет числовых значений параметров макромоделли с помощью минимизации погрешности между поведением модели и результатами эксперимента (либо физического, либо численного) с использованием более точных моделей, если таковые имеются.

5. Оценка точности и адекватности (области адекватности) макромоделли. Для оценки точности используются экспериментальные значения, не применявшиеся при четвертой операции.

Кроме того, применяют различные подходы на основе упрощения (сокращение несущественных переменных, объединение нескольких элементов в один и т. п.), а также методы функционального подбора.

6.2. Сложности многоуровневого моделирования

На современном этапе развития ЭВА разработчики сталкиваются с большими техническими сложностями. При проектировании сложных систем всегда имеется опасность «утонуть» в деталях, необходимо ограничивать количество информации, анализируемой и обрабатываемой в конкретный момент, стараясь не попадать в ситуацию, когда «за деревьями не видно леса» [54].

Это особенно характерно при разработке современной цифровой аппаратуры и СБИС. Сейчас специалисту, которому необходимо оценивать характеристики сложной СБИС или системы, построенной на базе таких СБИС, очень важно правильно выбрать уровень представления этой системы. В противном случае разобраться в работе системы будет очень трудно. Решение этой проблемы – работа на соответствующем уровне абстракции (детализации, представления), который дает всю информацию, необходимую в данный момент времени, и избавляет от необязательных деталей.

Уровни описания. Типичные уровни абстракции для цифровых систем приведены на рис. 6.8 [54]. Расширение усеченной пирамиды книзу отображает увеличение степени детализации при обработке системы.

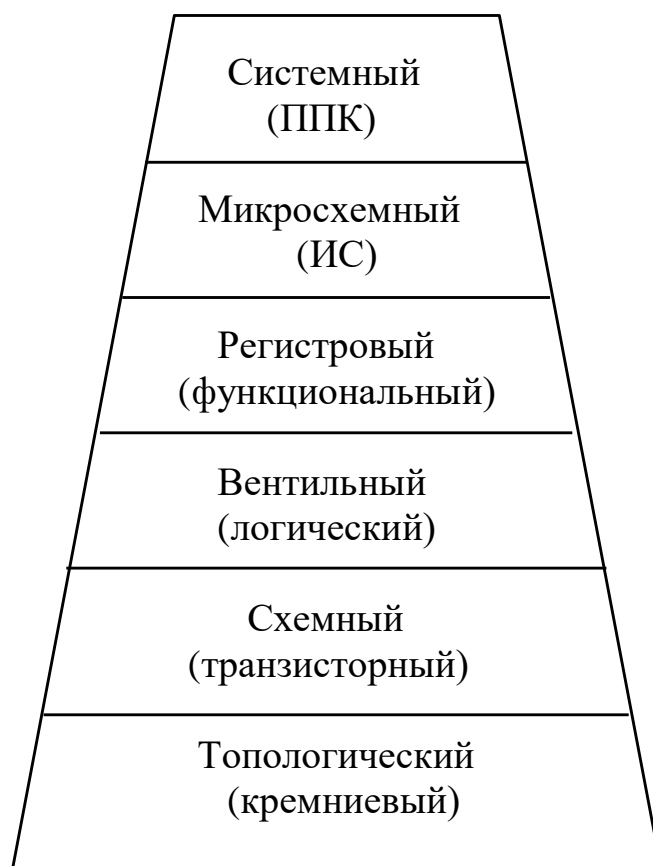


Рис. 6.8. Уровни описания цифровой аппаратуры

При любом иерархическом представлении, для любого технического объекта описание может относиться к одной из двух областей:

1) *структурная область* – область, где предусматривается описание компонента как совокупности взаимосвязанных компонентов более низкого (примитивного) уровня (отражаются структурные свойства: состав и связи);

2) *поведенческая область* – область, где предусматривается описание компонента по зависимостям вход/выход при помощи некоторой процедуры (отражаются функциональные свойства, алгоритм функционирования).

На рис. 6.9 показаны примеры представления фрагментов схем для уровней (по рис. 6.8).

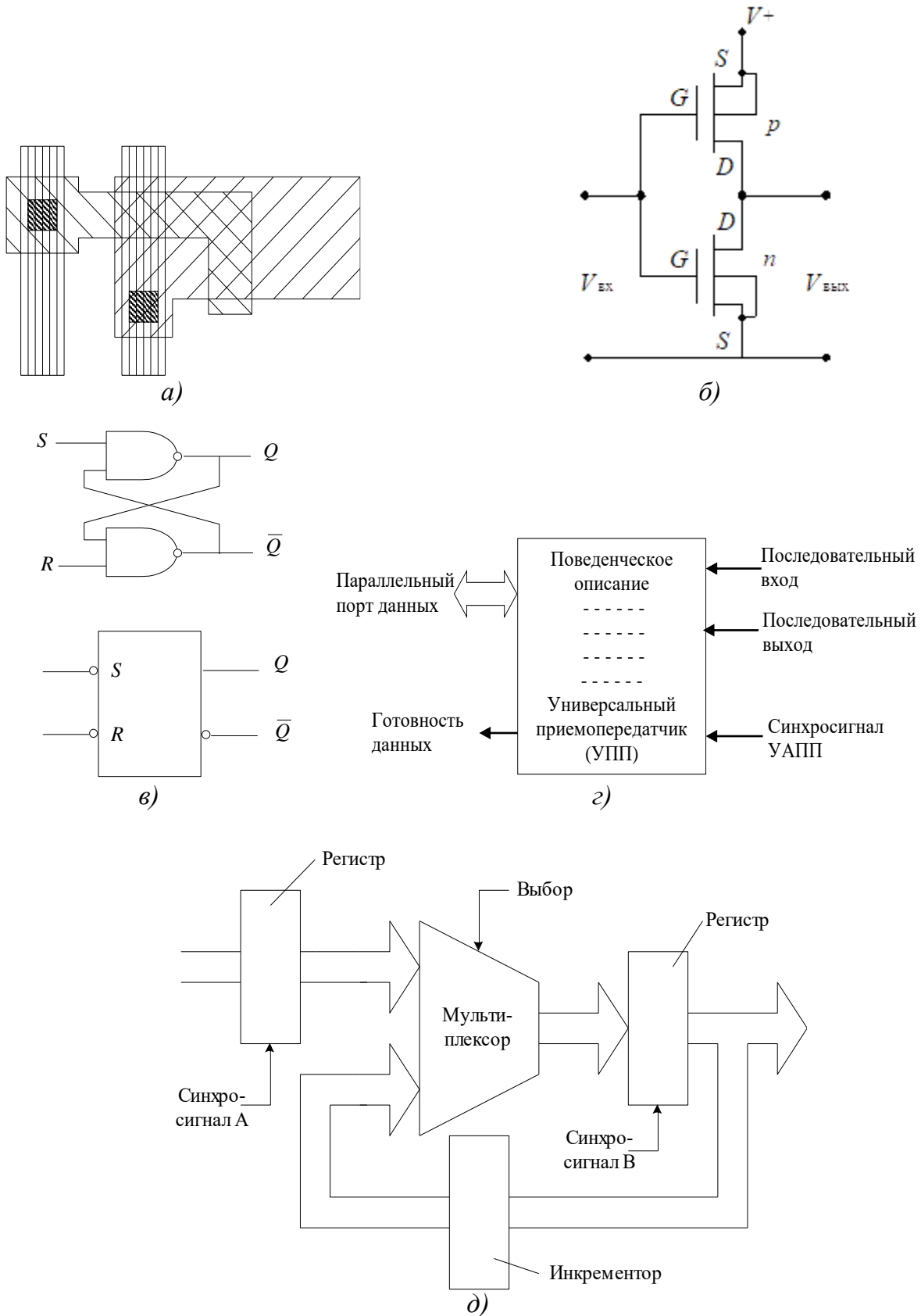
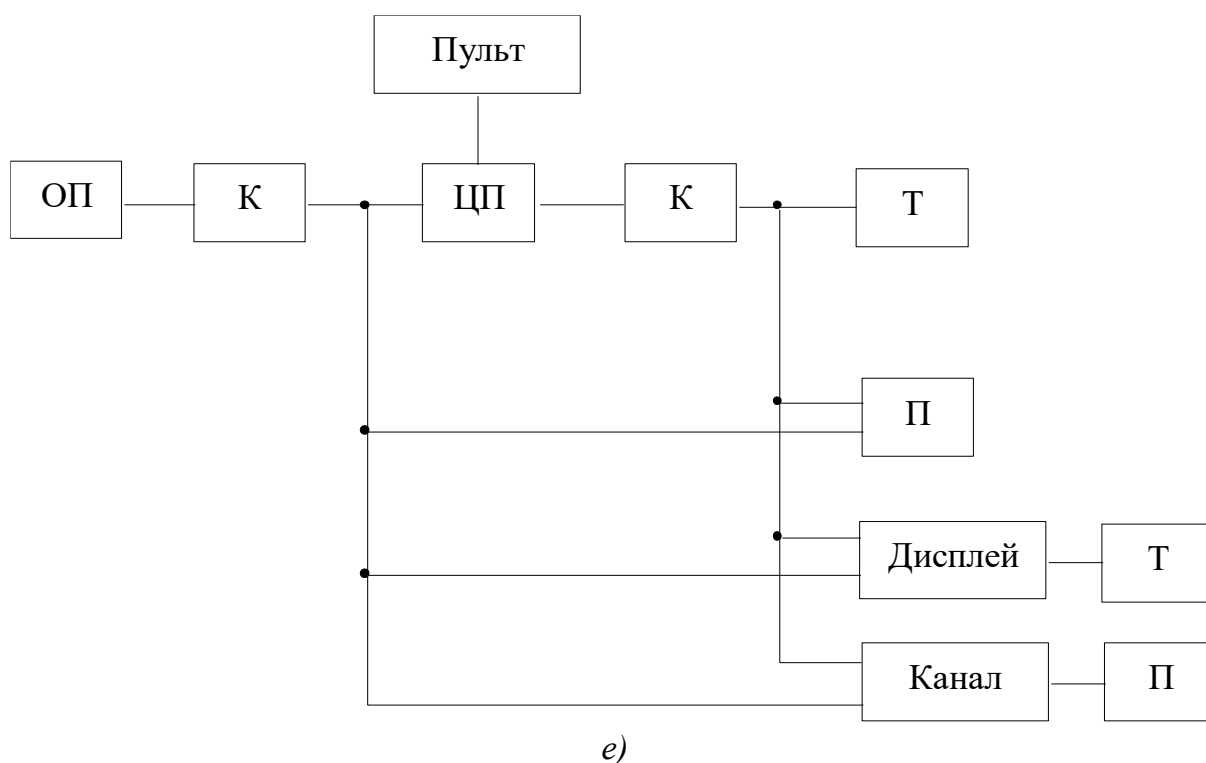


Рис. 6.9. Примеры уровней представления фрагментов схем:
 а – топологический; б – схемный; в – логический; г – регистровый; д – системный (начало)



e)

Рис. 6.9. Окончание

В табл. 9 приведены характеристики примитивов отдельных уровней для структурного и поведенческого представления [54]. На самом нижнем, топологическом, уровне в качестве базовых примитивов (из которых строится объект) используются геометрические объекты, которые представляют области диффузии поликремния и металлизации на поверхности кремниевого кристалла. Соединение этих компонентов как бы имитирует процесс изготовления кристалла. Отметим, что для такого представления поведенческой формы не существует, хотя часто в качестве математической модели поведения на данном уровне выступают дифференциальные уравнения в частных производных (ДУЧП).

На следующем, более высоком, уровне – схемном – представление проекта формируется на основе соединения примитивов более низкого уровня, традиционных пассивных и активных элементов электрической схемы: резисторов, конденсаторов и транзисторов. Эти структурные описания для отражения поведения электрической схемы используют вольт-амперные характеристики компонентов или в более общем случае дифференциальные уравнения.

Иерархическое представление моделей

Уровень представления	Структурные примитивы	Поведенческое представление
1. Системный	Процессоры, памяти, каналы, шины	Алгоритм функционирования, вероятностное описание, технические данные и характеристики
2. Микросхемный	Микропроцессоры, ПЗУ, УПП, порты	Входные/выходные зависимости, алгоритмы функционирования, микрооперации
3. Регистровый	Регистры, АЛУ, счетчики, дешифраторы, мультиплексоры	Таблицы истинности, таблицы состояний, микрооперации
4. Вентильный	Логические вентили, триггеры	Булевы (логические) формулы
5. Схемный	Транзисторы, резисторы, емкости	Дифференциальные уравнения
6. Топологический	Геометрические объекты	Нет (или ДУЧП)

На уровне логических вентилях в качестве базовых примитивов используются логические элементы И, ИЛИ, НЕ, различные типы триггеров. Соединение таких «кирпичиков» позволяет строить любые логические схемы. В качестве поведенческих описаний (моделей) на этом уровне используется алгебра логики или булевы формулы.

Выше вентильного уровня в иерархии находится регистровый. В качестве базовых примитивов используются регистры, счетчики, дешифраторы, арифметико-логические устройства и т. п. Иногда их называют функциональными блоками, а сам уровень – функциональным. Здесь также можно использовать примитивы в виде логических вентилях, но чаще применяют более крупные блоки, описываемые уже таблицами истинности и состояний. Поведенческое представление на регистровом уровне также возможно с помощью языков регистровых передач.

Над регистровым уровнем находится уровень микросхем. Здесь в качестве структурных примитивов выступают такие компоненты, как микропроцессоры, устройства памяти, последовательные и параллельные порты, контроллеры. Хотя микросхемы являются и моделями самих примитивов, возможна ситуация, когда несколько микросхем образуют одно функциональное устройство и составляют один примитив. Возможен и альтернативный вариант, когда примитивы представляют собой отдельные фрагменты одной микросхемы. Главная особенность состоит в том, что примитивом считается большой блок логики, описываемый зависимостями вход/выход, и представляющий собой единый объект-модель. Такие модели очень важны для системотехников, которые приобретают микросхемы у других фирм-изготовителей и не знают их внутренней структуры на уровне логических вентилях или регистровых передач. Поэтому поведенческое описание модели микросхемного уровня строится на основе входных/выходных зависимостей каждой конкретной ИМС или алгоритма, реализуемого данной ИМС.

Верхний уровень иерархии структур – системный, или уровень «процессоров памяти коммутаторов» (ППК). В качестве примитивов используются процессоры, памяти и коммутаторы (шины). Поведенческое описание на этом уровне основывается на алгоритмах функционирования и таких технических характеристиках, как быстродействие или пропускная способность.

Из приведенных рассуждений и табл. 9 видно, что структурные или поведенческие характеристики соседних уровней в определенной степени перекрываются. Например, и на системном, и на микросхемном уровнях может использоваться представление с помощью алгоритма функционирования, а на микросхемном и регистровом уровнях – микрооперации. Однако структурные представления для обоих случаев различаются, поэтому они и разделяются на разные уровни. Уровень регистров и вентилях имеет по сути одни и те же структурные примитивы, однако различаются по своим поведенческим характеристикам.

Многоуровневое моделирование. На основе структурных свойств легко осуществляется декомпозиция (разбиение) проекта, так как модель системы строится из совокупности взаимосвязанных примитивов более низкого уровня. Таким образом, любой проект может быть представлен в виде дерева, причем разным уровням иерархии детализации соответствуют свои уровни дерева (рис. 6.10). Но на основе структурных свойств не может быть получена полная модель системы,

поэтому листьями дерева всегда будут поведенческие описания для компонентов самого низкого уровня.

Таким образом, поведенческая модель – это базовая, примитивная модель, функционирование которой определяется ее собственной процедурой, не использующей другие примитивы. Модель системы в целом будет строиться из набора поведенческих моделей для каждого блока. Так как поведенческая модель может существовать на схемном, вентиляльном, регистровом или микросхемном уровнях, то и различные части проекта могут иметь поведенческие описания на разных уровнях.

На рис. 6.10 показано полное дерево проекта, где все поведенческие описания формируются на одном и том же уровне. На рис. 6.11 показан проект, представленный в виде неполного дерева, где поведенческие описания относятся к различным уровням. Представление проекта в виде неполного дерева необходимо при отладке фрагментов более низких уровней при учете их взаимосвязи с фрагментами более высоких уровней, не ожидая завершения проектирования остальных частей. Такой анализ осуществляется при помощи многоуровневого моделирования, при котором поведенческие описания моделей компонентов относятся к различным уровням иерархии.

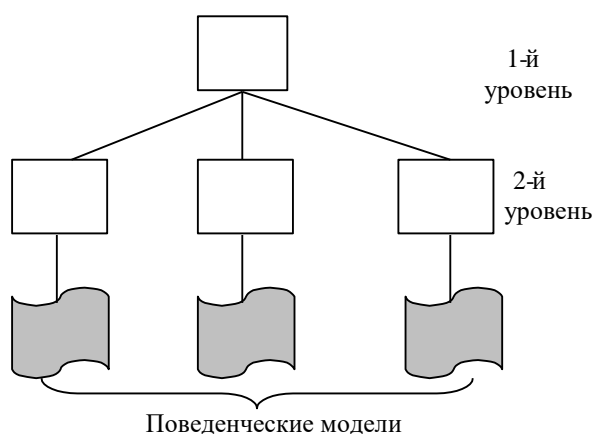


Рис. 6.10. Проект в виде полного дерева

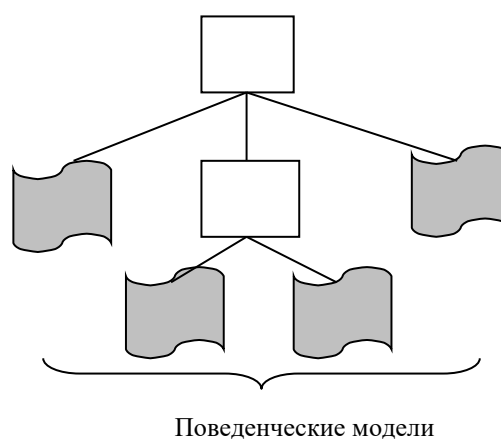


Рис. 6.11. Проект в виде неполного дерева

6.3. Организация компиляторов для многоуровневого моделирования

В основе большинства современных программ моделирования лежит принцип событийного моделирования. Условная работа системы моделирования приведена на рис. 6.12.

Модель каждого устройства представляется в виде модуля или некоторой процедуры, взаимодействие между модулями – как вызов процедур. События, которые происходят в рамках моделируемой структуры, отображаются во временной очереди (часто называется списком будущих событий). Каждый элемент в списке будущих событий указывает номер события (имя процедуры-подпрограммы), которое должно произойти в данный или будущий момент времени. Каждый очередной шаг моделирования начинается с выбора из списка событий ближайшего по времени, по ссылкам, которые имеются в списке, определяется номер модуля (процедуры), куда передается управление. Процедура моделирования данного блока включает все действия с сигналами или другими переменными, становится известно будущее событие, которое произойдет в новый (но уже известный) момент времени. Если в данный момент модельного времени событий в списке нет, то выбирается следующий момент времени, соответствующий ближайшему по времени событию. Рис. 6.12, б иллюстрирует работу такой системы.

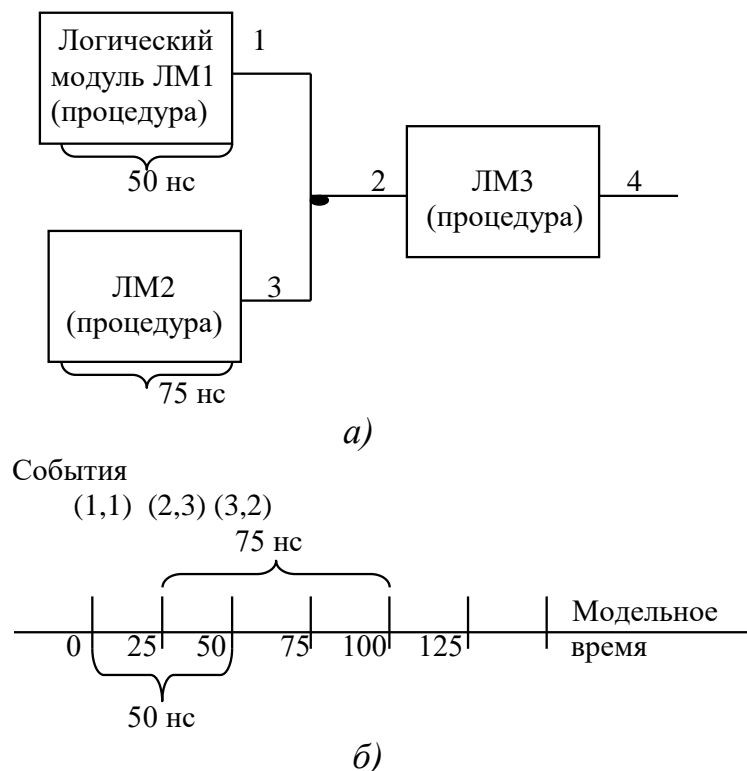


Рис. 6.12. Пример моделирования: а – фрагмент системы; б – модельное время и список событий

Пусть в момент модельного времени 0 произошло изменение на входе логического модуля ЛМ1. Вызывается процедура моделирования для ЛМ1, в результате чего становится известно, что через 25 нс задержки в ЛМ1 произойдет событие – изменение сигнала на выходе 1 ЛМ1. Это будущее (предвиденное) событие заносится в список событий с указанием номера входа ЛМ3, где задержалось прохождение сигнала. Далее проверяется список на наличие событий, относящихся к моменту модельного времени 0. Так как событий для момента времени 0 (в данном примере) нет, то модельное время изменяется на величину, равную времени до ближайшего на оси времени события. Затем инициализируются выполнения модулей, связанных с текущим (новым) моментом модельного времени, что соответствует вызову процедуры моделирования для ЛМ2. После моделирования становится известно возможное появление отклика на выходе ЛМ2 (будущее событие), которое заносится в список будущих событий с учетом задержки в ЛМ2, равной 75 нс. Обработка событий продолжается до тех пор, пока либо не исчерпается очередь событий, либо не истечет какое-то предельное время, задаваемое пользователем или иным способом.

Приведенный пример иллюстрирует, во-первых, что современные программы на основе событийного алгоритма могут использовать модели, описываемые любым способом и на любом уровне иерархии, так как под подпрограммой можно понимать любой алгоритм преобразования исходных данных на ее входе в какие-либо данные (реакцию) на выходе. Любые модели обрабатываются по принципу «событие-за-событием». Во-вторых, в моделях, а следовательно, и в языках описания аппаратуры должен иметься механизм планирования событий и времени.

Эффективность и удобство моделирования во многом зависят и от применяемого способа организации. Под способом организации программного обеспечения понимается вид системы моделирования – компилятор или интерпретатор [3]. В случае *интерпретатора* (табличное управление) исходные данные о системе после ввода в ЭВМ хранятся в виде таблиц и списков. Модели отдельных компонентов вызываются во время обработки в порядке, обусловленном прохождением сигналов между компонентами, т. е. модель «интерпретируется» системой моделирования во время прогона. В данном случае еще до

ввода исходных данных о проекте имеется полностью скомпонованный рабочий модуль программы, универсальный по отношению к моделям.

В компиляторах вся модель компилируется в код ЭВМ до начала моделирования и никакой интерпретации во время прогона не требуется. Системы моделирования с компиляцией работают значительно быстрее, чем с интерпретацией, так как в них нет лишних проверок (условных операторов и разветвлений) и вызова ненужных процедур, требуемый объем памяти также меньше, потому что подключаются модели тех компонентов, которые присутствуют именно в этих исходных данных. Проблемой компиляторов всегда была плохая переносимость с одного типа ЭВМ на другой или другую операционную систему. Но с появлением стандартных языков типа VHDL эта проблема исчезла.

6.4. Математическая модель схемы в расширенном координатном базисе и связь с механическими системами

Расширенный координатный базис. Ранее было отмечено, что в методе узловых потенциалов очень неудобно моделировать такие элементы, как индуктивности и источники напряжений. Для преодоления этого недостатка, а также моделирования элементов систем управления были исследованы несколько приемов. Для включения этих неудобных элементов без дополнительных преобразований был предложен модифицированный узловый анализ (МУА, или анализ в расширенном координатном базисе) [4].

Рассмотрим пример формирования уравнений МУА на основе линейной системы обыкновенных дифференциальных уравнений в наиболее полной форме [55]:

$$E \frac{dx}{dt} Ax(t) + Bu(t) = 0, \quad y(t) = Cx(t) + Du(t), \quad (6.1)$$

где E , A , B , C и D – матрицы; $x(t)$ – вектор переменных состояний размерностью N (размерность модели); $y(t)$ – вектор выходных сигналов схемы размерностью N_0 ; $u(t)$ – вектор входных воздействий размерностью N_i ; B – матрица размерностью $(N \times N_i)$; N_i – число входов схемы; t – время.

Возьмем простейшую схему, содержащую три узла, где присутствует индуктивность (рис. 6.13).

Запишем для этого примера уравнения для трех узлов:

$$- \text{ для узла } 1 \quad C \frac{dv_1}{dt} + \frac{v_1 - v_2}{R} - I_1^{\text{BX}} = 0;$$

$$- \text{ для узла } 2 \quad \frac{v_2 - v_1}{R} = I_L = 0;$$

$$- \text{ для узла } 3 \quad C \frac{dv_3}{dt} - I_L - I_2^{\text{BX}} = 0,$$

а также уравнение для индуктивности в виде $L \frac{dI_L}{dt} = v_2 - v_3$.

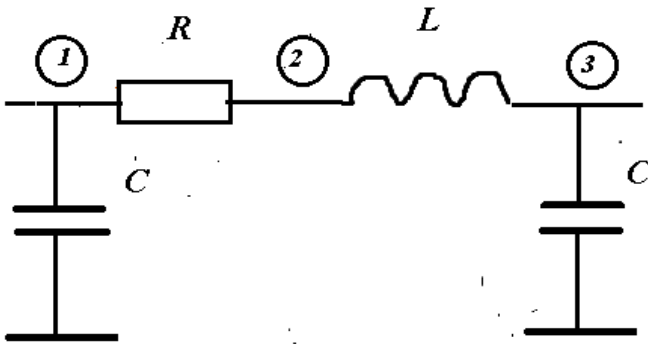


Рис. 6.13. Пример схемы для формирования уравнений МУА

Учтем, что в этих уравнениях введены входные источники тока:

$$- \text{ для узла } 1 \quad I_1^{\text{BX}};$$

$$- \text{ для узла } 3 \quad I_2^{\text{BX}}.$$

Под выходными напряжениями будем понимать $v_1^{\text{ВЫХ}} = v_1$ и $v_2^{\text{ВЫХ}} = v_3$. В этом случае векторы и матрицы системы уравнений (6.1) будут иметь вид

$$x = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ I_L \end{bmatrix}, \quad E = \begin{bmatrix} C & & & \\ & 0 & & \\ & & C & \\ & & & L \end{bmatrix}, \quad A = - \begin{bmatrix} \frac{1}{R} & -\frac{1}{R} & & \\ -\frac{1}{R} & \frac{1}{R} & & \\ & & 1 & \\ & & & -1 \\ & & -1 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad u = \begin{bmatrix} I_1^{\text{BX}} \\ I_2^{\text{BX}} \end{bmatrix}, \quad y = \begin{bmatrix} v_1^{\text{ВЫХ}} \\ v_2^{\text{ВЫХ}} \end{bmatrix}.$$

Модели механических систем. Существуют аналогии для систем схемотехнического моделирования электрических схем и систем другой физической природы. Это впервые отмечено в работе [2], реализовано в программном обеспечении [3], а затем применено при разработке программы моделирования гидравлических систем в ВЛГУ.

Компонентные и топологические уравнения в системах различной физической природы отражают разные физические свойства, но

могут иметь одинаковый формальный вид [4]. Такие аналогии существуют для механических поступательных, механических вращательных, электрических, гидравлических (пневматических), тепловых объектов. Таким образом, значительная часть алгоритмов формирования моделей для АСхП может быть применена к анализу проектируемых объектов в разных предметных областях.

Переменными в механических поступательных системах являются силы и скорости. Обычно скорость относят к переменным типа потенциала, а силу считают переменной типа потока. Компонентное уравнение, характеризующее инерционные свойства тел, в силу второго закона Ньютона имеет вид

$$F = Mdu/dt, \quad (6.2)$$

где F – сила; M – масса; u – поступательная скорость.

Компонентное уравнение для упругих свойств (из уравнения закона Гука) в одномерном случае

$$G = E\varepsilon, \quad (6.3)$$

где G – механическое напряжение; E – модуль упругости; ε – относительная деформация, $\varepsilon = \Delta l/l$; Δl – изменение длины l упругого тела под воздействием G . Учитывая, что $G = F/S$, где F – сила, S – площадь сечения, и, дифференцируя (6.3), имеем

$$dF/dt = (SE/l) d(\Delta l)/dt$$

или

$$dF/dt = gu, \quad (6.4)$$

где g – жесткость (гибкость), $g = SE/l$; u – скорость, $u = d(\Delta l)/dt$.

Таким образом, диссипативные свойства в механических системах соответствуют закону Ома в электрических системах, токам и напряжениям – силы (или моменты) и скорости механических систем.

В механических вращательных системах справедливы компонентные и топологические уравнения поступательных систем с заменой поступательных скоростей на угловые, сил – на вращательные моменты, масс – на моменты инерции, жесткостей – на вращательные жесткости [4].

Условные обозначения простых элементов механической системы показаны на рис. 6.14, б. Нетрудно заметить наличие аналогий между электрической и механической системами. Так, токам и напряжениям соответствуют силы (либо моменты) и скорости механической системы, а фигурирующие в уравнениях (6.2) и (6.4) параметры аналогичны топологическим уравнениям, содержащим емкости C и индуктивности L .

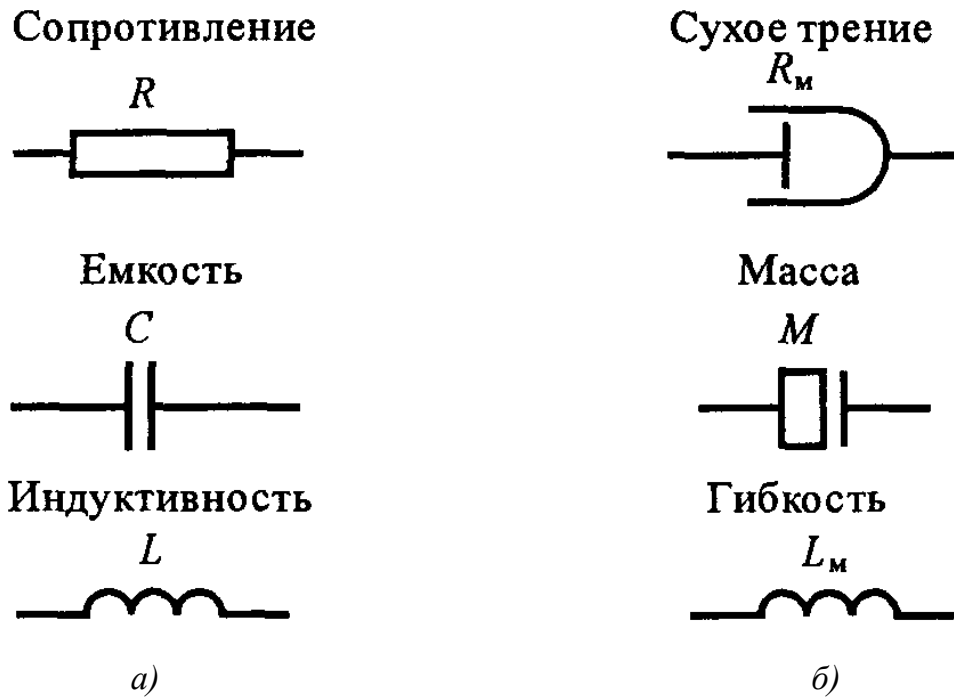


Рис. 6.14. Условные обозначения простых элементов в эквивалентных схемах: *a* – электрических, гидравлических, тепловых; *б* – механических

Гидравлические системы. Здесь переменными являются расход и давление. Компонентные уравнения описывают свойства жидкости рассеивать или накапливать энергию [4].

Рассмотрим компонентное уравнение для жидкости на линейном участке трубопровода длиной Δl и используем уравнение Навье-Стокса в форме

$$\rho dU/dt = - dp/dx - 2aU,$$

где ρ – плотность жидкости; U – скорость; p – давление; a – коэффициент трения. Так как $U = Q/S$, где Q – объемный расход, S – площадь сечения, заменяя производную отношением конечных разностей, получим

$$dQ/dt = S/(\Delta l \rho) \Delta p - 2a/\rho Q$$

или

$$\Delta p = L dQ/dt + RQ, \quad (6.5)$$

где Δp – падение давления; L – гидравлическая индуктивность (упругие свойства жидкости), $L = (\Delta l \rho)/S$; R – гидравлическое сопротивление (вязкое трение), $R = 2a/\rho$.

Интерпретация уравнения (6.5) приводит к эквивалентной схеме последовательного соединения R и L [4].

6.5. Модели и методы решения дифференциальных уравнений в частных производных

Математические модели на микроуровне. Математическими моделями на микроуровне являются дифференциальные уравнения в частных производных (ДУЧП, Partial Differential Equations, PDE), описывающие поля физических величин [4]. Другими словами, на микроуровне используются модели с распределенными параметрами. В качестве независимых переменных в моделях могут фигурировать пространственные переменные (координаты) x_1, x_2, x_3 и время t .

Характерными примерами моделей могут служить уравнения математической физики вместе с заданными краевыми условиями, например [4]:

1) уравнение теплопроводности

$$C\rho \frac{\partial T}{\partial t} = \operatorname{div}(\lambda \operatorname{grad} T) + g,$$

где C – удельная теплоемкость; ρ – плотность; T – температура; t – время; λ – коэффициент теплопроводности; g – количество теплоты, выделяемой в единицу времени в единице объема;

2) уравнение диффузии

$$\frac{\partial N}{\partial t} = \operatorname{div}(D \operatorname{grad} N),$$

где N – концентрация частиц; D – коэффициент диффузии;

3) уравнения непрерывности, используемые в физике полупроводниковых приборов:

– для дырок

$$\frac{\partial p}{\partial t} = -(1/q) \operatorname{div} J_p + g_p,$$

– для электронов

$$\frac{\partial n}{\partial t} = -(1/q) \operatorname{div} J_n + g_n;$$

4) уравнение Пуассона

$$\operatorname{div} E = \rho / (\epsilon \epsilon_0).$$

Здесь p и n – концентрации дырок и электронов; q – заряд электрона; J_p и J_n – плотности дырочного и электронного токов; g_p и g_n – скорости процессов генерации-рекомбинации дырок и электронов; E – напряженность электрического поля; ρ – плотность электрического заряда; ϵ и ϵ_0 – диэлектрическая проницаемость и диэлектрическая постоянная.

Краевые условия включают в себя начальные условия, характеризующие пространственное распределение зависимых переменных в начальный момент времени, и граничные, задающие значения этих переменных на границах рассматриваемой области в функции времени.

Методы решения ДУЧП. В САПР решение дифференциальных уравнений с частными производными выполняется численными методами. Эти методы основаны на дискретизации независимых переменных – их представлении конечным множеством значений в выбранных узловых точках исследуемого пространства. Эти точки рассматриваются как узлы некоторой сетки, поэтому используемые в САПР методы – это *сеточные методы*. Сетка, в свою очередь, представляет собой дискретную модель области определения искомой функции. С помощью дискретизации непрерывная математическая модель преобразуется в дискретную модель. В этом случае результатом дискретизации выступает система СНДУ для нестационарной задачи или система алгебраических уравнений для стационарной.

Среди сеточных методов наибольшее распространение получили метод конечных разностей (МКР), метод конечных элементов (МКЭ) и метод граничных элементов (МГЭ).

В настоящее время наиболее распространен метод конечных элементов, в то время как метод граничных элементов для решения задач применяется редко. Классический метод конечных разностей используется при решении некоторых частных задач. Если рассматривать нелинейные задачи, то здесь метод конечных элементов наиболее эффективный и доминирующий.

Методы конечных элементов и конечных разностей имеют ряд существенных отличий, основное из которых состоит в том, что в МКР аппроксимируются производные искомых функций, а в МКЭ – само решение, т. е. зависимость искомых функций от пространственных координат и времени. Методы сильно отличаются и в способе построения сеток.

В МКР строятся, как правило, регулярные сетки, особенности геометрии области учитываются только в околограничных узлах. В связи с этим метод чаще применяется для анализа задач с прямолинейными границами областей определения функций.

В МКЭ разбиение на элементы производится с учетом геометрических особенностей области, процесс разбиения начинается от границы с целью наилучшей аппроксимации её геометрии. Затем на элементы разбивают внутренние области, причем алгоритм разбиения

строится так, чтобы элементы удовлетворяли некоторым ограничениям, например стороны треугольников не слишком отличались по длине и т. д. Поэтому МКЭ наиболее часто используется для решения задач с произвольной областью определения функций.

Общая проблема всех методов – высокая размерность результирующей системы алгебраических уравнений (несколько десятков тысяч в реальных задачах). Поэтому реализация МКР и МКЭ в составе САПР требует разработки специальных способов хранения матрицы коэффициентов системы и ее методов решения.

В *методе конечных разностей* алгебраизация производных по пространственным координатам базируется на аппроксимации производных конечно-разностными выражениями. При использовании метода нужно выбрать шаги сетки по каждой координате и вид шаблона. Под шаблоном понимают множество узловых точек, значения переменных в которых используются для аппроксимации производной в одной конкретной точке.

Примеры шаблонов для одномерных и двумерных задач приведены на рис. 6.15, где кружком большого диаметра обозначены узлы, в которых аппроксимируется производная; черными точками – узлы, значения фазовой переменной в которых входят в аппроксимирующее выражение. Число, записанное около узла, равно коэффициенту, с которым значение фазовой переменной входит в аппроксимирующее выражение. Так, для одномерных шаблонов в верхней части рисунка показана аппроксимация производной dV/dx в точке k , и указанным шаблонам при их просмотре слева направо соответствуют аппроксимации:

1) правая схема $\Delta x(dV/dx) = V_{k+1} - V_k$ или левая схема $\Delta x(dV/dx) = V_k - V_{k-1}$;

2) центральная схема $2\Delta x(dV/dx) = V_{k+1} - V_{k-1}$;

3) аппроксимация более высокого порядка $h^2(d^2V/dx^2) = V_{k+1} - 2V_k + V_{k-1}$.

Здесь Δx – шаг дискретизации по оси x .

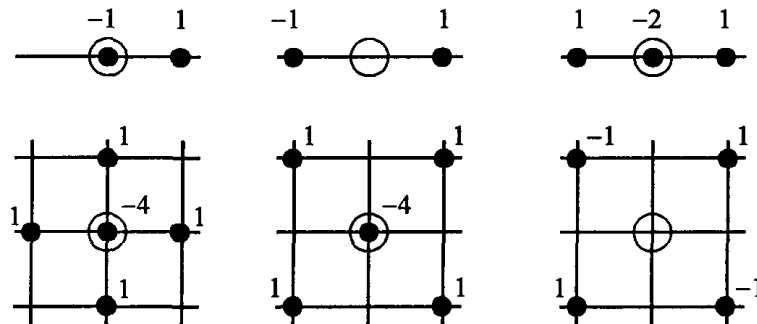


Рис. 6.15. Примеры шаблонов для МКР

На рис. 6.16 представлена графическая интерпретация некоторых конечно-разностных аппроксимаций для производных.

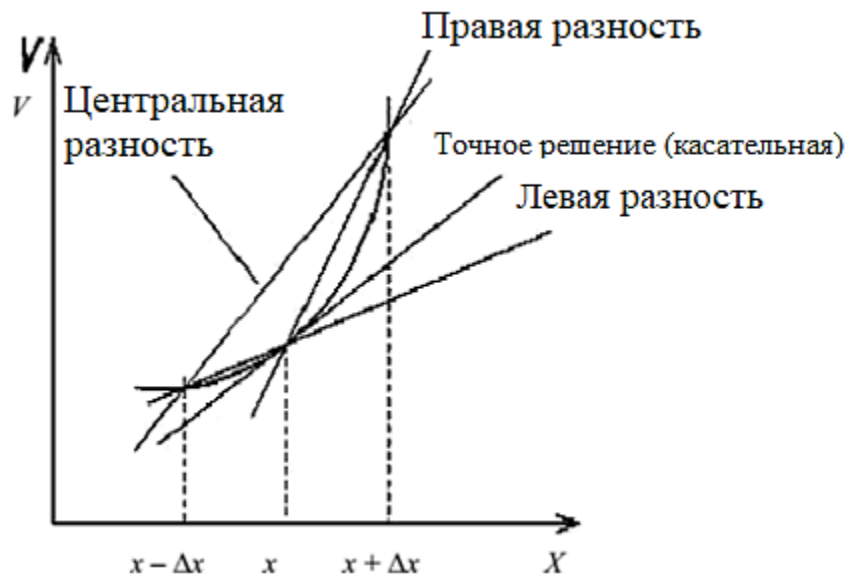


Рис. 6.16. Графическая интерпретация конечно-разностных аппроксимаций

Шаблоны для двумерных задач в нижней части рис. 6.15 соответствуют следующим конечно-разностным операторам [4]:

а) левый рисунок:

$$h^2 \nabla^2 V = h^2 (\partial^2 V / \partial x_1^2 + \partial^2 V / \partial x_2^2) = V_{k+1,j} + V_{k-1,j} + V_{k,j+1} + V_{k,j-1} - 4V_{k,j};$$

б) средний рисунок:

$$2h^2 \nabla^2 V = V_{k+1,j+1} + V_{k-1,j+1} + V_{k+1,j-1} + V_{k-1,j-1} - 4V_{k,j};$$

в) правый рисунок:

$$4h^2 \partial^2 V / (\partial x_1 \partial x_2) = V_{k+1,j+1} - V_{k-1,j+1} - V_{k+1,j-1} + V_{k-1,j-1}.$$

Здесь V_k — значение V в точке (x_{1k}, x_{2j}) ; приняты одинаковые значения шагов h по обеим координатам.

Алгоритм МКР состоит из этапов, традиционных для метода сеток.

1. Построение сетки в заданной области. В МКР используется сетка, задаваемая конечным множеством узлов. В узлах сетки определяются приближенные значения φ_h искомой функции φ . Совокупность узловых значений φ_h называют сеточной функцией.

2. Замена дифференциального оператора $L_h = \partial \varphi / \partial u$ в исходном дифференциальном уравнении разностным аналогом L_h , построенным

по одной из схем, рассмотренных ниже. При этом непрерывная функция φ аппроксимируется сеточной функцией φ_h . В результате должна получиться замкнутая система СНАУ.

3. Решение полученной нелинейной системы алгебраических уравнений.

Рассмотрим задачу в одномерной области. На рис. 6.17 представлена одномерная сетка, состоящая из четырех точек.



Рис. 6.17. Одномерная сетка с четырьмя узлами

Решим уравнение:

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) = 0,$$

где $k = k(T) = T^2$; T – функция температуры.

Применив центральную разностную схему, получим СНАУ (6.6), решив которую с учетом граничных условий значение температуры в узлах сетки будет иметь вид

$$\begin{cases} \frac{\left(\frac{T_2 + T_1}{2}\right)^2 \frac{T_2 - T_1}{\Delta x} - \left(\frac{T_1 - T_0}{2}\right)^2 \frac{T_1 - T_0}{\Delta x}}{\Delta x} = 0, \\ \frac{\left(\frac{T_3 + T_2}{2}\right)^2 \frac{T_3 - T_2}{\Delta x} - \left(\frac{T_2 - T_1}{2}\right)^2 \frac{T_2 - T_1}{\Delta x}}{\Delta x} = 0. \end{cases} \quad (6.6)$$

Метод конечных элементов основан на аппроксимации не производных, а самого решения $V(z)$. Пусть необходимо решить ДУЧП в общем виде

$$LV(z) = f(z)$$

с заданными краевыми условиями

$$MV(z) = \varphi(z),$$

где L и M – дифференциальные операторы; $V(z)$ – фазовая переменная; z – вектор независимых переменных, $z = (x_1, x_2, x_3, t)$; $f(z)$ и $\varphi(z)$ – заданные функции независимых переменных.

Поскольку решение неизвестно, то аппроксимация выполняется выражениями с неопределенными коэффициентами q_i

$$U(z) = Q^T \varphi(z),$$

где Q^T – вектор-строка неопределенных коэффициентов, $Q^T = (q_1, q_2, \dots, q_n)^T$; $\varphi(z)$ – вектор-столбец координатных (иначе опорных) функций, заданных так, что удовлетворяются граничные условия.

При этом речь идет об аппроксимациях решения в пределах конечных элементов, а с учетом их малых размеров можно говорить об использовании сравнительно простых аппроксимирующих выражений $U(z)$ (например, $\varphi(z)$ – полиномы низких степеней). В результате подстановки $U(z)$ в исходное дифференциальное уравнение и выполнения операций дифференцирования получаем систему невязок

$$\Delta(z, Q) = LU(z) - f(z) = L(Q^T \varphi(z)) - f(z), \quad (6.7)$$

из которой требуется найти вектор Q .

Эту задачу (определение Q) решают одним из следующих методов:

1) *метод коллокаций*, в котором, используя уравнение (6.7), формируют n уравнений с неизвестным вектором Q :

$$L(Q^T \varphi(z_i)) - f(z_i) = 0, \quad i = 1, 2, \dots, n,$$

где n – число неопределенных коэффициентов;

2) *метод наименьших квадратов*, основанный на минимизации квадратов невязок (6.7) в n точках или в среднем по рассматриваемой области;

3) *метод Галеркина*, с помощью которого минимизируются в среднем по области невязки со специально задаваемыми весовыми коэффициентами.

Основа физической концепции МКЭ – это разбиение математической модели конструкции на непересекающиеся компоненты (подобласти) простой геометрии, называемые конечными элементами или просто элементами (для краткости) [4]. Множество элементов, на которые разбита конструкция, называется конечно-элементной сеткой. Поведение математической модели, таким образом, аппроксимируется поведением дискретной модели, полученной путем сборки, или ансамблирования, всех элементов. Заметим также, что в отличие от метода конечных разностей конечные элементы не накладываются друг на друга в пространстве.

Конечные элементы могут описываться одной, двумя или тремя пространственными координатами в зависимости от размерности задачи, для решения которой они предназначены. В динамическом анализе время рассматривается как дополнительная размерность.

Каждый элемент описывается множеством характерных точек, называемых узловыми точками, или узлами (для краткости). Узлы

предназначены для описания геометрии элемента и для задания физических степеней свободы (числа неизвестных функций). Узлы обычно находятся в угловых или крайних точках элемента, но могут быть также расположены между угловыми узлами и внутри элемента. Данное различие связано с порядком аппроксимации, который обеспечивает данный конечный элемент. Элементы, имеющие только угловые узлы, называются линейными и обеспечивают линейную интерполяцию геометрии и функций. Элементы, имеющие дополнительные узлы на своих границах между угловыми точками, могут обеспечивать квадратичную или даже кубическую интерполяцию.

Ансамблирование, или сборка, представляет собой объединение отдельных элементов в конечно-элементную сетку. С математической точки зрения ансамблирование состоит в объединении матриц жесткости отдельных элементов в одну глобальную матрицу жесткости всей конструкции. При этом используются две системы нумерации узлов элементов: локальная и глобальная. Локальная нумерация представляет собой фиксированную нумерацию узлов для каждого типа конечных элементов в соответствии с введенной локальной системой координат на элементе. Глобальная нумерация узлов всей конструкции может быть совершенно произвольной, как и глобальная нумерация конечных элементов. Однако между локальными номерами и глобальными номерами узлов существует взаимно однозначное соответствие, на основе которого и формируется глобальная система конечно-элементных уравнений.

Вопросы для самопроверки

1. Макромодель: определения и примеры.
2. Многоуровневое моделирование.
3. Суть организации программного обеспечения для многоуровневого моделирования.
4. Отличие расширенного базиса и традиционного метода узловых потенциалов.
5. Взаимосвязь моделирования электрических схем и механики.
6. Метод конечных элементов при решении ДУЧП.
7. Метод конечных разностей и его особенности.

Глава 7. АВТОМАТИЗАЦИЯ КОНСТРУКТОРСКОГО ПРОЕКТИРОВАНИЯ

7.1. Введение

Цель автоматизации конструкторского проектирования (АКП) – разработка конструкторской документации для изготовления изделий. АКП начинается с передачи принципиальных схем, разработанных на этапе функционального проектирования, в конструкторское подразделение, с помощью САПР. Математическими моделями описания конструкций являются структурные топологические ($2D$) и геометрические ($3D$) модели. Эти модели, а также методы конструирования составляют основу математического обеспечения АКП.

В основе конструирования лежит применение функционально-узлового (модульного) метода проектирования, с помощью которого выделяются конструктивные узлы (модули) различной степени сложности и соподчинённости. Распространению модульного принципа содействовали создание и использование интегральных микросхем, печатных и многослойных печатных плат.

При разработке конструкций в целом характерно восходящее проектирование [3]. Так, на базе интегральных микросхем и других навесных элементов разрабатываются типовые элементы замены (ТЭЗы, ячейки), затем – панели, рамы и, наконец, стойки ЭВА. В свою очередь на каждом из этих этапов проектирования последовательно решаются задачи компоновки элементов конструкции в узлы данного иерархического уровня, их размещения по конкретным установочным местам и трассировки соединений между элементами. Эта группа задач относится к коммутационно-монтажному проектированию и считается одной из основных в автоматизированных системах конструкторского проектирования (АСКП).

При разработке конструкций используется иерархический принцип проектирования сборочных единиц. Можно выделить следующие конструктивы:

- 1) *элемент* – поставляется как отдельное изделие, например интегральная микросхема (ИС), транзистор, диод, резистор и другие радиоэлектронные элементы;

2) *ячейка*, или *типовой элемент замены* (ТЭЗ), – это объединение элементов в едином конструктивном исполнении на основе печатной платы в виде микронэлектронного узла или микросборки;

3) *устройства* (панели, шкафы, субблоки), *блоки* (стойки) – объединяют различные ячейки и панели.

Каждый из конструктивов последующего уровня состоит из нескольких конструктивов предыдущего уровня.

Печатные платы – одни из самых распространенных конструктивных элементов. В зависимости от числа слоев печатных проводников печатные платы (ПП) могут быть одно-, двух- и многослойными. Элементы, устанавливаемые на печатных платах, различаются габаритами, числом и типом выводов, их расположением, различными типами корпусов. К элементам относятся соединители для подключения внешних цепей к электрической схеме печатной платы. Соединительные элементы монтажа ПП: токопроводящий рисунок (внешних и внутренних слоев), межслойные переходы, сквозные отверстия и иногда навесные токопроводящие шины.

Рисунок токопроводящих слоев состоит из контактных площадок и печатных проводников различной ширины. Различают контактные площадки прямоугольной формы для монтажа планарных выводов элементов и контактные площадки металлизированных сквозных отверстий для монтажа жестких штыревых выводов элементов.

Задачи конструкторского проектирования делятся на две группы:

1) определение геометрических параметров конструкции – *геометрическое проектирование*;

2) синтезирование структуры (топологии) конструкции с учётом её функциональных характеристик – *топологическое проектирование*.

Классификация задач конструкторского проектирования показана на рис. 7.1.

Задача *синтеза геометрических объектов* (ГО) состоит в формировании сложных ГО из элементарных ГО заданной структуры. Такая задача возникает, например, при оформлении детализированных чертежей.

Решение задачи *синтеза формы изделия* обеспечивает получение оптимальной или рациональной формы деталей, узлов или агрегатов, влияющей на качество функционирования объекта проектирования.



Рис. 7.1. Классификация задач конструкторского проектирования

К типовым *позиционным задачам* относят:

- определение координат точки пересечения прямой с криволинейным контуром или поверхностью;
- установление факта пересечения контуров и вычисление координат их точек пересечения;
- определение взаимного расположения плоских или пространственных областей.

К *метрическим задачам* относят:

- вычисление длины отрезка;
- вычисление периметра и площади плоских и пространственных областей;
- определение массы, моментов инерции и моментов сопротивления.

Задачи топологического проектирования:

– *компоновка* конструктивов i -го уровня в конструктив $(i - 1)$ -го уровня (компоновка простейших функциональных элементов логических схем И, ИЛИ, НЕ в корпуса микросхем, компоновка интегральных микросхем (ИМС) и других электрорадиоэлементов (ЭРЭ) в ячейки, ячеек в блоки и устройства). Это наиболее трудоёмкая часть конструкторского проектирования;

– *размещение* конструктивов i -го уровня в конструктивах $(i - 1)$ -го уровня (цифровых и аналоговых ИМС, ЭРЭ на печатной плате, ячеек в блоке и т. д.);

– *трассировка* монтажных соединений между конструктивами на всех уровнях (соединения между кристаллами ИМС, печатный монтаж, межплатные соединения).

Существуют два класса алгоритмов – конструктивные и итерационные:

1. *Конструктивные алгоритмы* формируют проектное решение за ряд последовательных шагов:

- выбирается один элемент схемы рассматриваемого уровня;
- к выбранному элементу по определенным правилам присоединяется второй;
- к полученному добавляется третий и т. д.

2. *Итерационные алгоритмы* требуют задания начального приближения, которое затем улучшается. Начальное решение задается инженером-проектировщиком (пользователем САПР) или является результатом работы конструктивного алгоритма.

Возможно использование алгоритмов совместного решения задач компоновки и размещения, размещения и трассировки, компоновки и трассировки.

7.2. Задача компоновки

Первая задача АКП – задача компоновки, при которой определяется однозначное соответствие между функциональным и конструктивным делением проектируемого устройства [2]. Задача компоновки включает в себя два аспекта:

- покрытие функциональной схемы узла схемой соединений типовых конструктивных элементов;

– разбиение схемы соединений типовых элементов на подсхемы с целью компоновки конструктивных узлов более высокого уровня иерархии, например распределение микросхем по ТЭЗ.

Задачи компоновки можно представить в виде сложных комбинаторных моделей, для которых в большинстве случаев отсутствуют эффективные точные методы, поэтому применяют эффективные приближенные алгоритмы ее решения. Два основных класса задач компоновки представлены на рис. 7.2.

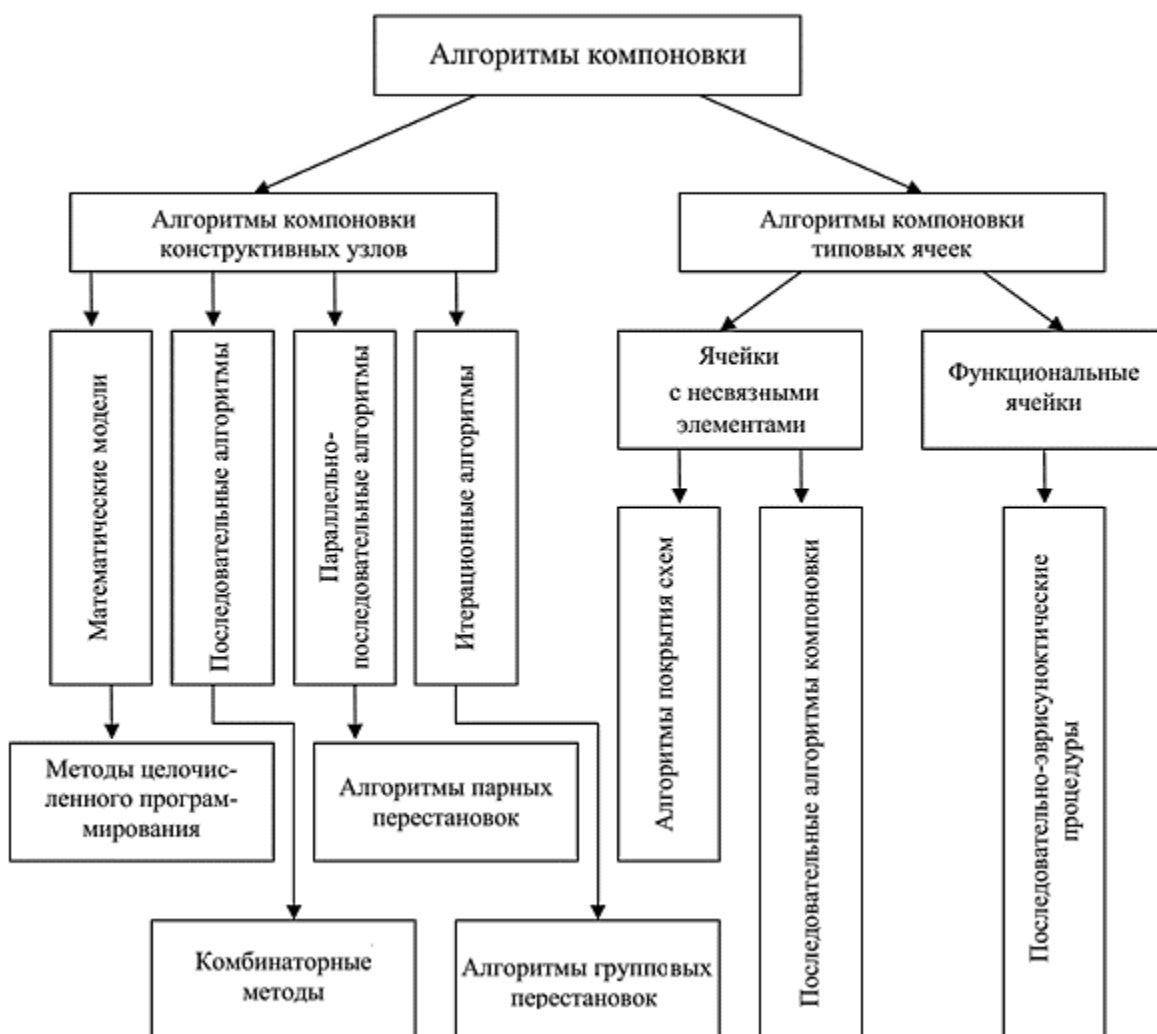


Рис. 7.2. Алгоритмы компоновки

Конструктивные алгоритмы первой группы можно классифицировать по критериям оптимизации, ограничениям на формирование узлов или структуре вычислительной процедуры.

В качестве основных критериев оптимизации определяем минимум числа узлов и межузловых соединений, а ограничениями – количество элементов в узле и число внешних выводов на узле.

Алгоритмы компоновки конструктивных узлов можно разделить на последовательные, параллельно-последовательные и итерационные. В алгоритмах первого типа вводится последовательный процесс компоновки узлов, на каждом шаге которого в очередной узел добавляется один из элементов схемы (например, алгоритм компоновки по связности). В параллельно-последовательных алгоритмах сначала выделяется некоторое исходное количество групп элементов, которые потом распределяют по узлам с учётом ограничений и критериев на компоновку.

Последовательные и параллельно-последовательные алгоритмы применяют для создания базового (начального) варианта компоновки при заданных ограничениях на число элементов в узле и число выводов на узле.

Итерационные алгоритмы компоновки служат для улучшения некоторого начального распределения в соответствии с принятыми критериями и используются в сочетании с другими алгоритмами компоновки. Пусть имеется некоторый начальный вариант компоновки, полученный либо одним из последовательных алгоритмов, либо вручную. Основа рассматриваемых алгоритмов компоновки – использование итерационного процесса обменов местами элементов или групп элементов, принадлежащих различным узлам, с целью минимизации некоторого критерия.

Главная задача алгоритмов компоновки типовых узлов – получение покрытия с минимальной суммарной стоимостью (минимум числа использованных типовых узлов).

7.3. Задача размещения

Вторая задача АКП – задача размещения, т. е. определение точного местоположения типовых конструктивных элементов в монтажном пространстве конструктивного узла более высокого уровня иерархии, например размещение микросхем на плате. Основные группы алгоритмов размещения представлены на рис. 7.3:

1) алгоритмы решения математических задач, являющиеся моделями задачи размещения;

- 2) конструктивные алгоритмы начального размещения;
- 3) итерационные алгоритмы улучшения начального варианта размещения;
- 4) непрерывно-дискретные методы размещения.

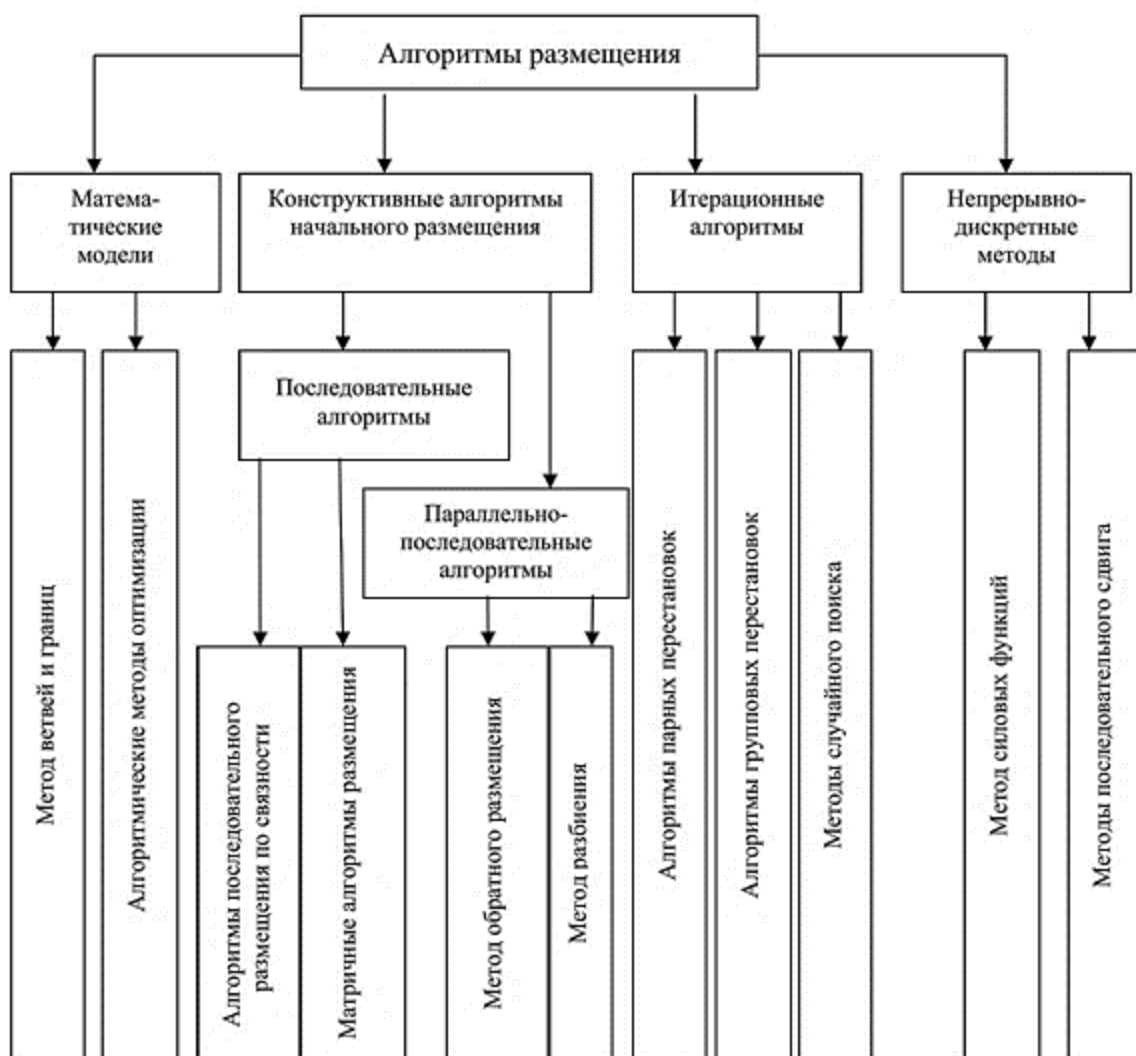


Рис. 7.3. Алгоритмы размещения

К первой группе относятся, прежде всего, метод ветвей и границ для задачи квадратичного назначения, к которой при определённых упрощениях сводится задача размещения: набор позиций считается фиксированным, элементы рассматриваются как геометрические точки, схема соединений представляется взвешенным графом. Другой класс моделей связан с оптимизацией размещения, когда набор позиций для установки заранее не фиксирован.

Вторая и третья группы включают приближённые алгоритмы, в основном предназначенные для оптимизации размещения элементов в фиксированном наборе позиций.

Характерной чертой конструктивных алгоритмов является то, что они создают размещение. Тогда как итерационные алгоритмы предполагают задание начального размещения. Конструктивные алгоритмы используют последовательный или параллельно-последовательный процесс установки элементов в позиции при локальной оптимизации критерия размещения.

В итерационных алгоритмах производится перерасположение элементов или их групп с целью минимизации выбранного критерия. Эти алгоритмы требуют существенных затрат машинного времени и используются для получения конечного размещения.

7.4. Задачи трассировки

Трассировка соединений – заключительный этап конструкторского проектирования электронных устройств, который состоит в определении линий, соединяющих контакты элементов, и компонентов проектируемого устройства. Задача трассировки – одна из наиболее трудоемких в общей проблеме АКП. С математических позиций трассировка – наисложнейшая задача выбора из огромного числа вариантов оптимального решения.

Одновременная оптимизация всех соединений при трассировке за счет перебора всех вариантов в настоящее время невозможна. Поэтому разрабатываются в основном локально оптимальные методы трассировки, когда трасса оптимальна лишь на данном шаге при наличии ранее проведенных соединений.

Основная задача трассировки формулируется следующим образом: по заданной схеме соединений прокладываются необходимые проводники на плоскости (плате, кристалле и т. д.), чтобы реализовать заданные технические соединения с учетом заранее заданных ограничений (основными считаются ограничения на ширину проводников и минимальные расстояния между ними).

Исходной информацией для решения задачи трассировки соединений обычно выступают список цепей, параметры конструкции элементов и коммутационного поля, а также данные по размещению элементов. Критериями трассировки могут быть процент реализованных

соединений, суммарная длина проводников, число пересечений проводников, число монтажных слоев, число межслойных переходов, равномерность распределения проводников, минимальная область трассировки и т. д. Часто эти критерии являются взаимоисключающими, поэтому качество трассировки оценивается по доминирующему критерию при выполнении ограничений по другим критериям либо применяют аддитивную или мультипликативную формы оценочной функции.

Алгоритмы трассировки печатных плат можно условно разбить на три большие группы.

1. Волновые алгоритмы, основанные на идеях Ли. Данные алгоритмы получили широкое распространение в существующих САПР, поскольку они позволяют легко учитывать технологическую специфику печатного монтажа со своей совокупностью конструктивных ограничений. Эти алгоритмы всегда гарантируют построение трассы, если для нее существует путь.

2. Ортогональные алгоритмы, обладающие бóльшим быстродействием, чем алгоритмы первой группы. Их реализация на ЭВМ требует в 75 – 100 раз меньше вычислений по сравнению с волновыми алгоритмами. Такие алгоритмы применяют при проектировании печатных плат со сквозными металлизированными отверстиями. Недостатки этой группы алгоритмов связаны с получением большого числа переходов со слоя на слой, отсутствием 100%-й гарантии проведения трасс, большим числом параллельно идущих проводников.

3. Алгоритмы эвристического типа. Эти алгоритмы частично основаны на эвристическом приеме поиска пути в лабиринте. При этом каждое соединение проводится по кратчайшему пути, обходя встречающиеся препятствия.

Волновой алгоритм Ли. Все ячейки монтажного поля подразделяют на занятые и свободные. Занятыми считаются ячейки, в которых уже расположены проводники, построенные на предыдущих шагах, или находятся монтажные выводы элементов, а также ячейки, соответствующие границе платы и запрещенным для прокладывания проводников участкам. Каждый раз при проведении новой трассы можно использовать лишь свободные ячейки, число которых по мере проведения трасс сокращается.

На множестве свободных ячеек коммутационного поля моделируют волну влияния из одной ячейки в другую, соединяемых впоследствии общим проводником. Первую ячейку, в которой зарождается волна влияний, называют источником, а вторую – приемником волны. Для того чтобы иметь возможность следить за прохождением фронта волны влияний, его фрагментам на каждом этапе присваивают некоторые веса:

$$P_k = P_{k-1} + \varphi(f_1, f_2, \dots, f_g),$$

где P_k и P_{k-1} – веса ячеек k -го и $(k-1)$ -го фронтов; $\varphi(f_1, f_2, \dots, f_g)$ – весовая функция, являющаяся показателем качества проведения пути, каждый параметр которой $f_i (i = 1, 2, \dots, g)$ характеризует путь с точки зрения одного из критериев качества (длины пути, числа пересечений и т. п.). На P_k накладывают одно ограничение – веса ячеек предыдущих фронтов не должны быть больше весов ячеек последующих фронтов. Фронт распространяется только на соседние ячейки, которые имеют с ячейками предыдущего фронта либо общую сторону, либо хотя бы одну общую точку. Процесс распространения волны продолжается до тех пор, пока её расширяющийся фронт не достигнет приемника или на каком-то шаге не найдется ни одной свободной ячейки, которая могла бы быть включена в очередной фронт, что соответствует случаю невозможности проведения трассы при заданных ограничениях.

Если в результате распространения волна достигла приемника, то выполняется «проведение пути», которое заключается в движении от приемника к источнику по пройденным на этапе распространения волны ячейкам, следя за тем, чтобы значения P_k монотонно убывали. В итоге получают путь, соединяющий эти две точки. Из описания алгоритма следует, что все условия, необходимые для проведения пути, закладываются в правила присвоения веса ячейкам.

Для того чтобы исключить неопределенность при проведении пути для случая, когда несколько ячеек имеют одинаковый минимальный вес, вводят понятие путевых координат, задающих предпочтительность проведения трассы. Каждое направление кодируют двоичным числом по $\text{mod } q$, где q – число просматриваемых соседних ячеек. Чем предпочтительнее то или иное направление, тем меньший числовой код оно имеет. Например, если задаться приоритетным порядком проведения пути сверху, справа, снизу и слева, то коды соответствующих

путевых координат будут 00, 01, 10 и 11. Путевые координаты присваиваются на этапе распространения волны.

Этапы волнового алгоритма [3]:

1. На масштабной сетке коммутационного поля (КП) отмечаются занятые и свободные дискреты. Трасса может проходить только через свободные дискреты; все ранее проложенные проводники через дискреты и контактные площадки – это занятые дискреты.

2. Вводится весовая функция $F(f_1, \dots, f_n)$ как критерий качества пути (например, f_1 – допустимая длина, f_2 – число пересечений, f_3 – число изгибов, f_n – число переходных отверстий).

3. Распространение волны для каждой цепи от начала А (источника) во все стороны до конца В (приемника) заключается в присвоении дискретам, которые располагаются рядом, определенного значения весовой функции F . Процесс распространения волны продолжается до тех пор, пока расширяющийся фронт волны не достигает дискрета В (конца цепи). Это означает, что путь построить можно. Если на очередном этапе распространения волны не окажется ни одного свободного дискрета, то путь построить нельзя.

4. Построение трассы от дискрета В до А по пройденным дискретам. Если несколько соседних дискретов имеют одинаковый вес, то используются весовые координаты (приоритеты поворотов: вверх, направо, вниз, налево), которые указывают предпочтительность направления трассы. Рассмотрим алгоритм на примере рис. 7.4.

Выберем точку А (5, 10) в качестве источника, точку В (14, 3) – в качестве приемника. Вес дискрета А равен 0. Соседним дискретам присвоим веса, равные 1, 2, 3 и т. д. (Для присвоения весов дискретам достаточно трех весов 1, 2, 3.). Распространим волну во все стороны от клетки А до тех пор, пока фронт не достигнет точки В. Волна распространяется только по свободным дискретам (на рис. 7.4 запрещенные дискреты помечены символом *). В обратном направлении построим путь от точки В до точки А, проходя дискреты в обратном направлении по весовым значениям 3–2–1–3–2–1 и так далее при условии приоритетов: вверх, влево, вниз, вправо (построенный путь отмечен жирным шрифтом).

2	1	3	2	1	2	3	1	2	3	1	2	3	1	2	3	1
1	3	2	1	3	1	2	3	1	2	3	1	2	3	1	2	3
3	2	1	3	2	3	1	2	*	3	1	2	3	1	2	3	1
2	1	3	2	1	2	3	1	*	1	2	3	1	2	3	1	2
1	3	2	1	A	1	2	3	*	2	3	1	2	3	1	2	3
2	1	3	2	1	2	3	1	*	3	1	2	3	1	2	3	1
3	2	1	3	2	3	1	2	*	1	2	3	1	2	3	1	2
1	3	2	1	3	1	2	3	*	3	1	2	3	1	2	3	1
2	1	3	2	1	*	*	*	*	2	3	1	2	3	1	2	3
3	2	1	3	2	3	1	2	3	1	2	3	1	2	3	1	2
1	3	2	1	3	1	2	3	1	2	3	*	2	3	1	2	3
2	1	3	2	1	2	3	1	2	3	1	*	3	B	2	3	1
3	2	1	3	2	3	1	2	3	*	*	*	1	2	3	1	2
1	3	2	1	3	1	2	3	1	2	3	1	2	3	1	2	3
2	1	3	2	1	2	3	1	2	3	1	2	3	1	2	3	1

Рис. 7.4. Иллюстрация работы волнового алгоритма

Следует отметить, что на распространение волны затрачивается около 90 % времени вычислений, а на проведение цепи – 10 %. Существенными недостатками волнового алгоритма являются малое быстроедействие и большой объем памяти, необходимый для хранения информации о текущем состоянии всех ячеек коммутационного поля, возможность построения лишь соединений типа «ввод-вывод». Попытки устранить указанные недостатки привели к созданию ряда модификаций волнового алгоритма.

Лучевой алгоритм трассировки предусматривает исследование свободных клеток КП по заранее заданным направлениям – подобно лучам. Для построения пути от точки А до точки В распространяют лучи одновременно из обоих источников до встречи их в некоторой точке С. Обычно выбирают два луча, идущих в двух направлениях из каждой точки (рис. 7.5). С помощью лучевых алгоритмов можно про-

ложить 50 – 60 % трасс, остальные – с помощью волнового алгоритма. Лучевой алгоритм экономичнее волнового по затратам машинного времени.

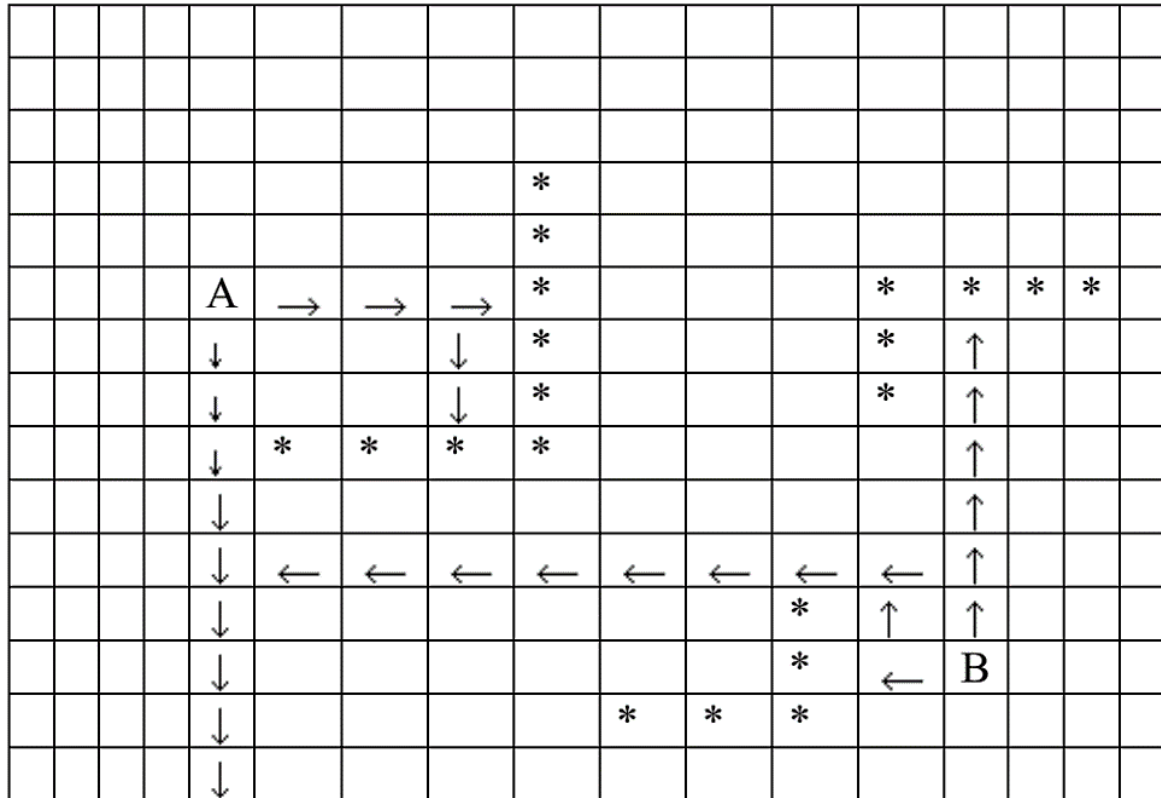


Рис. 7.5. Иллюстрация работы лучевого алгоритма

Канальные алгоритмы трассировки основаны на прокладке трасс по укрупненным дискретам коммутационного поля, представляющим собой каналы. Канал характеризуется пропускной способностью – допустимым количеством проходящих через него магистралей. Каналы образуются, как правило, между рядами установленных элементов или компонентов. В вертикальных и горизонтальных каналах трассировка осуществляется в различных слоях КП. Канальные алгоритмы основаны на определении максимального потока в сети с ограниченными пропускными способностями каналов.

На рис. 7.6 – 7.10 представлены пример принципиальной электрической схемы устройства, результаты размещения элементов этой схемы на печатной плате и результаты трассировки с помощью системы проектирования печатных плат *P-CAD* [56].

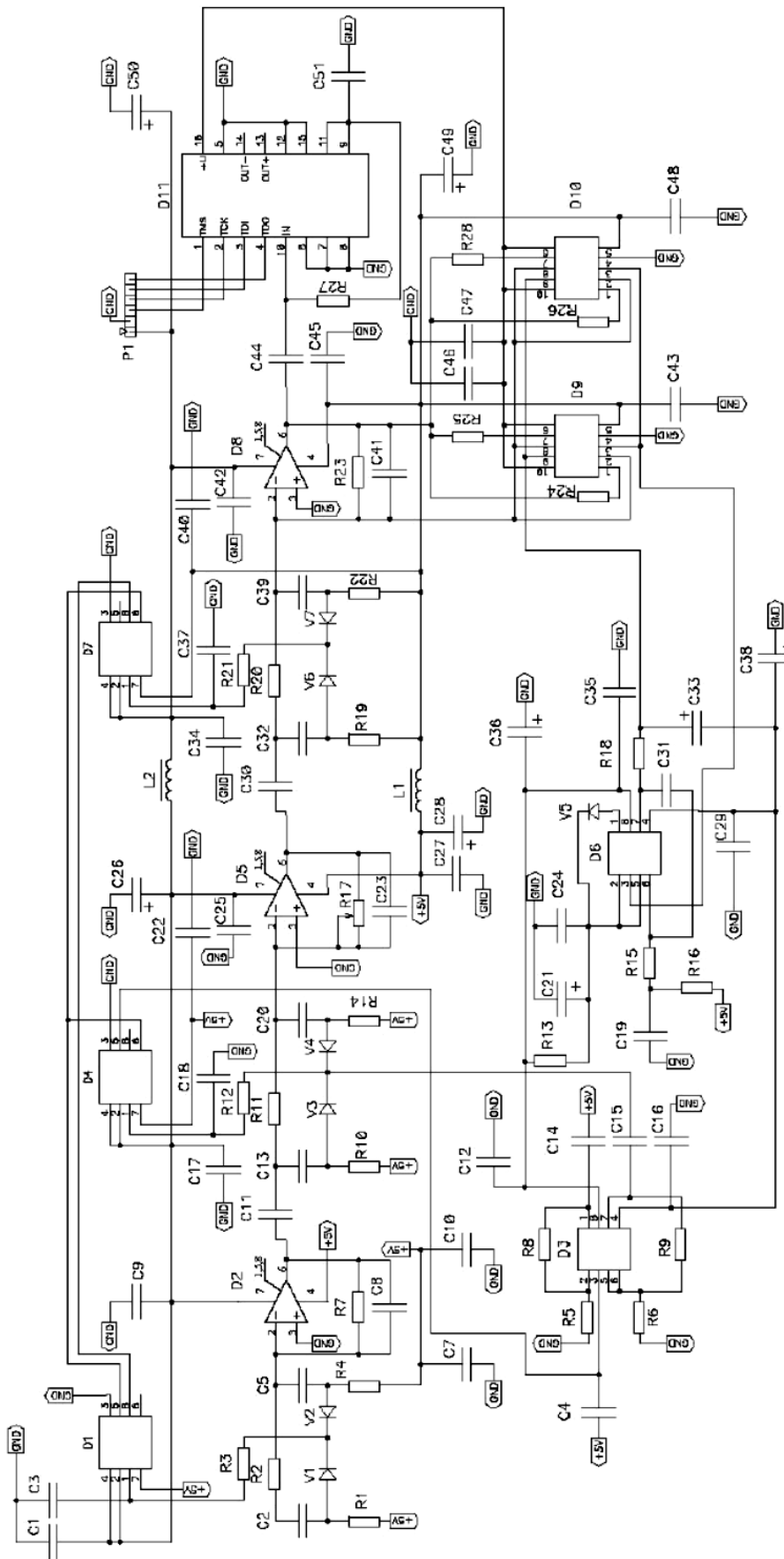


Рис. 7.6. Принципиальная электрическая схема

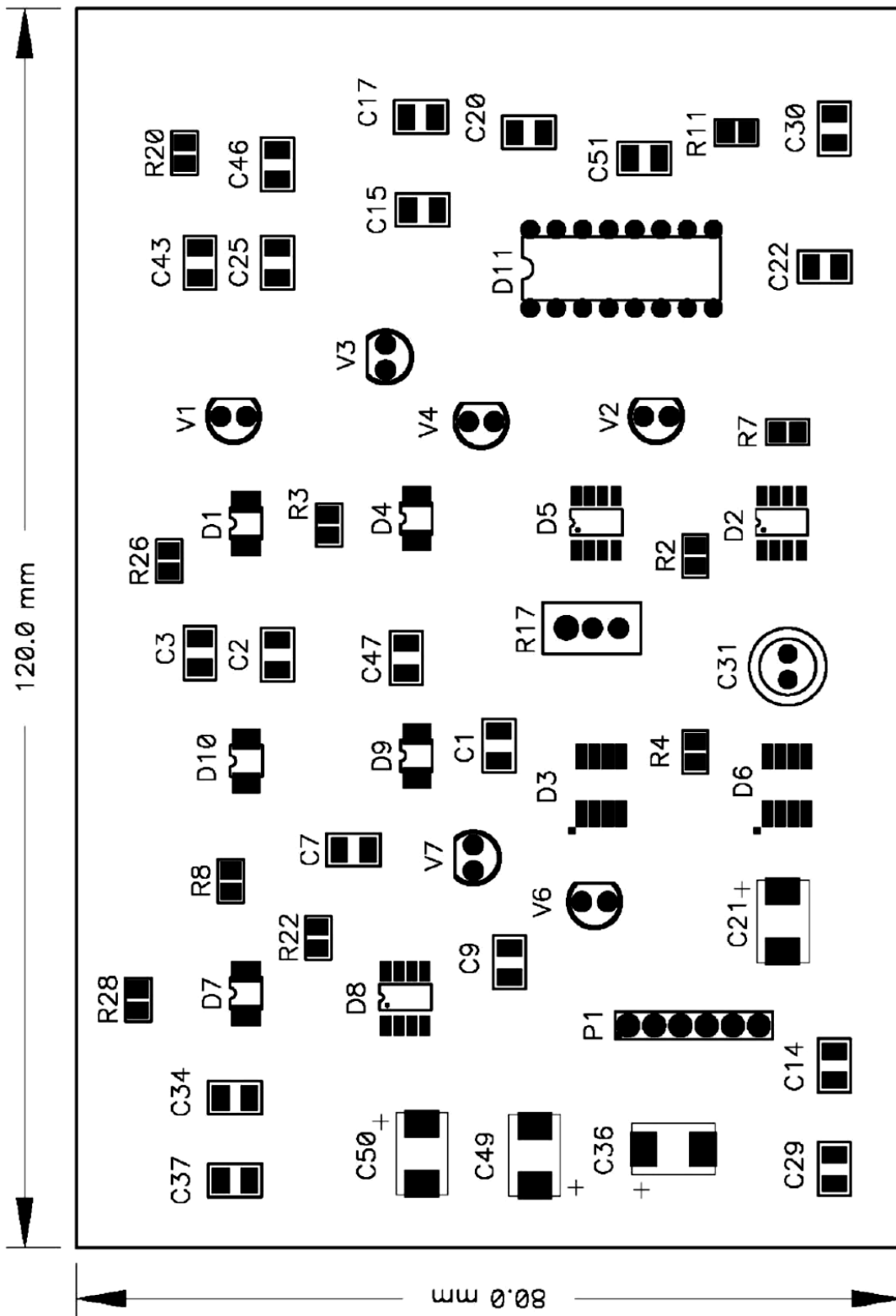


Рис. 7.7. Размещение компонентов (верхний слой)

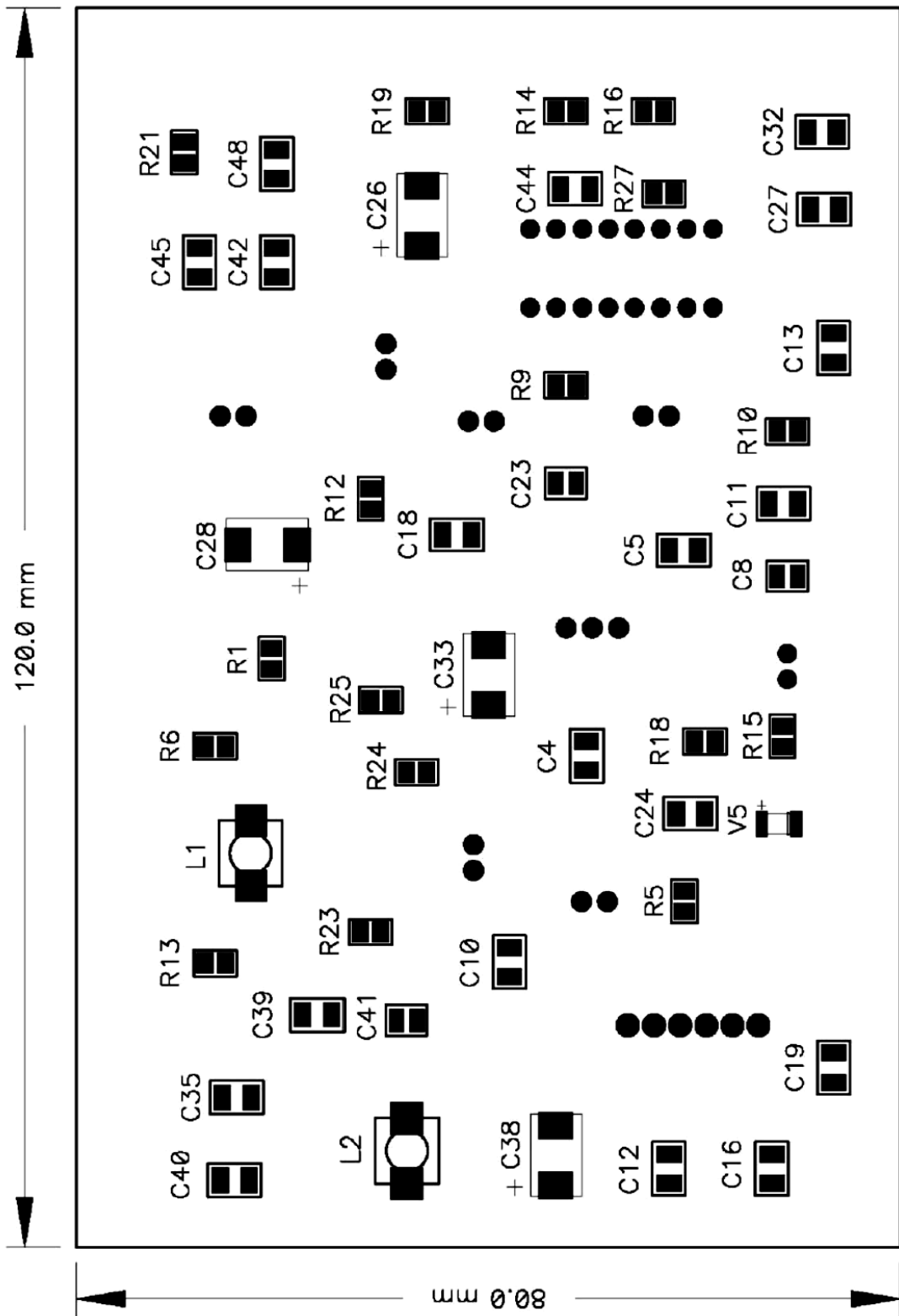


Рис. 7.8. Размещение компонентов (нижний слой)

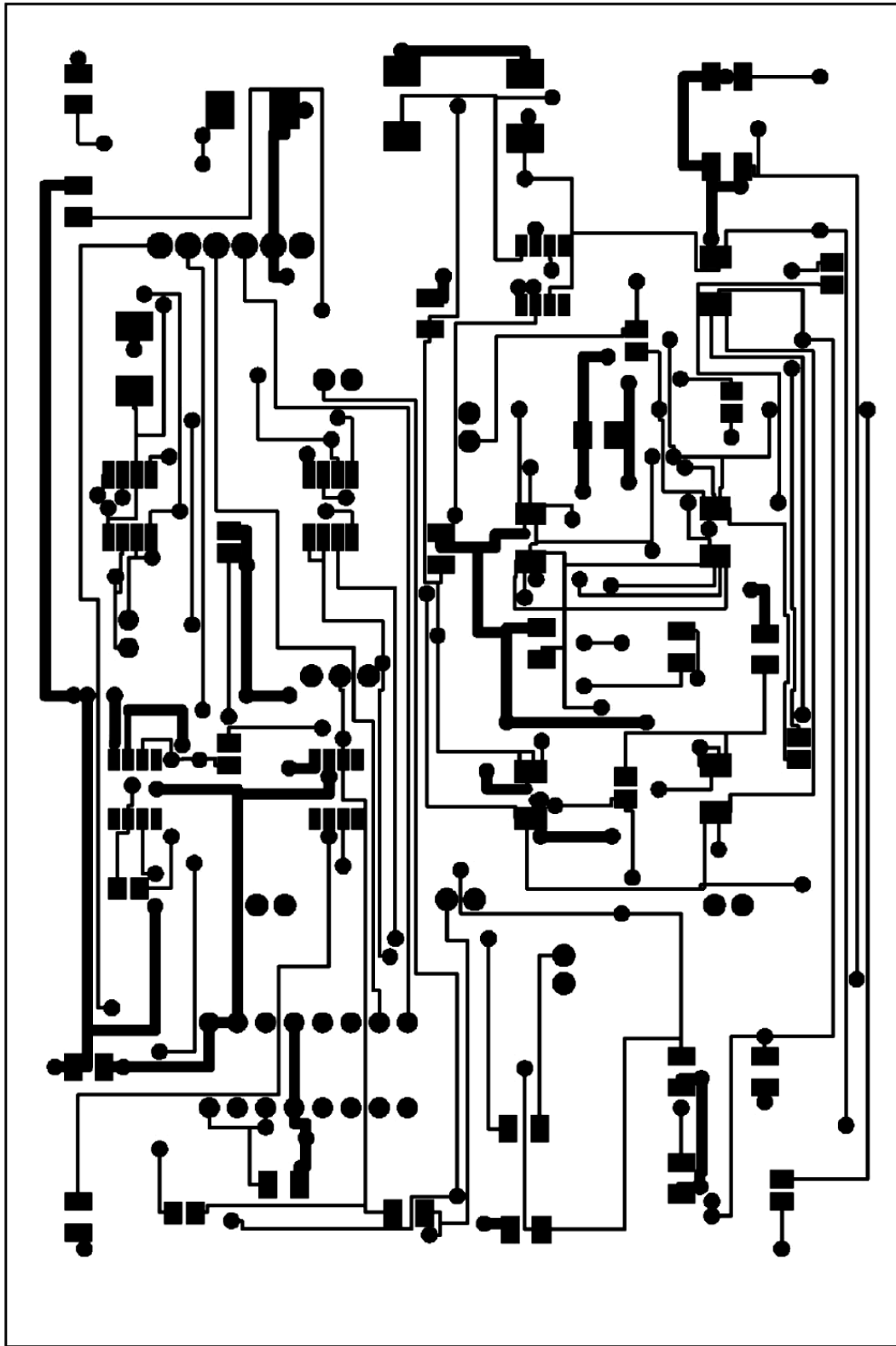


Рис. 7.9. Проводники (верхний слой)

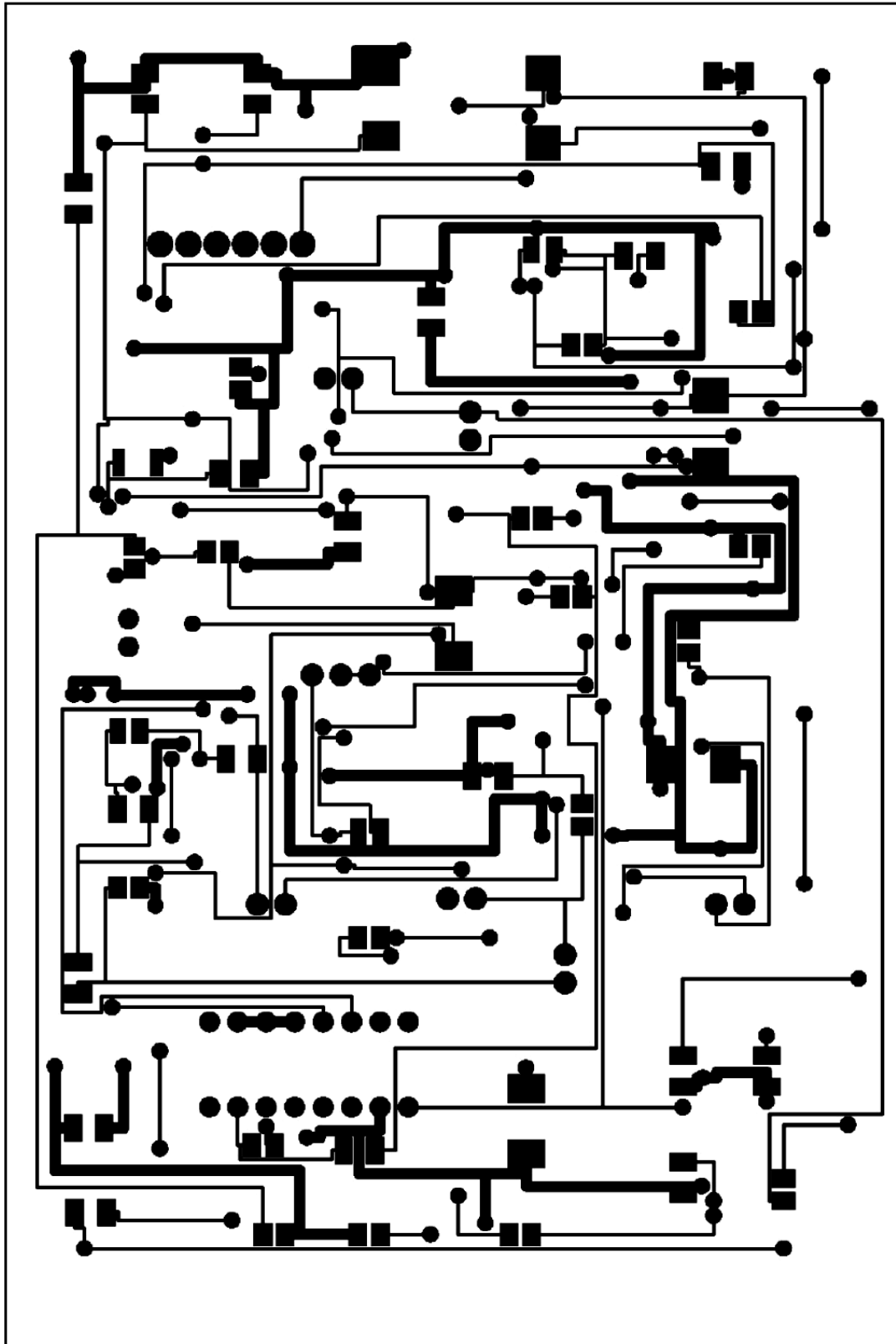


Рис. 7.10. Проводники (нижний слой)

Вопросы для самопроверки

1. Задачи конструкторского проектирования.
2. Задачи топологического проектирования.
3. Задачи компоновки.
4. Задачи размещения.
5. Задачи трассировки.
6. Этапы волнового алгоритма трассировки.
7. Суть лучевого алгоритма трассировки.
8. Суть канального алгоритма трассировки.

ЗАКЛЮЧЕНИЕ

В учебном пособии представлены ключевые разделы курса «Основы автоматизации проектирования». Основное внимание уделено автоматизации проектирования ЭВА: рассмотрены этапы и уровни ее проектирования, структура САПР для ЭВА, схема процесса проектирования, а также виды обеспечения САПР, особенно математическое обеспечение.

Проектированию ЭВА на системном и функциональном уровнях отведена отдельная глава, в которой описаны методы моделирования и проектирования цифровых схем и устройств, а также функциональных схем аналоговых устройств.

В пособии также проанализированы особенности автоматизации схемотехнического проектирования. Подробно рассмотрены модели элементов на схемотехническом уровне, вопросы формирования модели схемы, методы моделирования схем в статическом, малосигнальном и переходном режимах.

Значительное внимание в пособии уделено дополнительным видам анализа на схемотехническом уровне, в частности анализу чувствительностей, шумов, нелинейных искажений и др.

Заключительная часть пособия посвящена автоматизации конструкторского проектирования, а также задачам размещения и трассировки элементов схем на печатных платах.

СПИСОК БИБЛИОГРАФИЧЕСКИХ ССЫЛОК

1. Норенков И. П. Введение в автоматизированное проектирование технических устройств и систем. М. : Высш. шк., 1980. 311 с.
2. Норенков И. П., Маничев В. Б. Системы автоматизированного проектирования электронной и вычислительной аппаратуры. М. : Высш. шк., 1983. 272 с.
3. Норенков И. П., Маничев В. Б. Основы теории и проектирования САПР. М. : Высш. шк., 1990. 335 с.
4. Норенков И. П. Основы автоматизированного проектирования. М. : Изд-во МГТУ им. Н. Э. Баумана, 2000. 360 с.
5. Системы автоматизированного проектирования : учеб. пособие : в 9 кн. / под ред. И. П. Норенкова. М. : Высш. шк., 1986.
6. Системы автоматизированного проектирования в радиоэлектронике : справочник / под ред. И. П. Норенкова. М. : Радио и связь, 1986. 368 с.
7. Ильин В. Н. Машинное проектирование электронных схем. М. : Энергия, 1972. 280 с.
8. Ильин В. Н. Основы автоматизации схемотехнического проектирования. М. : Энергия, 1979. 392 с.
9. Ильин В. Н., Коган В. Л. Разработка и применение программ автоматизации схемотехнического проектирования. М. : Радио и связь, 1984. 368 с.
10. Автоматизация схемотехнического проектирования / под ред. В. Н. Ильина. М. : Радио и связь, 1987. 368 с.
11. Корячко В. П., Курейчик В. М., Норенков И. П. Теоретические основы САПР. М. : Энергоатомиздат, 1987. 400 с.
12. Муромцев Д. Ю., Тюрин И. В. Математическое обеспечение САПР. СПб. : Лань, 2014. 464 с.
13. Калахан Д. Методы машинного расчета электронных схем. М. : Мир, 1970. 344 с.
14. Аббасов М. Э. Методы оптимизации : учеб. пособие. СПб. : ВВМ, 2014. 64 с.

15. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах. М. : Наука, 1975. 320 с.
16. Васильев Ф. П. Методы оптимизации. М. : Факториал Пресс, 2002. 824 с.
17. Ланцов В. Н. Моделирование : в 2 ч. Ч. 1. Владимир : Владимирский гос. ун-т, 1999. 88 с.
18. Ланцов В. Н. Моделирование : в 2 ч. Ч. 2. Владимир : Владимирский гос. ун-т, 2001. 72 с.
19. Автоматизация проектирования аналого-цифровых устройств / под ред. Э. И. Гитиса. М. : Энергоатомиздат, 1987. 184 с.
20. Автоматизированное проектирование цифровых устройств / под ред. С. С. Бадулина. М. : Радио и связь, 1981. 240 с.
21. Ланцов В. Н. Основы автоматизации схемотехнического проектирования. Владимир : ВлГТУ, 1996. 88 с.
22. Борисов Ю. П. Математическое моделирование радиосистем. М. : Сов. радио, 1976. 296 с.
23. Автоматизация проектирования БИС : в 6 кн. / под ред. Г. Г. Казеннова. М. : Высш. шк., 1990.
24. Анисимов Б. В., Белов Б. И., Норенков И. П. Машинный расчет элементов ЭВМ. М. : Высш. шк., 1976. 336 с.
25. Баталов Б. В., Егоров Ю. Б., Русаков С. Г. Основы математического моделирования больших интегральных схем на ЭВМ. М. : Радио и связь, 1982. 168 с.
26. Белов Б. И., Норенков И. П. Расчет электронных схем на ЭЦВМ. М. : Машиностроение, 1971. 143 с.
27. Влах И., Сингхал К. Машинные методы анализа и проектирования электронных схем. М. : Радио и связь, 1988. 560 с.
28. Глориозов Е. Л., Ссорин В. Г., Сыпчук П. П. Введение в автоматизацию схемотехнического проектирования. М. : Сов. радио, 1976. 234 с.
29. Диалоговые системы схемотехнического проектирования / под ред. В. И. Анисимова. М. : Радио и связь, 1988. 288 с.

30. Калабеков Б. А., Лapidус В. Ю., Малафеев В. М. Методы автоматизированного расчета электронных схем в технике связи. М. : Радио и связь, 1990. 272 с.
31. Калниболотский Ю. М., Сундучков К. С., Солодовник А. И. Автоматизированное проектирование электронных схем. Киев : Техніка, 1987. 301 с.
32. Машинный расчет интегральных схем / под ред. Дж. Д. Герсковица. М. : Мир, 1971. 407 с.
33. Петренко А. И., Власов А. И., Тимченко А. П. Табличные методы моделирования электронных схем на ЭЦВМ. Киев : Вища шк., 1977. 192 с.
34. Анализ электронных схем на ЭЦВМ / А. И. Петренко [и др.]. Львов : Вища шк., 1975. 194 с.
35. Программирование систем автоматизированного проектирования РЭА / В. Г. Слипченко [и др.]. Киев : Выща шк., 1986. 255 с.
36. Разевиг В. Д. Применение программ P-CAD и PSpice для схемотехнического моделирования на ПЭВМ. М. : Радио и связь, 1992. 80 с.
37. Разевиг В. Д. Система схемотехнического моделирования Micro-CAP V. М. : СОЛОН, 1997. 273 с.
38. Сигорский В. П., Петренко А. И. Алгоритмы анализа электронных схем. Киев : Техніка, 1976. 608 с.
39. Слипченко В. Г. Организация системы автоматизированного моделирования электронных схем на ЭВМ. Киев : Вища шк., 1978. 176 с.
40. Фидлер Дж., Найтингейм К. Машинное проектирование электронных схем. М. : Высш. шк., 1985. 215 с.
41. Чахмахсаян Е. А., Бармаков Ю. И., Гольденберг А. Э. Машинный анализ интегральных схем. М. : Сов. радио, 1974. 269 с.
42. Чахмахсаян Е. А., Мозговой Г. П., Силин В. Д. Математическое моделирование и макро моделирование биполярных элементов электронных схем. М. : Радио и связь, 1985. 144 с.
43. Чуа Л. О., Пен-Мин Лин. Машинный анализ электронных схем. М. : Энергия, 1980. 640 с.

44. Денбновецкий С. В., Писаренко Л. Д., Резниченко В. К. Основы автоматизированного проектирования электронных приборов. Киев : Выща шк., 1987. 336 с.
45. Носов Ю. Р., Петросянц К. О., Шилин В. А. Математические модели элементов интегральной электроники. М. : Сов. радио, 1976. 304 с.
46. Автоматизированное проектирование устройств СВЧ / под ред. В. В. Никольского. М. : Радио и связь, 1982. 272 с.
47. Нефедов Е. И., Фиалковский А. Т. Полосковые линии передачи. М. : Наука, 1980. 312 с.
48. Богданович Б. М. Нелинейные искажения в приемно-усилительных устройствах. М. : Связь, 1980. 280 с.
49. Моделирование и оптимизация на ЭВМ радиоэлектронных устройств / под ред. З. М. Бененсона. М. : Радио и связь, 1981. 272 с.
50. Ризкин И. Х. Машинный анализ и проектирование технических систем. М. : Наука, 1985. 160 с.
51. Бондаренко В. М. Вопросы анализа нелинейных электрических и электронных цепей. Киев : Наук. думка, 1967. 159 с.
52. Автоматизация проектирования радиоэлектронных средств / под ред. О. В. Алексеева. М. : Высш. шк., 2000. 479 с.
53. Макромоделирование аналоговых ИС / А. Г. Алексенко [и др.]. М. : Радио и связь, 1983. 248 с.
54. Армстронг Дж. Р. Моделирование цифровых систем на языке VHDL. М. : Мир, 1992. 175 с.
55. Ланцов В. Н. Методы понижения порядка моделей сложных систем. Владимир : Изд-во ВлГУ, 2017. 84 с.
56. Разевиг В. Д. Проектирование печатных плат в P-CAD 2001. М. : СОЛОН, 2003. 560 с.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Норенков, И. П. Системы автоматизированного проектирования электронной и вычислительной аппаратуры / И. П. Норенков, В. Б. Маничев. – М. : Высш. шк., 1983. – 272 с.
2. Норенков, И. П. Основы теории и проектирования САПР / И. П. Норенков, В. Б. Маничев. – М. : Высш. шк., 1990. – 335 с.
3. Норенков, И. П. Основы автоматизированного проектирования / И. П. Норенков. – М. : Изд-во МГТУ им. Н. Э. Баумана, 2000. – 360 с.
4. Автоматизация схемотехнического проектирования / под ред. В. Н. Ильина. – М. : Радио и связь, 1987. – 368 с.
5. Корячко, В. П. Теоретические основы САПР / В. П. Корячко, В. М. Курейчик, И. П. Норенков. – М. : Энергоатомиздат, 1987. – 400 с.
6. Влах, И. Машинные методы анализа и проектирования электронных схем / И. Влах, К. Сингхал. – М. : Радио и связь, 1988. – 560 с.
7. Чуа, Л. О. Машинный анализ электронных схем / Л. О. Чуа, Пен-Мин Лин. – М. : Энергия, 1980. – 640 с.
8. Автоматизация проектирования радиоэлектронных средств : учеб. пособие / под ред. О. В. Алексеева. – М. : Высш. шк., 2000. – 479 с.
9. Ланцов, В. Н. Проектирование заказных микросхем на КМОП : учеб. пособие / В. Н. Ланцов. – Владимир : Изд-во ВлГУ, 2009. – 221 с. – ISBN 978-5-89368-941-9.
10. Ланцов, В. Н. Современные подходы к проектированию и тестированию интегральных микросхем : монография / В. Н. Ланцов, С. Г. Мосин. – Владимир : Изд-во ВлГУ, 2010. – 285 с. – ISBN 978-5-9984-0120-6.

Учебное издание

ЛАНЦОВ Владимир Николаевич

ОСНОВЫ АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

Учебное пособие

Редактор Е. А. Лебедева

Технический редактор Ш. Ш. Амирсейидов

Компьютерная верстка Л. В. Макаровой

Корректор О. В. Балашова

Выпускающий редактор А. А. Амирсейидова

Подписано в печать 21.09.23.

Формат 60×84/16. Усл. печ. л. 10,7. Тираж 30 экз.

Заказ

Издательство

Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых.
600000, Владимир, ул. Горького, 87.