

АВТОМАТИЗАЦИЯ СХЕМОТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Под редакцией профессора В. Н. Ильина

*Допущено Министерством высшего
и среднего специального образования СССР
в качестве учебного пособия для студентов
радиотехнических специальностей вузов*



Москва
«Радио и связь»
1987

ББК 32.973

А 22

УДК 681.14

В. Н. Ильин, В. Т. Фролкин, А. И. Бутко, Н. Ю. Камнева, Е. М. Тихомирова

Автоматизация схемотехнического проектирования:
А 22 Учеб. пособие для вузов/В. Н. Ильин, В. Т. Фролкин, А. И. Бутко и др.; Под ред. В. Н. Ильина. — М.: Радио и связь, 1987. — 368 с.: ил.

Даны основные сведения о составе и принципах функционирования системы автоматизированного проектирования радиоэлектронной аппаратуры. Описаны методы моделирования радиоэлектронной аппаратуры на различных этапах автоматизированного проектирования — от структурного до конструкторского. Изложены методы расчета больших интегральных схем, моделирования радиотехнических схем в режимах малого и большого сигналов, макро-моделирования типовых микросхем. Рассмотрены критерии и способы непрерывной и дискретной оптимизации и синтеза на ЭВМ.

Для студентов вузов радиотехнических специальностей.

А 2402010000-122
046(01)-87 62-87

ББК 32.973

Рецензенты: кафедра проектирования и конструирования интегральных микросхем Московского института электронной техники (зав. кафедрой д-р техн. наук, проф. Г. Г. Казеннов) и д-р техн. наук, проф. И. И. Шагурин

Редакция литературы по вычислительной технике

Учебное пособие

*Валерий Николаевич Ильин, Виктор Тихонович Фролкин,
Анатолий Иванович Бутко, Наталья Юрьевна Камнева,
Елена Марковна Тихомирова*

АВТОМАТИЗАЦИЯ СХЕМОТЕХНИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Заведующая редакцией Г. И. Козырева

Редактор Н. Г. Давыдова

Художественный редактор Н. С. Шенн

Переплет художника Ю. В. Архангельского

Технический редактор Г. И. Колосова

Корректор О. П. Иванецкая

ИБ № 739

Сдано в набор 10.12.86

Подписано в печать 15.04.87

Г. 10434

Формат 60×90/16

Бумага кн.-журн. № 2

Гарнитура литературная

Печать высокая Усл. печ. л. 23,0 Усл. кр.-отт. 23,0 Уч.-изд. л. 25,32 Тираж 15 000 экз.

Изд. № 20495

Зак. № 133

Цена 1 р. 30 к.

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

Московская типография № 5 ВГО «Союзучетиздат». 101000 Москва, ул. Кирова, д. 40

© Издательство «Радио и связь», 1987

ПРЕДИСЛОВИЕ

Как указано в решениях XXVII съезда КПСС, автоматизированное проектирование радиоэлектронной аппаратуры является одним из основных рычагов ускорения научно-технического прогресса. Важнейший этап автоматизированного проектирования — это схемотехническое проектирование.

Содержание понятия «схемотехническое проектирование» сформировалось в ту пору, когда элементную базу радиоэлектронной аппаратуры (РЭА) составляли дискретные элементы — резисторы, конденсаторы, корпусированные транзисторы и диоды, а основной технологией производства схем был навесной монтаж. В этих условиях автоматизация схемотехнического проектирования сводилась к расчету на ЭВМ электрических процессов в линейных и нелинейных цепях с сосредоточенными параметрами.

В настоящее время в связи с прогрессом интегральной технологии и ростом функциональной сложности дискретных элементов понятие «схемотехническое проектирование» существенно расширилось. Инженеру-схемотехнику при разработке интегральных схем (ИС) приходится не только рассчитывать электрические процессы, но и проектировать функциональные, а иногда и структурные схемы. Увеличение степени интеграции привело также к тесной связи электрического расчета с разработкой конструкции и топологии ИС. Кроме того, для устройств, состоящих из большого числа элементов, расчет на ЭВМ на электрическом уровне практически невыполним из-за больших затрат времени, и необходимо применение методов упрощенного расчета на логическом или функциональном уровне. В связи с этим в данное учебное пособие наряду с методами автоматизации электрического расчета включены методы автоматизации структурного, функционального и логического моделирования, а также конструкторского проектирования.

Учебное пособие построено следующим образом. В начальной части (гл. 1, 2) изложены общие сведения о процессах автоматизированного схемотехнического проектирования, особенностях их реализации на основе систем автоматизированного проектирования (САПР), а также основные понятия САПР.

В основной части (гл. 3—11) описаны методы, алгоритмы, входные языки и программы, используемые на разных уровнях автоматизированного схемотехнического проектирования. Особое

внимание уделено традиционному электрическому уровню, в частности — методам моделирования нелинейных радиотехнических схем, сравнительно мало освещенным в литературе, а также таким важным методам расчета схем, как макро моделирование и электрологическое моделирование.

Главы 12—16 посвящены так называемым инвариантным задачам схемотехнического проектирования, не зависящим от конкретного типа схем, — расчету, анализу, непрерывной и дискретной оптимизации и синтезу. Изложены также основные приемы конструкторского проектирования.

Данное учебное пособие предназначено для студентов радиотехнических специальностей вузов и заводов-вузов. Оно может быть полезно инженерно-техническим работникам, занимающимся автоматизированным проектированием РЭА, и в первую очередь — пользователям САПР РЭА, так как в книге основное внимание уделяется описанию принципов работы алгоритмов, а не деталям организации программ. По мнению авторов, знание этих принципов необходимо для эффективного применения САПР РЭА, включая грамотную постановку задачи проектирования РЭА на ЭВМ и правильный выбор алгоритмов и программ для ее решения с помощью САПР.

Главы 1, 2, 4, 7—9 написаны В. Н. Ильиным, гл. 3, 5, 6 — А. И. Бутко, гл. 12—15 — Н. Ю. Камневой, гл. 10, 11 — совместно Е. М. Тихомировой и В. Т. Фролкиным, гл. 16 написана Е. М. Тихомировой.

ОБЩИЕ СВЕДЕНИЯ ОБ ОБЪЕКТАХ И ЗАДАЧАХ ПРОЕКТИРОВАНИЯ

1.1. КЛАССИФИКАЦИЯ ПАРАМЕТРОВ И ЗАДАЧ ПРОЕКТИРОВАНИЯ

1.1.1. Классификация параметров. Прежде чем излагать методы автоматизации схемотехнического проектирования, определим некоторые общие понятия, относящиеся к проектированию любых устройств независимо от их физической природы.

Параметры элементов, из которых состоит проектируемое устройство, будем называть *внутренними*, параметры устройства, по которым оценивается его качество, — *выходными*, параметры действующих на устройство внешних информационных сигналов — *входными*, а параметры окружающей среды — *внешними*.

Пример 1.1. Применительно к проектированию транзисторного усилителя параметры транзисторов — внутренние параметры; потребляемая мощность, коэффициент нелинейных искажений — численные выходные параметры; амплитудно- и фазочастотные характеристики — функциональные выходные параметры (выходные характеристики); частота и амплитуда гармонического входного сигнала — численные входные параметры; спектральная характеристика входного сигнала — функциональный входной параметр; температура окружающей среды — внешний параметр.

1.1.2. Основные задачи проектирования. Схемотехническое проектирование включает решение задач расчета, анализа, оптимизации и синтеза. Эти задачи называются проектными процедурами и имеют следующее содержание.

Расчет — определение выходных параметров и характеристик устройства при неизменных значениях его внутренних параметров и постоянной структуре.

Анализ — определение изменения выходных параметров и характеристик устройства в зависимости от изменения его внутренних и входных параметров. В случае применения ЭВМ задача расчета часто называется *одновариантным* анализом, а задача анализа — *многовариантным* анализом.

Оптимизация — определение наилучших в том или ином смысле значений выходных параметров и характеристик путем целенаправленного изменения внутренних параметров устройства (при *параметрической оптимизации*) или структуры устройства (при *структурной оптимизации*). Внутренние параметры, за счет изменения которых выполняется параметрическая оптимизация, называются *варьируемыми*. Способ изменения варьируемых параметров определяется конкретным алгоритмом оптимизации. В качестве варьируемых следует выбирать *управляемые* внутренние пара-

метры, значение которых легко изменять и контролировать в процессе производства.

Наиболее сложными являются задачи *параметрического и структурного синтеза*. В общем случае синтезом называется генерация исходного варианта устройства, включая его структуру (структурный синтез) и значения внутренних параметров (параметрический синтез). Генерация может выполняться различными способами — выбором из уже известных устройств, построением на основе теоретических соотношений, путем изобретательства и др. Полученное в результате синтеза устройство не обязательно должно быть наилучшим, но обязательно работоспособным. Если же полученное устройство — наилучшее в каком-либо смысле, то такой синтез называется *оптимальным*. Однако процедуры оптимального синтеза разработаны лишь для узкого класса устройств, преимущественно линейного типа, например линейных схем с заданной структурой или линейных систем автоматического управления. В большинстве случаев устройство, полученное в результате синтеза, даже оптимального, требует доработки, чтобы удовлетворить многочисленным требованиям, учесть которые на стадии синтеза невозможно из-за их многообразия и сложности.

Вследствие этого процесс проектирования практически всегда включает наряду с синтезом решение всех перечисленных выше задач, как показано на рис. 1.1, где цепочка блоков 2-3-4-5-6-2 соответствует циклу параметрической оптимизации, а цепочка блоков 2-3-4-5-7-8 — циклу структурной оптимизации, например введением дополнительных положительных или отрицательных обратных связей.

Пример 1.2. При проектировании низкочастотного усилителя определение его рабочей точки по постоянному току, коэффициента нелинейных искажений, амплитудно-частотной характеристики, потребляемой мощности входят в задачу расчета; определение температурной стабильности рабочей точки на постоянном токе, влияния разброса внутренних параметров на выходные параметры — в задачу анализа; определение числа каскадов и способов их включения (с общим эмиттером, общим коллектором) — в задачу структурного синтеза; выбор параметров элементов каждого каскада (резисторов, транзисторов) — в задачу параметрического синтеза; расширение полосы пропускания путем снижения коэффициентов усиления каскадов — в задачу параметрической оптимизации, а путем введения дополнительных отрицательных обратных связей — в задачу структурной оптимизации.

1.1.3. Способы проектирования. При проектировании РЭА используют различные методы. Основными из них являются:

неавтоматизированный расчет по заранее полученным формулам;

физическое моделирование, т. е. исследование объектов одной физической природы с помощью объектов, имеющих другую физическую природу, но одинаковое с первыми математическое описание; в основе физического моделирования лежит обычно принцип электрофизических аналогий;

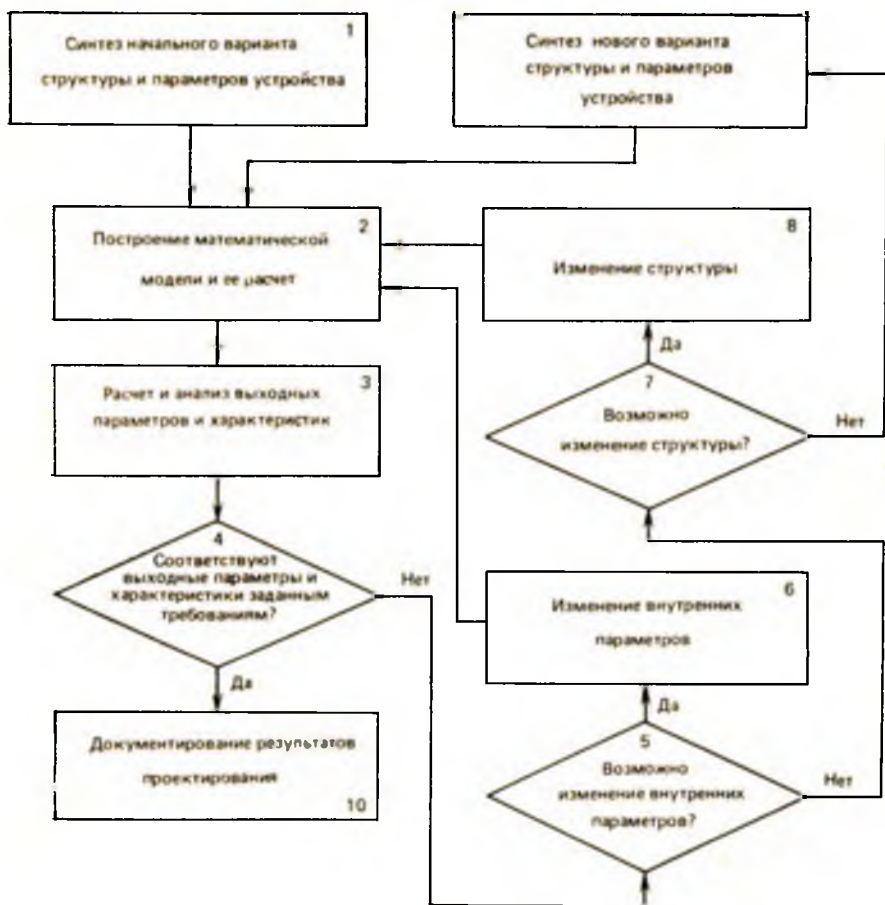


Рис. 1.1. Типовая схема отдельного этапа проектирования

натурное макетирование;
математическое моделирование на ЭВМ.

Недостатки *неавтоматизированного расчета* — низкая точность, ограниченные функциональные возможности и т. д. — общеизвестны и не нуждаются в обсуждении.

Физическое моделирование как способ проектирования РЭА используется довольно редко, чаще его применяют для изучения сопутствующих работе схем тепловых и других процессов, математическое моделирование которых слишком сложно и трудоемко.

Натурное макетирование — один из наиболее старых и распространенных способов проектирования РЭА. Его главное достоинство — максимальная достоверность результатов, обусловленная работой с реальными схемами, а не их приближенными

моделями. Кроме того, макетирование привлекает наглядностью получаемых результатов. В то же время макетирование имеет ряд крупных недостатков. Основные из них — высокая стоимость, длительность создания макета, ограниченные возможности макетирования.

Под математическим моделированием на ЭВМ обычно понимается весь комплекс вопросов, связанных с составлением математической модели устройства (см. § 1.5) и ее использованием на ЭВМ в процедурах расчета, анализа, оптимизации и синтеза (см. рис. 1.1). По сравнению с макетированием математическое моделирование на ЭВМ имеет следующие преимущества.

В задачах расчета с помощью модели можно найти выходные параметры схем или их характеристики, которые нельзя непосредственно измерить на макете из-за недоступности точек измерения, что особенно характерно для интегральных схем.

В задачах анализа моделирование позволяет проанализировать выходные параметры и характеристики схемы в предельных и запредельных режимах, физическая реализация которых опасна для макета. Кроме того, моделирование позволяет выполнить, например, расчет серийнопригодности и анализ различных статистических характеристик схемы без ее запуска в серию, анализ воздействия на схему внешних условий без реальных климатических и других испытаний, анализ нереализуемых на макете зависимостей выходных параметров схемы от внутренних, например зависимостей выходных параметров схемы от внутренних параметров транзистора.

В задачах оптимизации возможности макета ограничены небольшим числом регулировочных элементов, тогда как в модели можно варьировать любые управляемые параметры, добиваясь максимального улучшения выходных параметров.

Роль моделирования в задачах синтеза состоит в проверке правильности функционирования синтезированных схем путем расчета их математических моделей. Очевидно, это можно сделать гораздо быстрее, чем выполнить макетирование каждой синтезированной схемы.

Вместе с тем следует заметить, что нельзя полностью заменить макетирование моделированием. Перечисленные преимущества моделирования имеют место лишь в том случае, если проектировщик располагает хорошо отработанным комплексом программ, содержащим проверенные, достаточно точные модели элементов РЭА, гарантирующие достоверность результатов проектирования. Если же необходимо спроектировать новую оригинальную схему, для расчета которой имеющийся комплекс программ по каким-либо причинам непригоден, то макетирование этой схемы может оказаться более приемлемым, чем разработка новой программы.

Кроме того, даже проектируя схему путем ее моделирования, разработчики схем часто проверяют окончательные результаты проектирования на опытном макете, однако, разумеется, такой

способ проектирования нельзя назвать макетированием, хотя в конце концов и приходится иметь дело с макетом.

Отметим в заключение, что, как и макетирование, определенную роль в проектировании РЭА продолжают играть и традиционные неавтоматизированные методы. Ведь моделировать можно только то, что предварительно синтезировано (см. блок 1 на рис. 1.1). Довольно часто неавтоматизированный синтез первоначального варианта РЭА оказывается быстрее и проще автоматизированного.

Таким образом, реальный процесс автоматизированного проектирования РЭА обычно состоит из двух этапов:

неавтоматизированного синтеза структуры и эскизного, обычно тоже неавтоматизированного, по упрощенным формулам расчета ее параметров с целью получения работоспособного варианта РЭА, играющего роль начального приближения; в настоящее время ведутся работы по автоматизации этого этапа проектирования;

доводки полученного варианта до кондиций, соответствующих техническому заданию (ТЗ), с помощью программ автоматизированного проектирования.

1.2. УРОВНИ СЛОЖНОСТИ РЭА И УРОВНИ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

1.2.1. Иерархия уровней сложности. Радиоэлектронная аппаратура весьма разнообразна по сложности и содержит от единиц элементов (простейшие схемы) до 10^{10} элементов (сложные вычислительные и другие комплексы). Иерархия уровней сложности РЭА представлена в первой колонке табл. 1.1.

Следует отметить, что конкретное содержание понятий функциональный элемент (ФЭ), функциональный узел (ФУ), функциональное устройство (ФУС), функциональный комплекс (ФК), указанное в табл. 1.1, постоянно изменяется в соответствии с развитием технологии и методов проектирования РЭА. Так, в период ламповой и транзисторной техники триггер рассматривался как ФУ, а регистр — как ФУС, с переходом же на интегральную технологию ранг сложности этих устройств понизился. С появлением БИС и СБИС понижается ранг сложности таких ФУС, как микропроцессор, который часто рассматривают уже как ФУ.

1.2.2. Иерархия уровней проектирования. Очевидно, нельзя РЭА разной степени сложности анализировать и проектировать с одинаковой степенью детализации, например, на уровне процессов, протекающих в каждом транзисторе. Поэтому проектирование сложных РЭА представляет многоэтапный иерархический процесс. Содержание и число этапов проектирования зависит от многих причин, основными из которых являются особенности объекта и конечные цели его проектирования, используемая элементная база, имеющийся математический аппарат, возможности технологии и т. д.

Иерархия уровней сложности РЭА	Иерархия функциональных уровней автоматизированного проектирования		Математический аппарат	
	Уровень	Подуровень	Теоретические методы	Численные методы реализации
Функциональные комплексы (радиоизмерительные и радиоуправляющие системы, вычислительные системы и др.)	Автоматизированное системное проектирование (АСП)		Системный анализ, исследование операций, теория игр, теория массового обслуживания и др.	Численные методы исследования непрерывной и дискретной оптимизации, статистическое моделирование
	Автоматизированное структурное проектирование (АСТП)	Информационный Точностный	Спектральный анализ, теория автоматического управления и регулирования, теория цифровых автоматов, алгебра логики и др.	Численные методы моделирования и преобразования сигналов
Функциональные устройства (исердатчики, приемники, микропроцессоры, запоминающие устройства и др.)	Автоматизированное функциональное и логическое проектирование (АФЛП)	Функциональный Регистровый Логический		
	Автоматизированное схемотехническое проектирование (АСХП)	Макромодельный Элементный	Теория электрических цепей с сосредоточенными параметрами	Численные методы решения конечных и обыкновенных дифференциальных уравнений
Функциональные элементы (генераторы, усилители, детекторы, фильтры, триггеры, логические элементы и др.)	Автоматизированное проектирование компонентов (АПК)	Модели с сосредоточенными параметрами Модели с распределенными параметрами	Методы математической физики и физики твердого тела	Численные методы решения уравнений в частных производных
	Компоненты (транзисторы, диоды, резисторы, конденсаторы, полупроводниковые структуры и др.)			

Одним из основных факторов, определяющих структуру процесса проектирования, является ограниченность возможности выполнения большого количества расчетов, даже при использовании мощных ЭВМ. Так, максимальный размер электронной схемы, в которой можно рассчитать электрические процессы с помощью современных ЭВМ и программ автоматизированного проектирования, составляет несколько сотен транзисторов. В связи с этим процесс проектирования разбивается на уровни, каждому из которых соответствует решение определенных задач, характерных для каждого уровня. Классификация уровней, сложившаяся в автоматизированном проектировании РЭА, представлена во второй колонке табл. 1.1. Рассмотрим кратко содержание этих уровней.

На уровне *системного проектирования* изучается взаимодействие проектируемого объекта с окружающей средой.

На уровне *структурного проектирования* определяются типы ФУС, образующих функциональный комплекс (ФК), и структура связей между ними, обеспечивающая заданные информационные и точностные характеристики ФК.

На уровне *функционального* (для цифровых устройств его называют функционально-логическим) *проектирования* обеспечивается выполнение РЭА типа ФУС и ФУ своего функционального назначения (например, выдача заданной последовательности выходных сигналов и временных соотношений между ними) на основе знания приближенной или идеализированной формы входных и внутренних сигналов.

На уровне *схемотехнического проектирования* прорабатывается форма сигналов для отдельных ФУ или ФЭ, а также рассчитываются уточненные значения их внутренних и выходных параметров.

Наконец, на уровне *проектирования компонентов* определяют параметры конструкции, материалов и технологий, обеспечивающие заданные характеристики отдельных компонентов — диодов, транзисторов и т. д.

Легко заметить преемственность соседних уровней проектирования. Так, точные параметры компонентов нужны для точного определения формы сигналов в схемотехническом проектировании. Точная форма сигналов в свою очередь необходима для определения точности функционирования в функциональном проектировании и т. д.

Указанные в табл. 1.1 уровни проектирования можно реализовать на основе двух подходов — информационного и физического. При *информационном подходе* определяется лишь преобразование входного сигнала в выходной без изучения внутренних физических процессов и без учета физических законов сохранения или равновесия, определяющих или сопровождающих это преобразование. В связи с этим все информационные модели имеют вид «черного ящика» с однонаправленным прохождением информации, для которого известна функция или алгоритм F преобразования входного сигнала x в выходной y : $y = F(x)$. Информационный под-

ход используется обычно при системном, структурном и функциональном проектировании.

При *физическом подходе* проектирование выполняется с учетом реальных физических законов, например законов равновесия (законов Кирхгофа для электрических цепей) или законов сохранения (уравнений непрерывности для активных электрических компонентов). Эти законы играют роль ограничений-равенств, которые нужно соблюдать в каждый момент времени, что приводит к необходимости численного решения уравнений равновесия или сохранения. В связи с этим время моделирования на ЭВМ устройств при физическом подходе существенно (на несколько порядков) превышает время моделирования на ЭВМ устройств при информационном подходе. Соответственно, сложность РЭА, допускающей автоматизированное проектирование на основе физического подхода, на несколько порядков ниже, чем на основе информационного. Иногда при информационном подходе используются модели, составленные на основе физического подхода, но с исключенными внутренними переменными, так что модель приобретает вид $y = F(x)$, характерный для информационных моделей, однако это удается сделать лишь в частном случае простейших линейных моделей.

Важной особенностью табл. 1.1 является смещенность относительно друг друга иерархий уровней сложности и уровней проектирования, отражающая возможность использования одного и того же уровня проектирования для устройств разной сложности. При этом для каждого уровня проектирования вводятся два подуровня — один для РЭА большей сложности и второй для РЭА меньшей сложности. Например, структурное проектирование имеет подуровни информационного проектирования для ФК и точностного проектирования для ФУС, функциональное проектирование — подуровень того же названия (в случае цифровых устройств он называется регистровым, или уровнем регистровых передач) для ФУС (радиоприемные устройства, микропроцессоры, ЭВМ) и логический — для проектирования более простых цифровых устройств типа ФУ (счетчики, мультиплексоры и др.) и т. д.

Следует отметить, что на верхнем подуровне каждого уровня проектирования обычно используется тот же математический аппарат, что и на нижнем, однако применяются более простые, огрубленные или агрегированные модели отдельных элементов, из которых строится модель всего устройства. Это позволяет моделировать на данном уровне проектирования с использованием одного и того же математического аппарата заметно более сложные устройства. Например, применение макромоделей на уровне АСхП позволяет моделировать схемы на два-три порядка сложнее схем, рассчитываемых теми же методами схемотехнического проектирования на основе обычных моделей элементов.

1.2.3. Математический аппарат. Отметим некоторые особенности математического аппарата, используемого на разных уровнях проектирования (третья колонка табл. 1.1). Прежде всего следу-

ет сказать о том, что математический аппарат для каждого уровня проектирования состоит из двух частей. С одной стороны, он включает теоретические методы (или концепции), разработанные для конкретных задач проектирования (системный анализ, спектральный анализ, теория электронных цепей и т. д.). С другой стороны, каждый теоретический метод можно реализовать по-разному, например на основе аналитических преобразований или численных методов.

Автоматизированное проектирование в процессе своего развития выработало для каждого теоретического метода наиболее эффективные методы их численной реализации, которые и называются обычно методами автоматизированного проектирования. Следует отметить, что эти методы универсальны, они носят инвариантный характер и обычно ориентированы не на конкретный объект, а на определенный тип задач. Например, для решения на ЭВМ задач теории массового обслуживания (структурный уровень) в качестве основного обычно используется метод статистического моделирования (метод Монте-Карло). Этот же метод применяется и на других уровнях проектирования, например в схемотехническом проектировании при расчете допусков, однако здесь он играет лишь вспомогательную роль.

Аналогично следует оценивать роль и других методов численной реализации.

На уровне функционально-логического проектирования в качестве теоретических методов применяются два математических аппарата, соответствующих двум типам устройств — цифровым и аналоговым, различие между которыми проявляется именно на функциональном уровне. Для функционально-логического проектирования цифровых радиоэлектронных устройств (РЭУ) используется аппарат алгебры логики в виде теории комбинационных устройств для проектирования логических схем, не запоминающих информацию, и теории конечных автоматов для схем с памятью. Для функционального проектирования аналоговых РЭУ радиотехнического типа (приемники, передатчики и т. д.), а также аналого-цифровых РЭУ время-импульсного типа (преобразователи информации) общего математического аппарата не существует, а используются различные частные способы, отражающие специфику решаемых задач и объектов проектирования. Каждый из этих способов порождает свой численный метод расчета на ЭВМ выходного сигнала по заданному типу входного.

В качестве примера можно привести способ быстрого преобразования Фурье для численного решения задач спектрального анализа или способ численного расчета интеграла Дюамеля для определения выходных сигналов линейных РЭУ во временной области. Совокупность этих и других способов можно назвать методом моделирования и преобразования сигналов.

1.3. ТИПЫ ОБЪЕКТОВ ПРОЕКТИРОВАНИЯ

Обычно все РЭУ, проектируемые на ЭВМ, разделяют на три типа: цифровые, аналоговые и аналого-цифровые.

К *цифровым* относят устройства, рабочие сигналы которых закодированы в виде чисел, обычно представляемых в двоичном коде цифрами 0 и 1 (триггеры, счетчики, регистры и т. д.). При проектировании цифровых устройств наибольший интерес представляют временные соотношения между моментами появления двоичных цифр или чисел и сами эти числа, меньшее значение имеет форма электрических сигналов, соответствующих каждой цифре или числу. Поэтому при проектировании цифровых устройств обычно используют методы функционального моделирования — регистрового (гл. 5), для сложных устройств типа специализированных вычислителей, ЭВМ, и логического (гл. 6) — для более простых, типа счетчиков, арифметико-логических устройств и др. При этом рассчитываются временные диаграммы, определяющие соответствие между моментами появления тех или иных двоичных цифр или чисел в разных точках устройства, т. е. проверяется согласованность работы различных блоков в составе одного устройства. Если данный подход не обеспечивает необходимой точности, то приходится применять более точные методы схемотехнического моделирования (гл. 7, 8), позволяющие исследовать структуру сигналов, возникающих при смене цифр, например переходные процессы при переключении устройства из состояния 0 в состояние 1 или наоборот.

К *аналоговым* относят устройства, информация о работе которых заключена в различных характеристиках сигнала — форме, спектре и т. д. (усилители, генераторы, преобразователи формы и параметров сигнала и др.). Для анализа этих устройств часто необходимо использовать более точные, чем функциональное моделирование, методы (гл. 7—9). При этом для анализа видеочастотных схем используются методы, основанные на составлении и решении систем обыкновенных дифференциальных уравнений, а для анализа радиочастотных схем с высокочастотным несущим сигналом, модулированным по амплитуде, частоте или фазе — специальные методы (гл. 9). К этому же классу относятся схемы СВЧ и оптического диапазона, однако методы их расчета на ЭВМ разработаны пока недостаточно полно.

Наконец, широкий класс образуют *аналого-цифровые устройства* — разного типа преобразователи аналог — код, код — аналог, специализированные вычислители и т. д. В связи с развитием цифровых методов обработки аналоговой информации разработка методов расчета на ЭВМ этого класса устройств особенно актуальна. Здесь широко применяются гибридные методы моделирования на ЭВМ, когда аналоговые элементы, где нужна повышенная точность расчета, моделируются методами схемотехнического проектирования, т. е. на основе физического подхода, а цифровые элементы, где высокая точность не нужна, — методами функционального и логичес-

кого проектирования с широким использованием макромоделей. В целом такой подход получил название электрологического или функционально-электро-табличного (ФЭТ) моделирования (гл. 8, 11).

1.4. ТИПЫ ПРОЦЕССОВ ПРОЕКТИРОВАНИЯ

Проектирование сложной РЭА носит многоэтапный характер, на каждом этапе проектируются РЭА определенного уровня сложности (ФЭ, ФУ, ФУС) на том или ином функциональном уровне проектирования (вторая колонка табл. 1.1), а каждый функциональный уровень проектирования включает решение задач синтеза (С), анализа (А), расчета (Р), оптимизации (О) и выпуска документации (Д) (см. рис. 1.1). Совокупность этих этапов в их взаимодействии образует процесс проектирования.

Теоретически можно различать два основных типа процессов проектирования: сверху вниз (от ФУС к ФЭ) и снизу вверх (от ФЭ к ФУС). Практически процесс проектирования носит комбинированный характер: с одной стороны, проектирование сложной РЭА типа систем и устройств ведется сверху вниз, с другой, — проектирование относительно простой РЭА типа элементов и узлов ведется снизу вверх. На некотором промежуточном этапе эти два независимых процесса встречаются и производится процесс «покрытия» требований к ФУ и ФЭ, выдвигаемых процессом сверху вниз номенклатурой тех ФУ и ФЭ, которые изготовлены по процессу снизу вверх. Граница, где встречаются эти процессы, не-

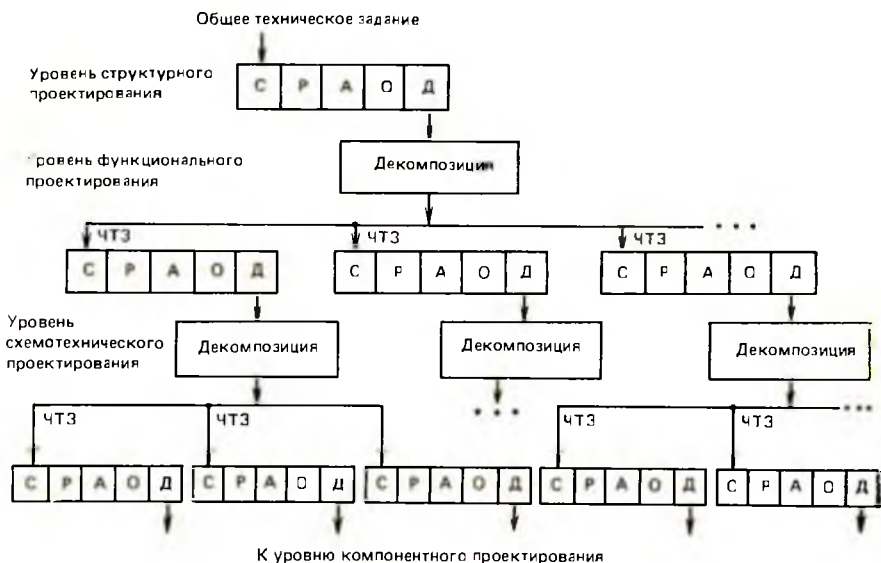


Рис. 1.2. Типовая иерархическая схема проектирования сложного объекта

постоянна и зависит от типа РЭА, уровня технологии, сложившейся методики проектирования и т. д.

В процессе проектирования сверху вниз РЭА высшего уровня сложности разбивается (декомпозируется) на более простые устройства (рис. 1.2). При этом для каждого устройства формируется частное техническое задание (ЧТЗ), обеспечиваемое на более нижних уровнях проектирования. Далее процесс декомпозиции и формирования ЧТЗ повторяется для более простых устройств. Поскольку конкретно сформулировать частные задачи на основе более общих легче, чем общие на основе частных, то проектирование сверху вниз предпочтительнее, чем снизу вверх.

1.5. ПОНЯТИЕ О МАТЕМАТИЧЕСКИХ МОДЕЛЯХ

Математической моделью какого-либо объекта называется любое формализованное (записанное с помощью математических, т. е. условных однозначно трактуемых символов) описание, отражающее состояние или поведение объекта с требуемой степенью точности.

Основными характеристиками модели являются тип рабочего сигнала, способ представления, характер зависимостей, уравнений и т. д. Классификация моделей по этим признакам приведена на рис. 1.3. Говоря о различных способах представления моделей, следует отметить аналитические модели как наиболее простые, алгоритмические модели как наиболее точные и универсальные, табличные модели как наиболее быстрые в смысле затрат машинного времени на их обработку, но требующие больших объемов памяти.

Классификацию моделей по уровню проработки лучше всего пояснить примером.

Пример 1.3. Пусть модель в виде *исходного уравнения* имеет вид $F(x) = 0$. Тогда модель в виде *теоретического алгоритма* может быть записана как алгоритм решения исходного уравнения методом Ньютона $F'(x^{(h)})\Delta x^{(h)} = -F(x^{(h)})$ или методом простых итераций $x^{(h+1)} = x^{(h)} - \lambda F(x^{(h)})$.

Модель в виде *машинного алгоритма* в отличие от теоретического алгоритма учитывает все реальные особенности численной реализации теоретического алгоритма. К ним относятся в данном случае, во-первых, ограниченная точность вычислений, в связи с чем в машинном алгоритме должен быть предусмотрен критерий остановки вычислений $\| \Delta x^{(h)} \| = \| x^{(h+1)} - x^{(h)} \| < \epsilon$, во-вторых, возможность расхождения вычислений, в связи с чем в машинном алгоритме нужно предусмотреть специальные меры (остановку вычислений, переход на другой метод вычислений и т. д.) при несходимости за заданное число итераций. Обычно машинный алгоритм представляется в виде блок-схемы.

Программная реализация модели может быть представлена в виде *модульной структуры* программы, *структурной модели* программы (с детальной структурой каждого модуля) и, наконец, в виде *текста* программы на каком-либо языке программирования. По существу, эти три уровня представления модели

соответствуют трем основным этапам разработки программы, составляющим метод *структурного программирования* (сверху вниз).

Наконец, последним уровнем проработки модели, необходимым для ее использования в автоматизированном проектировании, является ее *описание на языке проектирования* (входном языке). Фактически это описание служит обращением к модели, записанной на уровне текста программы.

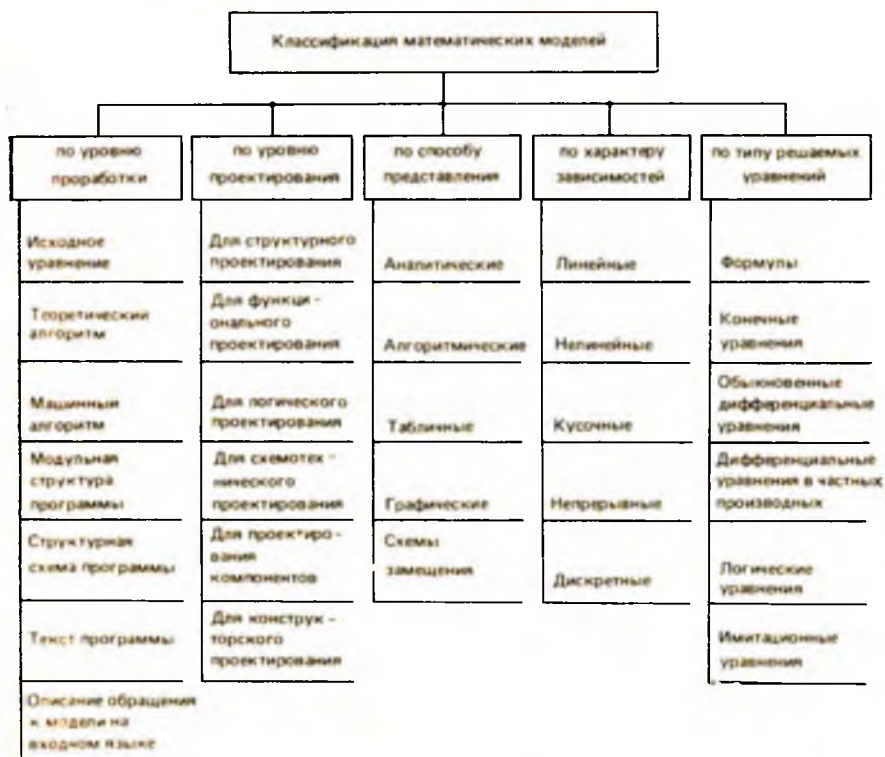


Рис. 1.3. Классификация математических моделей

Рассмотрим модели объектов на уровне исходных уравнений, используемых на разных функциональных уровнях проектирования.

Распределенные модели (модели с распределенными параметрами) для проектирования компонентов (элементов) интегральных микросхем представляют собой уравнения в частных производных, описывающие процессы в пленочных или полупроводниковых структурах. Иногда, если распределенным характером процессов можно пренебречь, как, например, в МДП-транзисторах, вместо распределенных используются сосредоточенные модели, в которых внутренними параметрами обязательно должны быть физико-топологические параметры компонента (элемента), характеризующие его размеры, топологию, свойства материалов и т. д.

Модели компонентов (элементов) для схемотехнического проектирования в общем случае представляются в виде обыкновенных дифференциальных уравнений, описывающих компонент (элемент) его схемой замещения с сосредоточенными параметрами.

Модели компонентов для функционального проектирования — это функции или алгоритмы, позволяющие по заданной форме входного сигнала $x_{вх}(t)$ или $x_{вх}(\omega)$ найти форму выходного сигнала $x_{вых}(t)$ или $x_{вых}(\omega)$.

Модели компонентов для структурного проектирования обычно считаются идеальными в смысле выполнения своих функций, так как на структурном уровне чаще всего обычно изучаются лишь характеристики структуры, в том числе точностные, обусловленные тем или иным способом соединения компонентов и их составом, а не ошибками в их работе.

Пример 1.4. Пусть компонентом является плечовая RC -структура (рис. 1.4). Трансформация модели структуры при проектировании РЭА разного уровня сложности, содержащих различное число RC -структур, отражена в табл. к рис. 1.4.

Проектирование отдельных RC -структур выполняется на основе максимальной точной распределенной модели структуры, описываемой уравнениями в частных производных.

Проектирование схем, содержащих несколько десятков RC -структур, потребует при использовании компонентной распределенной модели слишком больших затрат машинного времени на численное решение уравнений в частных производных, поэтому целесообразно заменить каждую RC -структуру сосредоточенной моделью — RC -цепью, в которой для повышения точности можно ввести нелинейную зависимость $C(u_c)$. Такие модели описываются обыкновенными дифференциальными уравнениями (ОДУ), численное решение которых существенно быстрее.

Если проектируется РЭА типа ФУ или ФУС, содержащих сотни RC -структур, то их расчет даже на основе численного решения ОДУ будет слишком длителен, поэтому целесообразно разбить их на фрагменты (группы), содержащие, например, десятки структур, а для каждого фрагмента найти приближенную переходную характеристику $h(t)$, считая отдельные структуры сосредоточенными и линейными RC -звеньями.

Такие фрагменты можно рассматривать как самостоятельные элементы, предназначенные для функционального уровня проектирования. В отличие от предыдущих двух уровней, относящихся к физическому моделированию, функциональный уровень относится к информационному моделированию. Его характерная особенность — пренебрежение взаимной реакцией компонентов при их соединении и однонаправленный характер передачи информации внутри компонента. Ошибки из-за этих допущений были бы существенны при последовательном соединении двух RC -структур, при соединении же двух фрагментов устройства, каждый из которых содержит десятки RC -структур, этими ошибками можно пренебречь.

Наконец, при проектировании РЭА, содержащей тысячи и более RC -структур, необходимо их объединение в еще более крупные фрагменты и дальнейшая идеализация, как показано в двух нижних строках табл. к рис. 1.4, где

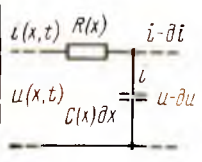
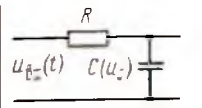
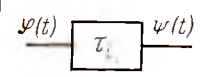
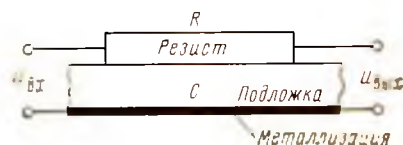
Уровень проектирования	Схема замещения	Математическая модель	Примечание
Компонентный		$\frac{\partial u}{\partial x} = -R(x) i(x, t)$ $\frac{\partial i}{\partial x} = -C(x) \frac{\partial u(x, t)}{\partial t}$	Точная распределенная модель
Схемотехнический		$\frac{du_C}{dt} = \frac{u_{вх}(t) - u_C(t)}{RC(u_C(t))}$	Сосредоточенная модель, распределенность не учитывается
Функциональный		$u_{вых}(t) = \int_0^t u_{вх}(\tau) h(t - \tau) d\tau$	Нелинейность не учитывается, $R_{вх} = 0$, $R_{вых} = \infty$
Выходной функциональный		$u_{вых}(t) = \int_0^t u_{вх}(t) dt$	Неидеальность интегрирования не учитывается
Структурный (информационная модель)		$\psi(t) = \varphi(t + \tau_i)$ $\varphi(t)$ — входной поток, $\psi(t)$ — выходной поток, τ_i — время срабатывания	Учитывается только одна информационная характеристика $\tau_i(t)$

Рис. 1.4. Математические модели пленочных RC-структур на разных функциональных уровнях проектирования



учитываются лишь идеализированные функциональные (идеальное интегрирование) и временные (время срабатывания) свойства. Иногда, при исследовании точностных характеристик системы на структурном уровне, применяют идеализированные модели не самих элементов, а вносимых этими элементами ошибок, например модель в виде постоянной задержки или ошибки воспроизведения входного сигнала. Модели, учитывающие лишь отдельные свойства элементов, например только вносимую задержку, называются *частными*.

1.6. ПОНЯТИЕ ОБ ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ

В основу решения на ЭВМ различных задач проектирования можно положить два принципиально различных подхода. Первый подход состоит в теоретическом исследовании надежности, точности и других характеристик проектируемого объекта, выводе расчетных формул, уравнений или алгоритмов и реализации их на ЭВМ. Второй подход заключается в построении модели проектируемого устройства, реализации ее на ЭВМ и расчете надежности, точности и т. д. на основе результатов моделирования. Если временная последовательность событий в модели и в реальном устройстве одинаковая, то такое моделирование называется *имитационным*.

При развертывании во времени событий и процессов, происходящих в реальном устройстве и его модели, различают четыре типа времени.

1. *Реальное время* — в нем протекают реальные процессы в реальном устройстве.

2. *Системное (или модельное) время* — абстрактное время, в котором протекают процессы или события при их моделировании на ЭВМ. В программах это время представляется отдельной переменной, которую пользователь или разработчик программы может менять по своему усмотрению. Применение системного времени позволяет изменять масштаб реального времени или даже останавливать его для изучения событий путем более тщательного расчета в отдельные моменты времени. Системное время — это реальное время, учитываемое в программе в некотором масштабе

(секунды, микросекунды, наносекунды и т. д.). На распечатках или на экране дисплея это время отображается цифрами, обозначающими реальное время, но расположение этих цифр может быть произвольным — редким, частым, с неравномерным расстоянием друг от друга — в зависимости от шага печати, заданного в программе вывода данных.

3. *Машинное время* — реальное время моделирования на ЭВМ. Обычно в нем выделяют время трансляции программы (перевода с языка про-

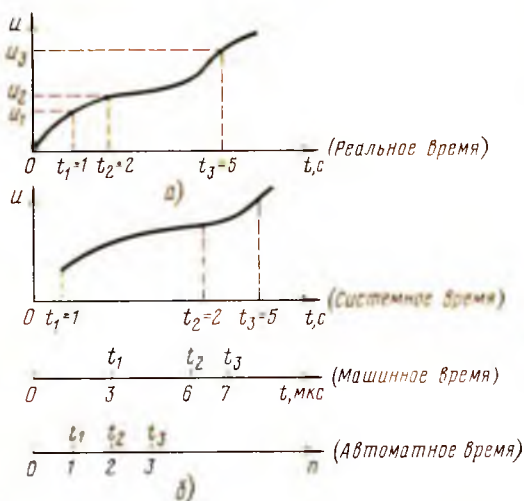


Рис. 1.5. Представление отдельных моментов реального процесса проектирования в разных типах времени

граммирования в коды команд ЭВМ) и процессорное время (время решения задачи после трансляции). Машинное время регистрируется специальным устройством ЭВМ — таймером.

4. *Автоматное время* — абстрактное время, обычно выраженное целыми числами 0, 1, 2, ... и регистрирующее лишь порядок следования событий независимо от их длительности.

Пример 1.5. Пусть процесс $u(t)$ (рис. 1.5,а), происходящий в некоторой схеме, достигает уровня u_1 при $t_1=1$ с, уровня u_2 при $t_2=2$ с и уровня u_3 при $t_3=5$ с.

Отметки t_1, t_2, t_3 на осях разных времен показаны на рис. 1.5. Как видим, в системном времени интервал t_1-t_2 внешне выглядит длиннее интервала t_2-t_3 , тогда как на самом деле в реальном времени он короче. Это значит, что процесс $u(t)$ на интервале t_1-t_2 в системном времени рассчитывался более подробно (с более мелким шагом изменения переменной, обозначающей время), чем на интервале t_2-t_3 . В машинном времени интервал t_1-t_2 длиннее, чем t_2-t_3 . Это значит, что расчет процесса $u(t)$ на интервале t_1-t_2 занял больше времени (3 мкс), чем на интервале t_2-t_3 (1 мкс). При этом сами численные значения t_1, t_2, t_3 в отличие от системного времени никак не связаны с численными значениями t_1, t_2, t_3 в реальном времени. В автоматном времени все события (пересечения процессом $u(t)$ соответствующих уровней) протекают равномерно.

1.7. ПОНЯТИЕ ТЕХНОЛОГИЧНОСТИ ПРОЦЕССА ПРОЕКТИРОВАНИЯ

Под широко распространенным и важным понятием технологичности любого промышленного изделия обычно понимают удобство и простоту его изготовления. В связи с быстрым моральным старением изделий, в том числе радиоэлектронных, предъявляются высокие требования к скорости, дешевизне и удобству не только процесса их производства, но и процесса проектирования. Поэтому в настоящее время можно говорить не только о технологичности производства РЭУ, но и о технологичности их проектирования.

Это понятие особенно важно в связи с широким внедрением САПР, затраты на создание которых весьма велики. Например, может оказаться, что затраты на проектирование какого-либо устройства на апробированной элементной базе с помощью готовой САПР будут значительно меньше затрат на проектирование модификации того же устройства, улучшенной благодаря использованию каких-либо элементов из другой элементной базы, так как для этого нужно провести дорогостоящую модификацию самой САПР. В данном случае новая модификация нетехнологична в смысле проектирования. В более общем случае необходимо оценивать экономическую эффективность новой РЭА, т. е. степень улучшения ее характеристик и требуемое для этого увеличение затрат на ее проектирование. Стоимость такого улучшения может оказаться слишком высокой из-за роста затрат на создание ново-

то проектного обеспечения (новой САПР). Для повышения технологичности процесса проектирования нужно широко использовать стандартные проектные решения и процедуры.

ГЛАВА 2

ОСНОВНЫЕ ПОНЯТИЯ САПР

2.1. ОБЩИЕ СВЕДЕНИЯ О САПР

2.1.1. Принципы построения САПР. Современная САПР представляет собой сложную *программно-информационно-аппаратурную человеко-машинную систему*, построенную по *иерархическому принципу*, так что каждый уровень иерархии отражает определенный уровень проектирования — структурный, функциональный и т. д. (см. табл. 1.1).

Разработка САПР занимает десятки и сотни человеколет труда высококвалифицированных специалистов, поэтому САПР должна в максимальной степени удовлетворять требованию моральной долговечности. В связи с этим САПР строится по *агрегатному принципу*, т. е. как открытая и развивающаяся система с максимальным использованием унифицированных модулей (отдельных программных подсистем, технических средств). Агрегатный принцип построения позволяет создать на основе базового варианта САПР любую необходимую пользователю конфигурацию путем объединения соответствующих программных и технических модулей.

Требование унификации программных модулей означает их полную *информационную согласованность*, исключающую необходимость ручной переработки общих для нескольких модулей массивов информации. Унификация технических модулей означает полную *согласованность стандартов* на передаваемые ими сигналы (согласованность уровней постоянного напряжения, амплитуд выходных сигналов и уровней срабатывания, временных соотношений и т. д.). Для обеспечения согласованности программных и технических модулей часто используется *межмодульный интерфейс* — специальные программы для перекомпоновки информации (драйверы) и технические устройства для изменения параметров сигналов (адаптеры).

2.1.2. Классификация пользователей САПР. Специалистов, работающих с САПР, называют пользователями САПР. Их можно разделить на три категории.

1. Пользователи-разработчики САПР. Это наиболее квалифицированная категория пользователей, в совершенстве владеющая программированием, математическими методами, а также хорошо знающая предмет проектирования.

2. Пользователи-сопроводители САПР. Эта категория пользователей выполняет профилактические работы по поддержанию

САПР в рабочем состоянии (пополнение библиотек САПР необходимой информацией, модификация САПР в рамках запланированных возможностей, учет и диагностика ошибок, выявляемых в процессе эксплуатации САПР). Эта же категория консультирует инженеров по вопросам методики использования САПР для проектирования. Пользователи-сопроводители должны в совершенстве знать алгоритмы и программы, содержащиеся в эксплуатируемой САПР.

3. Пользователи-разработчики РЭА. Это наиболее широкая категория пользователей, использующая САПР для решения конкретных прикладных задач и называемая часто конечными пользователями. По некоторым оценкам эта категория составляет около 80% от общей численности всех пользователей. Разработчики РЭА должны хорошо знать, какие задачи проектирования можно решать с помощью САПР, какие модели можно использовать для проектирования, приближенно оценивать затраты машинного времени для решения своих задач, уметь составить исходную информацию на входном языке данной САПР и самостоятельно разбираться в выходной информации, выдаваемой САПР. Разумеется, необходимо всемерно стремиться к повышению квалификации этой группы пользователей, так как от этого зависит общая культура проектирования с помощью САПР, а следовательно, и его эффективность.

2.1.3. САПР как человеко-машинная система. Выше отмечалось, что САПР должна строиться как человеко-машинная система, в которой часть проектных операций выполняется человеком, а часть — ЭВМ. Проектирование без участия человека, т. е. *автоматическое проектирование*, в полном объеме — от формулировки технического задания до получения проектной документации — как творческий процесс невозможно и всегда будет иметь характер *автоматизированного проектирования* (с участием человека), хотя отдельные его этапы (расчет, анализ) уже сейчас полностью выполняются на ЭВМ.

Участие человека в процессе проектирования формально носит характер диалога между ЭВМ и проектировщиком РЭУ. Этот диалог эффективен при выполнении следующих трех условий.

1. Человек работает эффективнее ЭВМ. Это возможно в двух случаях:

при решении трудно формализуемых или неформализуемых задач, например задач синтеза новых технических решений или выбора наилучшего из многих вариантов по нескольким критериям одновременно;

при решении формализуемых задач, которые ЭВМ на основе строгих методов решает медленнее, чем человек на основе своих эвристических возможностей.

Эвристика — это не обоснованное теоретически строго правило, практика позволяющее быстро получать удовлетворительный результат, с высокой вероятностью близкий к оптимальному. Особенно часто этот случай имеет место при решении ком-

бинаторных задач поиска оптимума с большим числом вариантов исходных данных, когда их направленный эвристический перебор, выполняемый человеком, оказывается значительно быстрее методов перебора на ЭВМ по формальным алгоритмам.

Другим примером могут служить задачи оптимизации с непрерывными переменными, время решения которых часто заметно уменьшается, если человек имеет возможность оперативно влиять на процесс оптимизации — менять алгоритмы оптимизации, их параметры и критерии оптимальности в зависимости от складывающейся по ходу оптимизации обстановки.

2. Время ожидания $t_{ож}$ ответа на запрос в режиме диалога (время реплики) должно быть небольшим, чтобы у пользователя не появилось чувство психологического дискомфорта (неуверенность в правильности запроса, в безошибочной работе программы и т. д.). Время ожидания индивидуально для каждого пользователя, однако принято считать, что оно должно быть соизмеримым с временем формирования запроса и во всяком случае не превосходить несколько десятков секунд.

Это условие ограничивает круг задач, решаемых в режиме диалога, задачами невысокой размерности (десятки переменных), время решения которых на ЭВМ между двумя моментами, требующими участия человека, невелико. В противном случае необходимо принимать специальные меры к снижению времени $t_{ож}$ — упрощать математические модели и методы, разбивать объект с большим числом переменных на части с меньшим числом переменных (пространственная декомпозиция), вводить промежуточные уровни упрощенного математического описания объекта (функциональная декомпозиция), использовать обобщенные переменные и т. д.

3. Объем изменяемой в процессе диалога информации не должен быть слишком большим, иначе диалог теряет оперативность, к тому же возрастает вероятность ввода ошибочной информации.

В целом, диалог наиболее эффективен при так называемом эскизном проектировании, когда используются приближенные оценочные соотношения в виде формул, математическая модель РЭА довольно проста и время расчета одного варианта мало. Целью диалогового проектирования должно быть получение исходного приближенно оптимального варианта РЭА, уточнение которого далее выполняется на основе более точных математических моделей без применения диалога. В терминах оптимизации задача диалога — выход в окрестность оптимума, где дальнейшее проектирование (поиск точного оптимума) может быть выполнено автоматически.

2.1.4. Режимы взаимодействия пользователя и САПР. Различают следующие режимы взаимодействия пользователя и САПР: пакетной обработки, прямого доступа и с использованием автоматизированного рабочего места (АРМ).

1. Режим пакетной обработки задач пользователя — это режим, когда из отдельных задач — колод перфокарт — оператор

ЭВМ комплектует в виде общей колоды пакет задач, вводимых в ЭВМ и решаемых затем поочередно без вмешательства пользователя. Каждая задача решается в течение заданного времени, положительный результат решения не гарантируется, возможна преждевременная остановка процесса решения отдельных задач из-за ошибок в их программах или данных.

Этот режим характерен для многих вычислительных центров общего назначения с малым объемом услуг, предоставляемых индивидуальным пользователям. Обычно пользователь, работающий с САПР в «режиме пакета», имеет возможность выйти со своей задачей на ЭВМ не более двух-трех раз в сутки. Характерным для этого режима является также участие оператора ЭВМ как связующего звена между пользователем и САПР, упорядочивающего организацию работы на вычислительном центре.

2. Режим прямого доступа (пультовый, терминальный) — это режим, когда пользователь непосредственно работает на ЭВМ, без посредничества оператора. В режиме прямого доступа способ ввода исходной информации зависит от ее количества — при большом количестве информации ее целесообразно вводить с перфокарт, а при малом — с помощью экранного пульта (дисплея), который может находиться далеко от ЭВМ.

Однако при любом способе ввода ошибки исходной информации корректируются непосредственно во время вычислительного сеанса с помощью дисплея, что существенно ускоряет процесс отладки программ и решения задач. В зависимости от числа используемых дисплеев различают однопультный и многопультный режим прямого доступа. В последнем случае САПР представляет собой систему коллективного пользования.

Следует заметить, что корректирование ошибок в исходных данных с помощью дисплея до начала проектирования иногда называют режимом диалога, хотя это не совсем верно. Подлинный режим диалога предполагает возможность вмешиваться в процесс проектирования после ввода исходных данных, например изменять ход процесса проектирования в зависимости от текущих результатов. При этом используется специально разработанный для данного процесса проектирования язык диалога.

3. Режим с использованием АРМ. Применение одного лишь дисплея для ввода информации в ЭВМ или даже работы в режиме диалога оказывается в ряде случаев недостаточным для эффективной эксплуатации больших ЭВМ в процессе автоматизированного проектирования. Помимо дисплея пользователь САПР нуждается в широком наборе технических и программных средств для оперативного и долговременного документирования и корректировки текстовой и графической информации, а также для простейших расчетов, не требующих больших ЭВМ. Совокупность этих средств получила название АРМ. Обычно АРМ расположено непосредственно возле пользователя. Режим эксплуатации САПР с использованием АРМ — наиболее эффективен как с позиций

пользователя, так и в отношении загрузки больших ЭВМ, к которым обычно подключается несколько АРМ.

2.1.5. Классификация САПР. В зависимости от сложности решаемых задач существующие САПР можно разделить на четыре типа.

1. Уникальные САПР, каждая из которых создается специально для решения какой-либо одной крупной научно-технической проблемы. Так, в свое время была создана уникальная САПР для проектирования многопроцессорного вычислительного комплекса «Эльбрус» с высокой производительностью.

2. Отраслевые САПР, решающие типовые задачи отрасли. Примером может служить САПР «ПРАМ», предназначенная для проектирования РЭА.

3. САПР отдельных предприятий, ориентированные на решение типовых задач предприятия, например задач проектирования СВЧ-устройств или устройств вычислительной техники.

4. Мини-САПР для решения отдельных задач проектирования, например электрического расчета схем или трассировки печатных плат.

Первые три типа относятся к многофункциональным САПР коллективного пользования и реализуются по двухступенчатой иерархической схеме — на верхнем уровне находится мощная ЭВМ, на нижнем — периферийные малые ЭВМ и АРМ. Мини-САПР реализуются обычно на малых и средних ЭВМ, могут эксплуатироваться автономно и относятся к наиболее распространенному сейчас типу САПР.

2.1.6. Виды обеспечения САПР. Современная САПР — сложный комплекс математических, программных, технических и других средств. Принято выделять в составе САПР следующие основные части:

математическое обеспечение;

лингвистическое обеспечение (языковые средства);

программное обеспечение;

информационное обеспечение;

техническое обеспечение;

организационное обеспечение;

методическое обеспечение.

Ниже подробно анализируется каждая из этих частей.

2.2. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

2.2.1. Классификация основных алгоритмов. Математическое обеспечение включает теорию, методы, способы и алгоритмы для организации вычислений в САПР. В составе математического обеспечения можно выделить следующие типы алгоритмов (рис. 2.1).

1. *Алгоритмы решения общих задач вычислительной математики:* решение конечных уравнений $F(x)=0$, линейных уравнений $Ax=B$, обыкновенных дифференциальных уравнений $dx/dt=f(x(t), t)$ и т. д. Обычно эти алгоритмы в связи с их частым

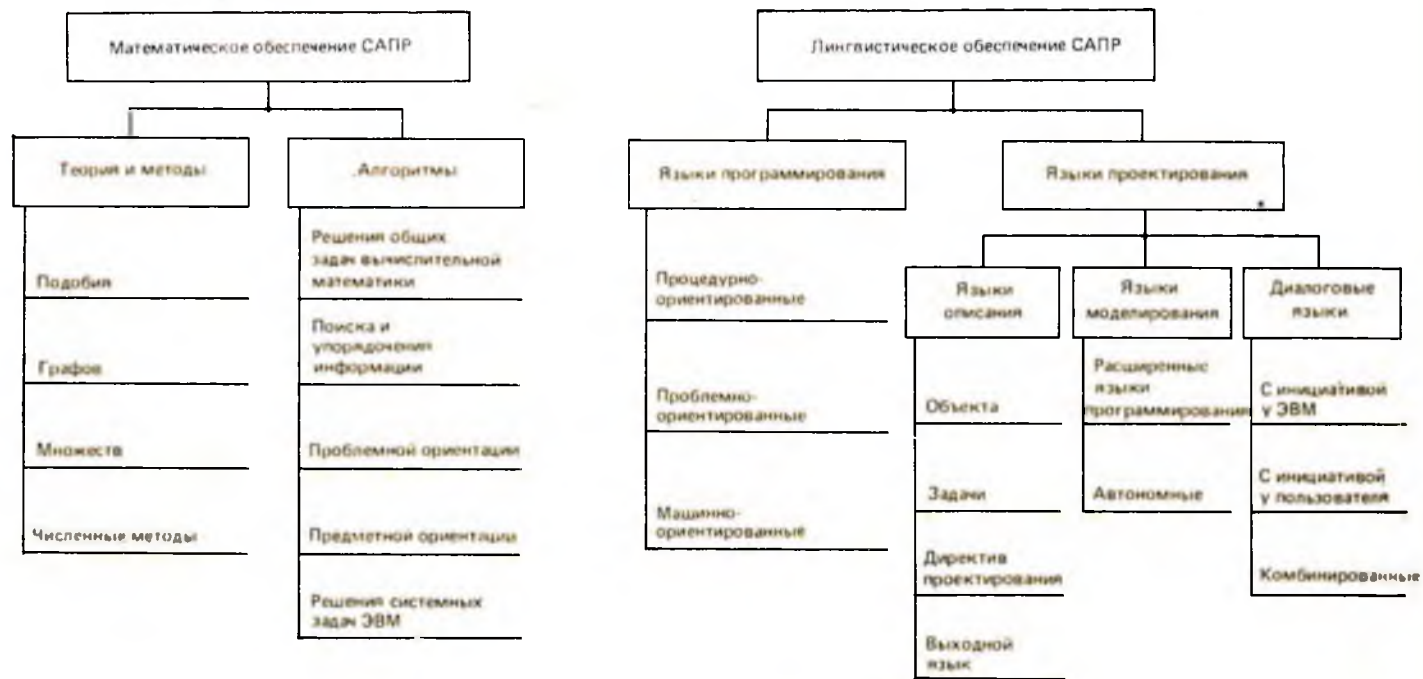


Рис. 2.1. Классификация математического и лингвистического обеспечения САПР

употреблением в задачах различного типа оформляются в виде стандартных программ, однако в САПР эти программы используются редко из-за разных форм представления исходной информации, определяемых, в частности, алгоритмами нижеследующей группы.

2. Алгоритмы поиска и упорядочения информации:

алгоритмы поиска нужного элемента или группы элементов в массиве данных; поиск можно вести различными способами: перебором, делением на части, по дереву признаков и т. д.;

алгоритмы редактирования информации (перекодировки, переконфигурации, корректировки); одним из примеров редактирования является *упаковка информации*, например запись матрицы в виде совокупности одномерных массивов, хранящих информацию только о ненулевых элементах матрицы;

алгоритмы сортировки информации, т. е. разделения ее на группы в соответствии с какими-либо признаками.

Данная группа алгоритмов хотя и играет роль сервисных, но во многом определяет вычислительные затраты на решение задач в САПР. Эти алгоритмы имеет смысл применять только при создании программ многоразового пользования.

3. Алгоритмы проблемной ориентации: решение задач, связанных общей научной тематикой, например алгоритмы решения статистических задач (построение гистограмм, аппроксимация гистограмм теоретическим законом, вычисление статистических характеристик и т. д.).

4. Алгоритмы предметной ориентации: формирование и расчет математических моделей объектов определенного типа, например электронных переключательных схем, схем СВЧ и т. д. Эта группа алгоритмов — основная в САПР. Они обычно включают, как составные части, алгоритмы трех предшествующих групп.

5. Алгоритмы решения системных задач ЭВМ: организация работы операционных систем, диспетчеризации, управление программами, данными и т. д. [7, 8].

2.2.2. Основные требования к алгоритмам. К алгоритмам, используемым в САПР, предъявляются следующие требования.

Высокая алгоритмическая надежность, т. е. гарантированное получение правильного результата при любых численных значениях исходных данных, значениях параметров в заданных диапазонах варьирования и для любых видов функциональных зависимостей в задачах данного класса. В связи с этим требованием многие численные методы, хорошо работающие в частных случаях, но не обеспечивающие сходимости вычислений в достаточно общем случае, в САПР не применяются.

Возможность формализации, что ограничивает применение в САПР таких численных методов, принципиальным моментом которых являются искусство и опыт вычислителя. Исключением здесь могут быть лишь диалоговые алгоритмы проектирования, в которых неформальный опыт проектирования играет принципиальную роль.

Малые вычислительные затраты при реализации, причем в соотношении «память — время» в связи с быстрым ростом объема оперативной памяти современных ЭВМ основным становится сейчас требование уменьшения времени счета.

Разумное соотношение «точность — время» с учетом того, что незначительное ухудшение точности моделирования может существенно уменьшить время моделирования.

Алгоритмическая совместимость, т. е. согласованность и достаточность входных и выходных данных разных алгоритмов, совместно работающих в составе одной и той же программы. Например, если для расчета статистики и динамики используются алгоритмы нулевого порядка, не требующие расчета производных, то для оптимизации тоже нужно применять алгоритм нулевого (а не первого) порядка, иначе потребуются большие дополнительные затраты времени и памяти на расчет производных, а коэффициент их использования будет мал (только в блоке оптимизации). Иначе говоря, для комплекса алгоритмов, используемых в одной программе, необходима *алгоритмическая гармония*.

2.3. ЛИНГВИСТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

2.3.1. Языки программирования. Языки, используемые в САПР (рис. 2.1), можно разбить на две группы — для программирования и для проектирования.

Языки программирования предназначены для написания текстов программ.

Процедурно-ориентированные языки предназначены для решения широкого класса задач. Из этих языков наиболее распространен фортран, используется ПЛ1, перспективны Паскаль, АДА.

Проблемно-ориентированные языки ориентированы на решение задач определенного типа. Среди них можно выделить группу языков, предназначенных для организации диалога и текстовой обработки информации.

Машинно-ориентированные языки, обычно называемые ассемблерами, позволяют создавать программы, наиболее эффективные в смысле вычислительных затрат (памяти и времени счета).

Выбор языка программирования зависит от задач, поставленных перед началом разработки программы, например:

- разработать программу возможно быстрее;
- разработать программу, возможно более эффективную по вычислительным затратам;
- разработать максимально *мобильную* программу, т. е. пригодную для эксплуатации на ЭВМ различных типов.

Поскольку эти требования противоречивы, необходимо выделить главное из них и в соответствии с ним выбрать язык программирования. Если главным является скорость разработки программы, то программу нужно писать на языке высокого уровня — процедурно- или проблемно-ориентированном. Если же основное — эффективность программы, то нужно использовать языки низкого

уровня, так как, например, программа на языке Ассемблера работает в несколько раз быстрее, чем на языках высокого уровня. Однако, учитывая высокую трудоемкость программирования на Ассемблере, обычно при разработке больших программ идут на компромисс: наиболее ответственные, часто используемые подпрограммы пишут на Ассемблере, а менее ответственные (сервисные, однократно используемые) — на языке высокого уровня, хотя это и порождает проблемы мобильности программы, т. е. ее использования на ЭВМ разных типов и серий.

Если главное требование — мобильность, то программные вставки на машинно-ориентированных языках недопустимы.

2.3.2. Языки проектирования. Перечисленные выше стандартные языки программирования мало пригодны для описания объектов и задач проектирования, так как в них отсутствуют такие понятия как «тип объекта», «связи объекта», «параметры объекта», не предусмотрены описания типовых процедур проектирования, обычно учитывающих специфику объекта проектирования. Необходимость специальных конструкций, выражений языка для описания перечисленных понятий приводит к созданию предметно-ориентированных языков, называемых *входными языками*, или *языками проектирования*.

Языки проектирования можно разделить на три группы — описательные, моделирующие и диалоговые. Их называют также соответственно языками структурного, процедурного и директивного типов.

1. *Язык описания* обычно состоит из трех частей — описания объекта, описания задачи и описания директив проектирования.

В *описание объекта* входят описания отдельных элементов, каждое из которых обычно имеет следующую структуру:

тип элемента, тип модели элемента, параметры модели элемента, топологические связи элемента.

Например, описание транзистора включает тип транзистора, тип его модели в программе (один и тот же транзистор можно моделировать разными схемами замещения), параметры указанной модели и топологические связи транзистора, т. е. номера узлов его подключения, записанные в определенной последовательности, например номер узла коллектора, затем базы и эмиттера.

Способ описания топологических связей элементов зависит от типа элемента — направленного (ориентированного) или ненаправленного. Так, резистор — ненаправленный элемент, поэтому порядок указания узлов его включения в схему безразличен. Логический элемент, наоборот, является направленным, поэтому порядок перечисления его выводов имеет значение.

Пример 2.1. Для схемы рис. 2.2,а возможны следующие варианты описания связей элементов. Если рассматривать элементы 1, 2, 3, 4 как ненаправленные, то связи элементов можно описать таблицами 1 и 2 на с. 31.

Первая таблица указывает, какие узлы связаны с каждым элементом, а вторая — какие элементы связаны с каждым узлом. По первой таблице можно

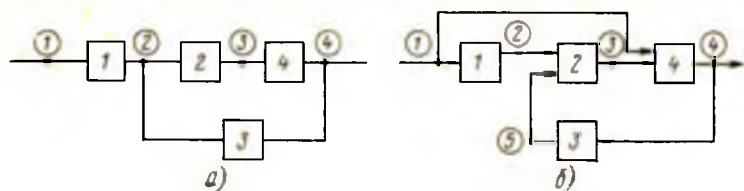


Рис. 2.2. Функциональная схема устройства с ненаправленными (а) и с направленными (б) элементами

Таблица 1
элемент—узел

Номер элемента	Номера узлов
1	1, 2
2	2, 3
3	2, 4
4	3, 4

Таблица 2
узел—элемент

Номер узла	Номера элементов
1	1
2	1, 2, 3
3	2, 4
4	3, 4

легко построить вторую, и наоборот. Обычно используют для описания связей ненаправленных элементов первую таблицу.

Если рассматривать элементы 1, 2, 3, 4 как направленные (рис. 2.2,б), причем сигнал в элементах 1, 2, 4 проходит только слева направо, а в элементе 3 обратной связи — справа налево, то связи элементов можно описать одной из трех таких таблиц:

Таблица элемент—узел

Номер элемента	Номера входных узлов	Номера выходных узлов
1	1	2
2	2, 5	3
3	4	5
4	1, 3	4

Таблица узел—элемент

Номер узла	Номера элементов, подключенных к узлу входами	Номера элементов, подключенных к узлу выходами
1	1, 4	—
2	2	1
3	1	2
4	3	4
5	2	3

Таблица элемент—элемент

Номер элемента	Элементы с выходами, подключенными к входу	Элементы с выходами, подключенными к выходу	Элементы с входами, подключенными к входу	Элементы с входами, подключенными к выходу
1	—	—	4	2
2	1, 3	—	—	4
3	4	—	—	2
4	2	—	1	3

Третья колонка в последней таблице всегда пустая, так как выходы направленных элементов замыкать нельзя. Обычно в языке проектирования связи описывают с помощью наиболее просто составляемой вручную таблицы «элемент — узел» или «элемент — входные узлы — выходные узлы», которая затем с помощью сервисных программ перерабатывается в таблицу, наиболее удобную для использования в программе.

Язык описания задачи включает обычно следующую информацию:

- описание рассчитываемых выходных параметров (тип параметра, уровни отсчета, условия расчета и т. д.);

- описание условий анализа параметров (тип варьируемых внутренних параметров, шаг и диапазон варьирования и т. д.);

- описание условий оптимизации параметров (сведения о варьируемых параметрах, выходных оптимизируемых параметрах, ограничениях, критериях оптимизации);

- описание алгоритмов расчета, анализа и оптимизации (типы алгоритмов и параметры, определяющие их скорость, точность и надежность);

- описание задания на вывод результатов проектирования (что выводить и в каком виде — таблица, график, чертеж; параметры выходного документа — шаг печати, масштаб, диапазон и т. д.).

Язык описания директив на проектирование в простейшем случае состоит из перечисления режимов, в которых должна последовательно работать САПР. Хотя каждая директива, например «Статика», «Частотный анализ» и др., вызывает определенные действия, однако непосредственно в самих директивах никаких операторов действия нет, поэтому язык сохраняет описательный характер.

2. Языки моделирования (процедурные языки) описывают не только структуру и параметры объекта проектирования, но и алгоритм, процедуру его функционирования, например процесс передачи и преобразования сигнала от блока к блоку.

Часто язык моделирования строится на базе какого-либо языка программирования, к которому добавляются несколько новых конструкций, необходимых для моделирования в заданной предметной области. Такой моделирующий язык называется *расширением языка программирования*. Если язык моделирования основан на самостоятельных конструкциях, то он называется *автономным*.

Как правило, языки моделирования применяются только на верхних уровнях проектирования — структурном и функциональном, когда алгоритмы моделирования еще достаточно просты и доступны для описания на языке моделирования пользователю-разработчику РЭУ, не имеющему высокой квалификации в алгоритмизации и программировании.

Пример 2. 2. Пусть на рис. 2.2,б, где все элементы однонаправленные, элемент 1 описывается функцией $x_{вых} = f_1(x_{вх})$, элемент 2 — функцией $x_{вых} = f_2(x_{вх})$ и т. д., причем элемент 3 — это элемент обратной связи, тогда про-

цесс функционирования системы на языке моделирования будет описываться языком вида

$$\begin{aligned}x_{\text{вх}1} &= \text{Пр } 1(x_{\text{вх}1}), \\x_{\text{вых}2} &= \text{Пр } 2(x_{\text{вх}1}, x_{\text{вых}3}), \\x_{\text{вых}3} &= \text{Пр } 3(x_{\text{вых}4}), \\x_{\text{вых}4} &= \text{Пр } 4(x_{\text{вх}1}, x_{\text{вых}2}).\end{aligned}$$

Здесь $\text{Пр } 1 \dots \text{Пр } 4$ — записанные на языке моделирования программные выражения для функций $f_1, \dots, f_4$, а $x_{\text{вх}1}, x_{\text{вых}4}$ — сигналы на входе и выходе i -го элемента.

После трансляции полученная система уравнений относительно $x_{\text{вых}1}, \dots, x_{\text{вых}4}$ ($x_{\text{вх}1}$ известно) может быть решена в программе каким-либо численным методом.

Таким образом, пользователь сам составляет математическую модель схемы, но не на общепринятом математическом, а на специальном языке моделирования. Очевидно, вероятность ошибки составления модели схемы на языке моделирования несколько выше, чем на языке описания.

3. *Языки диалога* предназначены для организации взаимодействия пользователя и САПР в процессе проектирования. Различают три типа диалоговых языков: *с инициативой у пользователя, с инициативой у ЭВМ и комбинированный*.

В первом случае вопросы и указания задает пользователь, а ЭВМ отвечает на вопросы и реализует указания, во втором случае — наоборот, а в третьем пользователь и ЭВМ в процессе проектирования могут меняться ролями.

Основными элементами языка диалога являются четыре — *подсказка* ЭВМ пользователю, *директива* пользователя ЭВМ, *меню*, представляющее для ЭВМ или, чаще, для пользователя возможность выбора, и *анкета* (бланк).

Пример 2.3. Рассмотрим типичный фрагмент диалогового процесса схемотехнического проектирования, ведущегося с помощью алфавитно-цифрового дисплея.

ЭВМ: Введите информацию о схеме

Пользователь: (с помощью клавиатуры дисплея вводит описание схемы)

ЭВМ: Неправильно введена информация о транзисторе T5

Пользователь: (корректирует информацию о транзисторе T5)

ЭВМ: Информация о схеме введена. Выбрать режим

Статика

Динамика

Частотный анализ

Оптимизация

Печать

Пользователь: (указывает световым пером или меткой-курсором требуемый режим проектирования, например «Статика»)

ЭВМ: Укажите параметры режима: алгоритм ..., точность = ..., максимальное число итераций = ..., начальные условия = ...

Пользователь: (указывает параметры, после этого дает директиву на экране дисплея: «Данные введены, выполнить режим»)

ЭВМ: Итерации не сходятся. Что делать?

Пользователь: Перейти на алгоритм...

Здесь первые две реплики [вопросы (указание) — реакция] соответствуют диалогу с инициативой у ЭВМ, которая работает в режиме подсказки. Перечисление возможных режимов работы программы в третьей реплике — это элемент типа «меню», четвертая реплика — элемент типа «анкета», последняя реплика — это директива, когда инициатива переходит к пользователю.

Использование диалоговых языков существенно упрощает освоение методики автоматизированного проектирования, ускоряет процесс ввода и корректировки данных, позволяет оперативно изменять ход проектирования в зависимости от его текущих результатов.

2.4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

2.4.1. Общие сведения. В информационное обеспечение САПР (рис. 2.3) входят, во-первых, сведения о типовых элементах РЭА и их параметрах, типовых материалах, типовых фрагментах схем

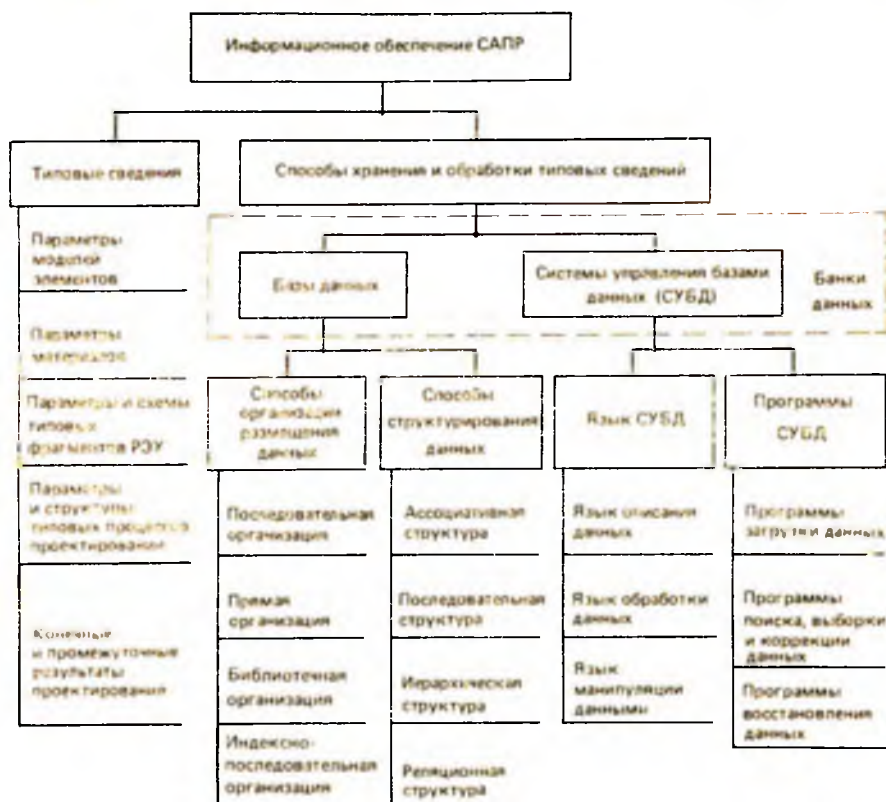


Рис. 2.3. Классификация информационного обеспечения САПР

и, во-вторых, способы, алгоритмы и программы, предназначенные для упорядоченной записи, хранения, перемещения и извлечения этих данных. Со второй частью информационного обеспечения связаны такие понятия, как база данных (БД), система управления базой данных (СУБД) и банк данных [4—8].

База данных — совокупность массивов данных, организованных таким образом, чтобы обеспечить быстрый и удобный поиск любых данных по запросу или их перемещение и корректировку.

Система управления базой данных — совокупность языковых средств и программ, предназначенных для поиска нужных данных, их перемещения и модификации независимо от прикладных программ разных пользователей.

В совокупности БД и СУБД образуют **банк данных**.

Основными требованиями к информационному обеспечению являются полнота, гибкая организация структур данных и способов управления ими, позволяющая пополнять, корректировать и изымать данные без их существенной перестройки, а также обеспечение быстрого и простого поиска нужных данных.

2.4.2. Способы организации размещения данных. Простейшая единица данных (одно или группа) называется **записью**.

Различают **логическую** и **физическую** форму записи. Обозначение записи какими-либо символами и цифрами называется логической записью, а ее содержание, размещенное на каком-либо физическом носителе, например магнитной ленте или диске, — физической записью или блоком. Например, R11 — логическая запись информации о резисторе с номиналом 11. Если символы R, 1, 1 занимают по четыре байта каждый, то эти 12 байтов, содержащие логическую запись R11 и размещенные на магнитной ленте или диске, образуют физическую запись. Таким образом, логическая запись — форма записи, воспринимаемая пользователем, а физическая запись — материализованная форма логической записи.

Физическая и логическая формы записей могут совпадать, а могут и различаться. Например, пользователь может воспринимать цепочку логических записей A, B, C, D как записи, расположенные подряд, последовательно друг за другом, тогда как физические записи A, B, C, D могут располагаться на физическом носителе — диске в любой последовательности, при этом в физической записи A должно находиться указание начального адреса записи B, в B — указание на начало C и т. д. Далее, если не оговорено, под записью понимается логическая форма записи.

Обычно запись содержит не только числовые значения данных, но и вспомогательную информацию, позволяющую отличать записи друг от друга. Эта информация называется **ключом записи**. Различают два типа ключей — **ключи порядка**, с помощью которых определяется место записи в базе данных среди других записей, и **смысловые ключи** (ключи типа данных), определяющие смысловое содержание записи и позволяющие выделить данную запись среди других. Например, ключом порядка может служить

номер транзистора, а смысловым ключом — символ Т301, определяющий вид элемента (транзистор) и его тип (301). Другой пример записи — 12R6, где 12 — номер резистора, который можно использовать как ключ порядка, R — смысловой ключ, 6 — номинал резистора (числовое данное).

Работа с записями может выполняться в трех основных режимах: *последовательная* обработка, *произвольная* обработка (в произвольном порядке) и *корректировка*.

Рассмотрим теперь основные четыре способа организации *размещения записей*: последовательный, прямой, библиотечный и индексно-последовательный (рис. 2.4).

Последовательная организация записей состоит в последовательном размещении логических записей друг за другом.

При этом физические записи могут либо следовать подряд друг за другом (рис. 2.4,а), либо располагаться в различных местах носителя информации (рис. 2.4,б). В последнем случае записи организуются в виде связного списка, когда в конце каждой записи имеется указание адреса начала следующей записи (A2, A3 и т. д. на рис. 2.4,б).

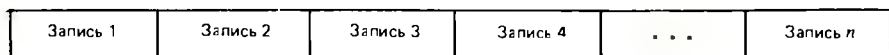
Обычно последовательная организация размещения записей предполагает их последовательную обработку — запрос поступает на группу записей, считываемых друг за другом. Однако при произвольной обработке файла последовательная организация становится неэффективной из-за больших затрат времени на поиск отдельных записей, так как для этого нужно последовательно просмотреть все предыдущие записи.

Прямая организация записей (рис. 2.4,в) состоит в том, что ключ логической записи однозначно определяет физический адрес ее начала.

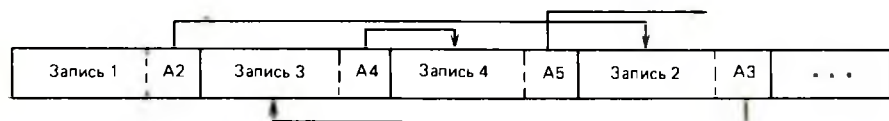
В простейшем случае ключ и адрес совпадают, например ключ 10R резистора одновременно есть номер ячейки (10), где хранится его значение. Прямая организация файла данных предполагает произвольный режим его обработки и отличается высокой скоростью поиска отдельных логических записей, который состоит лишь в поиске нужного ключа и определении по нему адреса начала записи.

Библиотечная организация данных (рис. 2.4,г) может рассматриваться как вариант прямой организации. Записи разделяются на группы, называемые разделами. Каждый раздел получает свое имя. Список имен и адреса начала первых и конца последних записей каждого раздела помещаются в начале файла и образуют его оглавление (каталог). Таким образом, поиск нужного раздела состоит в просмотре каталога. Внутри раздела записи могут организовываться любым способом.

Индексно-последовательная организация данных (рис. 2.4,д) предназначена для ускорения поиска записей по сравнению с последовательной организацией. Логические записи размещаются по мере возрастания значений их ключей порядка. Файл ключей порядка записей разбивается на части так, что начальные ад-



а)

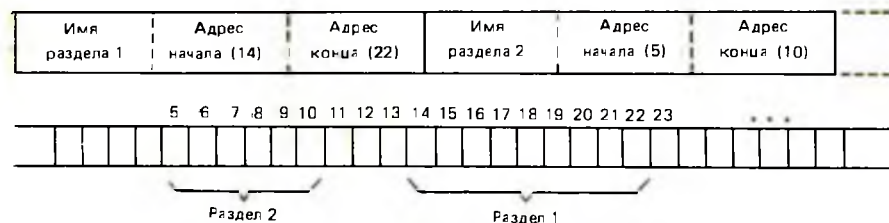


б)

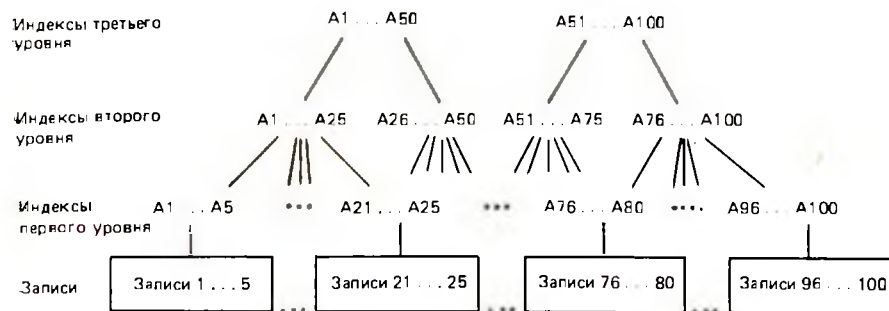


в)

Оглавление (каталог) библиотеки



г)



д)

Рис. 2.4. Типы организации размещения информации:
а, б — последовательная; в — прямая; г — библиотечная; д — индексно-последовательная

реса этих частей, называемые индексами, образуют иерархическую структуру типа дерева, верхние уровни которого соответствуют более крупным частям, а нижние — более мелким. Пусть ключ порядка записи — это ее начальный адрес. Сравнивая значение ключа порядка записи с индексами верхнего уровня, затем следующего и т. д., определяем минимальный диапазон адресов, где может находиться искомая запись, далее в этом диапазоне производится последовательный поиск.

Пример 2.4. Пусть имеется 100 начальных адресов от A1 до A100, разбитых на группы по пять адресов каждая. Тогда иерархические уровни начальных адресов будут иметь вид, изображенный на рис. 2.4.б. Если логическая запись имеет ключ порядка A77, то, сравнивая его с индексами верхнего уровня, попадаем в группу A51 ... A100, затем на нижеследующем уровне — в группу A76 ... A100 и на нижнем уровне — в группу A76 ... A80, после чего для поиска нужной записи A77 потребуется просмотреть только эту группу.

Таким образом, файл с индексно-последовательной организацией содержит две области — область индексов и основную область, содержащую записи. Вначале просматривается область индексов с целью выделения нужной подобласти основной области, а затем — выделенная подобласть. При этом внутри подобластей (блоков записей A1 ... A5, ..., A96 ... A100) могут быть зарезервированы свободные места для размещения новых записей без перемещения старых. Если же этих свободных мест не хватает, то они помещаются в специальной третьей области индексно-последовательного файла — области переполнения и связываются с записями из основной области обычно с помощью связанных списков.

2.4.3. Способы структурирования данных. В отличие от способов организации размещения отдельных частей информации (рис. 2.4) под способом *структурирования данных* понимается способ формирования из этих частей каких-либо конфигураций по тому или иному признаку. Например, если база данных содержит независимые смысловые части A1, A2, A3, A4, A5, A6, размещенные прямым или индексно-последовательным способом, то пользователю могут потребоваться такие конфигурации данных, как цепочка A1, A3, A5 без A2, A4, A6 или цепочка A2, A4, A6 без A1, A3, A5.

Принцип, на котором основан способ структурирования данных, часто называют *концептуальной моделью* базы данных, моделью или просто структурой данных [6].

Основными типами моделей данных являются ассоциативная, последовательная, иерархическая и реляционная.

Ассоциативная модель позволяет выделять часть информации или отдельные данные, имеющие какое-либо указанное общее свойство, например все резисторы из всех элементов схемы — транзисторов, конденсаторов, резисторов.

Последовательная модель устанавливает связь между предыдущей и последующей частью информации и может представляться либо в виде строго последовательной цепочки записей, физическая и логическая структуры которых совпадают (рис. 2.4.а),

либо в виде связанного списка, когда они не совпадают, но в конце каждой физической записи есть указание на место начала следующей физической записи (рис. 2.4,б) при этом логическая структура записей считается последовательной.

Иерархические модели данных имеют две разновидности — **древовидную** и **сетевую** (рис. 2.5,а,б).

Иерархические модели данных характерны тем, что отдельные части информации располагаются на разных иерархических уровнях, связи частей однонаправлены сверху вниз и организованы так, что можно образовать любую цепочку связанных сверху вниз частей, например цепочки 1, 2, 5 или 1, 3, 7. При этом, если каждая часть может быть связана лишь с одной предыдущей частью, т. е. имеет один вход, то структура называется **древовидной**, а если имеет несколько входов, то **сетевой**. Очевидно, сетевая структура мощнее древовидной по разнообразию связей.

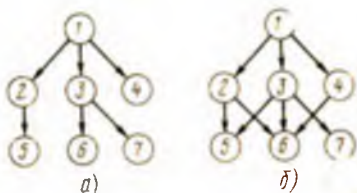


Рис. 2.5. Иерархические модели данных:
а — древовидная; б — сетевая

Реляционные модели данных основаны на выделении смысловых групп информации с использованием реляционного исчисления (алгебры отношений между данными). Данные задаются в виде таблиц, реляционные операции над которыми (выделение элементов по принципу соответствия тому или иному отношению — эквивалентно, равно, больше и т. д.) порождают новые таблицы с другими отношениями. Простейшим примером реляционной модели данных может служить всем знакомая киноафиша. Приведенные в ней отношения кинотеатр — фильм — день — время сеанса позволяют легко получить другие отношения, например фильм — кинотеатр, день — фильм. Реляционные модели данных наиболее сложны, но и наиболее универсальны.

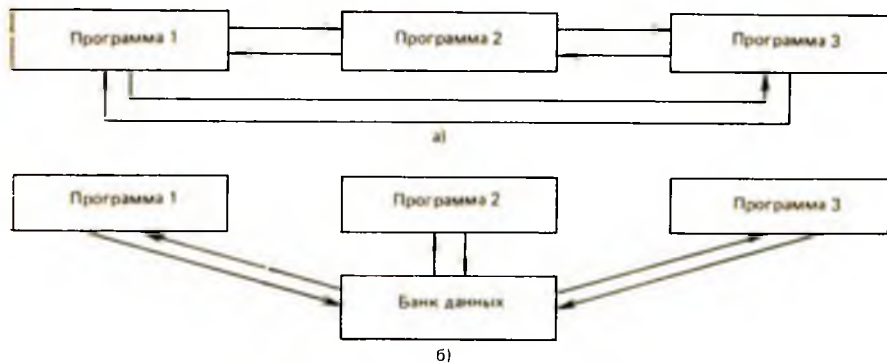


Рис. 2.6. Информационные связи между программами при использовании прикладных баз данных (а) и системных баз данных (б)

2.4.4. Системы управления базами данных. СУБД играет роль интерфейса между пользователями и базой данных и представляет собой автономную программную систему, обычно не входящую в операционную систему и состоящую из трех групп программ: 1) обработки и организации вводимых данных, 2) создания и корректировки файлов, 3) поиска данных в файлах.

Назначение СУБД — обеспечить независимость прикладных программ от конкретного способа организации и структуры данных и тем самым сделать базу данных инвариантной в отношении пользователей, что позволяет использовать одну и ту же СУБД и БД для обслуживания различных подсистем САПР (рис. 2.6). Свойство инвариантности позволяет, в частности, применять в САПР стандартные универсальные банки данных, однако это требует от пользователя решения специфических задач привязки своих прикладных программ к этим банкам данных, что снижает их эффективность.

СУБД имеет свои языковые средства для описания базы данных и обеспечения доступа прикладных программ САПР, а также самих пользователей к базе данных. Язык СУБД состоит из трех частей (см. рис. 2.3):

1) *языка описания данных*, не зависящего от прикладных программ и описывающего способ организации, размещения и связи данных;

2) *языка манипулирования данными*, позволяющего связать базу данных с прикладными программами или пользователем, т. е. сформировать нужную конфигурацию из отдельных частей;

3) *языка обработки данных* (ввода, вывода, стирания, корректировки данных и их восстановления, если они испорчены).

При этом каждый способ организации модели данных порождает свой способ доступа к ним; так, для иерархических моделей данных применяется иерархический индексно-последовательный или иерархический последовательный метод доступа.

2.4.5. Прикладные и системные базы данных САПР. Базы и библиотеки данных, используемые в САПР, могут организовываться двумя способами — как *прикладные* (внутри прикладных программ САПР как часть этих программ) и как *системные* (общие для всех подсистем САПР).

В первом случае обычно хранение данных организуется по библиотечному принципу, программирование способа хранения данных и доступа к ним выполняется на языке программирования. Во втором случае данные организуются по прямому, последовательному или индексно-последовательному способу в зависимости от частоты их использования и образуют системную базу данных, допускающую одновременную согласованную модификацию данных, используемых в разных подсистемах САПР.

Например, запись в системную базу данных сведений о каком-либо элементе РЭУ требует одновременной записи параметров его различных моделей — структурной, функциональной, схематической и т. д. — в разные части системной базы. Системные базы

данных могут быть организованы на основе языка управления заданиями операционной системы, автоматически формирующего последовательные, прямые, а в системе ОС ЕС и библиотечные структуры данных.

Информационные связи программ, использующих одни и те же данные, организованные в виде прикладных и системных баз, показаны на рис. 2.6, а, б.

Основные преимущества системных баз данных состоят в следующем. Во-первых, данные оказываются независимыми от прикладных программ. Это облегчает разработку прикладных программ, так как программист может не знать полную структуру данных, место их хранения и должен знать лишь способ обращения к базе данных. Кроме того, в этом случае упрощается модификация как прикладных программ, так и базы данных, поскольку вследствие их независимости объем переделок существенно уменьшается.

Во-вторых, использование разными подсистемами САПР общей базы данных исключает необходимость дублирования данных в разных подсистемах, а также создания специальных информационных стыковочных массивов для передачи данных из одной системы в другую в согласованном общем формате.

В-третьих, системная база данных позволяет легко организовать их защиту, например, от попыток одновременной модификации данных разными пользователями или от несанкционированного доступа к данным, а также организовать контроль правильности запросов пользователей к базе.

Недостатком системных баз данных является большое время получения данных по запросу, связанное с использованием операционных систем (ОС) и поэтому существенно превышающее это время в прикладных базах. Поэтому данные в САПР организуются обычно по двухступенному иерархическому методу — имеется долговременная системная база, обслуживающая оперативные прикладные базы данных внутри прикладных программ, которые могут наполняться по запросу пользователя нужными данными каждый раз перед решением очередной задачи или группы задач.

Использование языка манипулирования БД позволяет записывать в тексте программ запросы к системной БД, на основе которых СУБД отыскивает нужные данные и передает их прикладным программам.

2.5. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

В программное обеспечение входят тексты программ и документы для их эксплуатации (инструкции для пользователя, тестовые программы для диагностики ошибок и сбоев и т. д.).

Основными требованиями к программному обеспечению являются: 1) гибкая организация (модульное построение, взаимозаменяемость модулей и др.), допускающая возможность построения различных конфигураций программных систем и их расширения;

2) хорошее сервисное обеспечение (возможность диагностики ошибок, работы в режиме диалога, с разделением времени); 3) высокое качество текстов программ (структурированность, быстрое действие, эффективное использование памяти).

2.5.1. Системное программное обеспечение. Системное программное обеспечение включает *общие и специализированные* операционные системы. К общим относятся системы типа ОС ЕС, ДОС ЕС, ОС РВ СМ и др., к специализированным — системы, создаваемые специально для управления уникальными программными комплексами, например подсистемами САПР.

Операционной системой называется совокупность программ, управляющих работой ЭВМ, обеспечивающих согласование работы ЭВМ и поступающих в нее программ пользователей для максимальной загрузки всех устройств ЭВМ и осуществляющих связь ЭВМ с пользователем с целью предоставления ему максимальных удобств (услуг) при решении задач (информации о ходе решения задачи и причинах остановок в решении, возможности использования различных устройств ввода-вывода, облегчения процесса отладки программ и т. д.).

Структура типовой ОС дана на рис. 2.7. Основными ее частями являются управляющие и системные обрабатывающие программы. В мощные ОС типа ОС ЕС входят также программы телеобработки данных, обеспечивающие передачу данных в ЭВМ по телефонным, телеграфным и другим каналам связи, и программы машинной графики, обеспечивающие работу ЭВМ с алфавитно-цифровыми и графическими дисплеями.

Следует отметить, что по принятой в ОС терминологии основной единицей выполняемой в ЭВМ работы является задание, которое состоит в выполнении одной или нескольких связанных программ. Для ОС общего назначения примером задания является ввод, трансляция, редактирование и выполнение некоторой программы, а для специализированной ОС — ввод описания схемы и задания на ее проектирование, расчет статистики, динамики и статистический анализ. Задания никак друг с другом не связаны, представляют собой независимые единицы работы и могут выполняться параллельно в режиме многопрограммной работы.

Каждое задание состоит из одной или нескольких последовательно выполняемых частей, называемых *пунктами* (шагами) *задания*. Примеры пунктов задания — пункт ввода данных, пункт трансляции, пункт выполнения программы. Задания выполняются по пунктам, при этом для выполнения каждого пункта создается задача (простая или с подзадачами), называемая задачей пункта задания (ее не следует смешивать с задачей пользователя). Таким образом, с каждым пунктом задания связано выполнение некоторой программы, реализующей задачу пункта задания. Имя этой программы указывается пользователем при описании пункта задания на языке управления заданиями. Пункты задания выполняются при однопрограммной работе ЭВМ последовательно друг за другом.

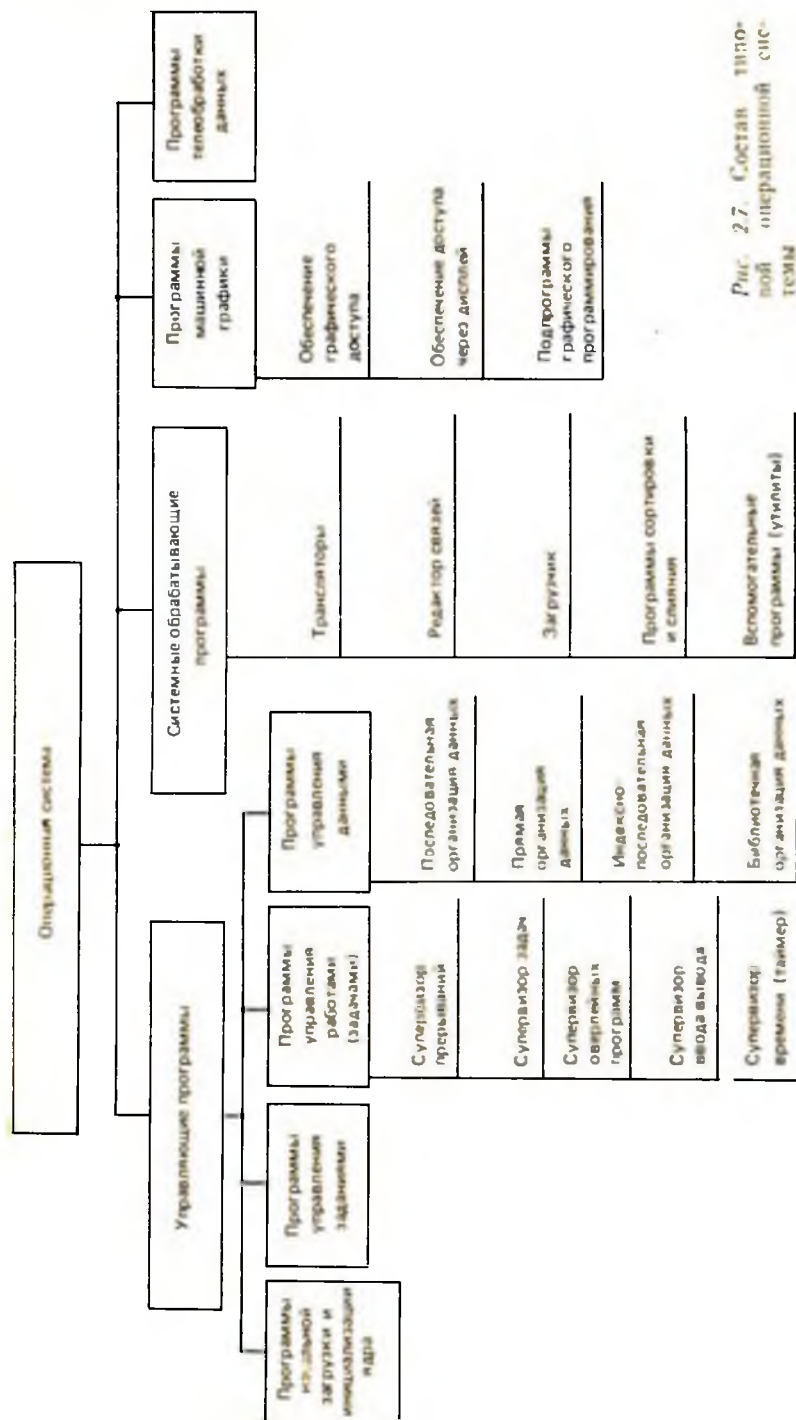


Рис. 27 Состав типовой операционной системы

Язык управления заданиями состоит из управляющих операторов. В ОС ЕС основными операторами являются: JOB (оператор задания), EXEC (оператор пункта задания, т. е. задачи) и DD (оператор описания данных).

Первым оператором каждого задания является оператор JOB, в котором указывается имя выполняемой программы, фамилия программиста, номер задания и его приоритет, объем требуемой памяти и т. д.

Каждый пункт задания начинается с оператора EXEC, в котором указывается тип выполняемой задачи и необходимая для ее выполнения информация.

Операторы DD служат для описания наборов данных, используемых в задаче, места их расположения, структуры, характеристик, действий, которые нужно с ними выполнить и др. Каждый оператор DD описывает один набор, причем вслед за оператором DD может следовать не только описание набора, но и сам набор, что создает большие удобства для пользователя. Последовательность заданий, описанных на языке управления заданиями, образует входной поток заданий, обрабатываемый и выполняемый с помощью ОС.

В дальнейшем, чтобы не смешивать понятия «задача пользователя» и «задача пункта задания», последние будем называть работами.

Рассмотрим функции управляющих программ ОС (см. рис. 2.7). *Программа начальной загрузки* подготавливает оперативную память ЭВМ, выделяя необходимые разделы, вводит в нее ядро ОС — системные программы, постоянно находящиеся затем в оперативной памяти, и запускает (инициирует) эти программы, выполняя инициацию ядра ОС. При этом в память могут загружаться различные варианты и сочетания системных программ и тем самым генерироваться различные конфигурации ОС, соответствующие типу ЭВМ, составу периферийных устройств, режиму обработки заданий и т. д.

Управление заданиями выполняется *программой-планировщиком* (супервизором заданий). В его функции входит: 1) контроль правильности записи управляющих операторов на языке управления заданиями, наличия разделителей между заданиями, правильности записи имен отдельных модулей программ; 2) назначение устройств ввода-вывода разным программам, управление установкой носителей информации (нужных катушек магнитной ленты, пакетов дисков, рулонов бумаги и т. д.); 3) планирование прохождения заданий (последовательное, приоритетное).

Управление задачами (работами) выполняется *супервизором работ*, который контролирует прохождение программы (задания) с момента ввода до получения результатов. В функции супервизора работ, часто называемого просто супервизором, входит обработка всех видов прерываний, возникающих при выполнении задач.

Прерыванием называется сигнал, генерируемый в вычислительной системе при возникновении определенных ситуаций. В ОС ЕС возможны пять типов прерываний: прерывания от ввода-вывода, программные прерывания, прерывания при обращении к супервизору, внешние прерывания, прерывания от схем контроля ЭВМ. Обработка прерывания супервизором заключается в выявлении причины прерывания и выполнении действий, регламентируемых возникшим прерыванием (изменение работы отдельных устройств ЭВМ, сообщение оператору, остановка вычислений и т. д.).

Кроме обработки прерываний в функции супервизора входит управление выполнением задач, памятью, таймером, устройствами ввода-вывода, программами с оверлейной структурой (с использованием одних и тех же разделов памяти для хранения и выполнения разных модулей программы) и т. д. Отдельные управляющие программы, выполняющие перечисленные функции, также называются супервизорами (см. рис. 2.7).

Управление данными включает программы организации и перемещения данных между оперативной памятью и внешними носителями, ввод и вывод данных с разным типом организации, анализ ошибок при вводе-выводе, совмещение операций ввода-вывода и обработки данных, объединение записей в блоки и деблокирование записей и т. д.

Системные обрабатывающие программы работают на разных этапах обработки программы. Исходная программа, написанная на языке программирования, средствами языка управления заданиями разделяется на отдельные автономные части, называемые *исходными* (символьными) *модулями*. Программы-трансляторы переводят текст каждого исходного модуля с языка программирования на язык, близкий к машинному. Оттранслированный исходный модуль называется *объектным модулем*.

Объектные модули представляются независимо от языка программирования в некотором промежуточном, общем для всех трансляторов формате. В объектном модуле обращения к подпрограммам заменяются на так называемые адресные ссылки, неопределенные при трансляции, поэтому объектные модули еще не являются готовой к исполнению программой и требуют дополнительной обработки, называемой редактированием.

Программа *редактор связей* устанавливает связи по данным и соответствие по адресам между объектными модулями, которые после этого называются *загрузочными* или *абсолютными модулями*. Редактор связей формирует также из отдельных объектных и загрузочных модулей один общий, готовый к выполнению загрузочный модуль.

Программа сортировки и слияния (объединения) данных размещает их в заданном порядке, сортирует, объединяет данные, размещенные на разных носителях и др.

Программы-утилиты (вспомогательные программы) перемещают данные с одного носителя на другой, обеспечивают печать

данных, каталогов, оглавлений библиотек, разметку дисков и лент и т. д.

2.5.2. Специализированные операционные системы (СОС). В отличие от общих ОС они предназначены для управления выполнением конкретных прикладных задач с учетом их содержания. Обычно САПР состоит из отдельных программ, реализующих тот или иной вид проектирования (расчет, анализ, моделирование, трассировка, размещение, компоновка и т. д.) и работающих под управлением СОС.

Типовой состав СОС представлен на рис. 2.8,а. СОС работает, как правило, в так называемом режиме РСР — первичной управляющей программы, когда выполняется одно задание и его обслуживают все ресурсы, выделенные для данного задания общей операционной системой. Таким образом, СОС играет роль вторичной операционной системы по отношению к общей ОС, которая может

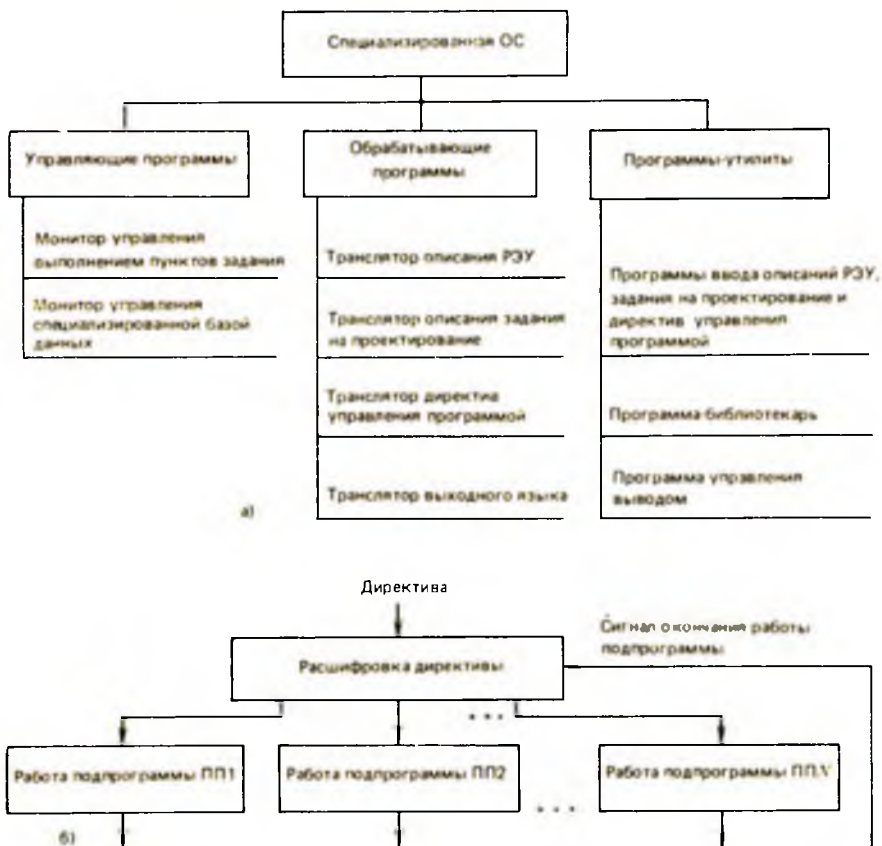


Рис. 2.8. Состав типовой специализированной ОС (а) и структура программы-монитора (б)

выделять ресурсы и планировать задания для разных СОС, причем возможна их параллельная работа.

Эта особенность СОС объясняет отсутствие в ней программ начальной загрузки СОС и программ управления заданиями, функции которых выполняются общей ОС, например ОС ЕС.

Пунктами задания при решении задач проектирования РЭА могут быть ввод описания РЭА, трансляция описания РЭУ с входного языка во внутренние форматы программы, расчет статического режима, расчет переходных процессов, печать результатов расчета и т. д. Цепочка из этих пунктов образует задание на проектирование РЭА.

Роль супервизора пунктов задания в СОС играет специальная управляющая программа, называемая обычно *монитором*. Монитор инициирует подпрограммы ПП1, ПП2, ..., ПП_N (рис. 2.8,б), реализующие отдельные пункты задания. В простейшем случае программа-монитор состоит из операторов обращения к этим подпрограммам (например, операторов CALL ... на языке Фортран).

Помимо монитора управления выполнением пунктов задания в СОС могут входить различные другие программы.

Конкретный состав программ СОС определяется назначением САПР и может быть различным, однако в качестве типовых программ можно назвать следующие.

Обслуживающие программы-утилиты:

программы ввода описаний и директив, входящих в язык проектирования; обычно из-за различия конструкций этих описаний каждое описание вводится отдельной программой;

программы управления выводом; вывод результатов может осуществляться в виде отдельных чисел, таблиц или графиков (в ОС ЕС для графического вывода можно использовать системные программы машинной графики);

программа-библиотекарь, используемая для организации библиотек моделей элементов и выполняющая роль СУБД для организации специализированной базы данных;

программы, печатающие диагностические сообщения о причинах сбоев в работе САПР.

Обрабатывающие программы:

трансляторы описаний и директив, входящих в язык проектирования; обычно эта трансляция состоит в сортировке и перекомпоновке описаний в отдельные массивы, структура и содержание которых определяется удобством разработки программы;

транслятор с языка моделирования, обычно представляющего расширение какого-либо процедурно-ориентированного языка.

2.5.3. Предметные программы САПР. К предметным относятся программы, выполняющие непосредственно моделирование состояний и процессов в объектах проектирования. Примерами предметных программ САПР могут служить программы составления математических моделей РЭУ, расчета переходных процессов, расчета частотных характеристик, логического моделирования и т. д. Разработка этих программ занимает основную часть времени при

создании САПР, а их функционирование — основную часть машинного времени работы САПР. Качество САПР в основном определяется качеством предметных программ, поэтому их разработка ведется с особой тщательностью.

Все остальное программное обеспечение — общие и специализированные ОС, стандартные программы — играют вспомогательную, обслуживающую роль.

2.5.4. Функционирование САПР в среде ОС. Рассмотрим принципы функционирования САПР в среде операционной системы общего назначения (рис. 2.9). Сначала составляется задание на системном языке управления заданиями, т. е. с использованием операторов JOB, EXEC, DD и других, принятых в данной ОС. Назначение этого задания — указать устройства ввода, тип транслятора, если САПР используется впервые, определить состав модулей и создать из них требуемую конфигурацию САПР.

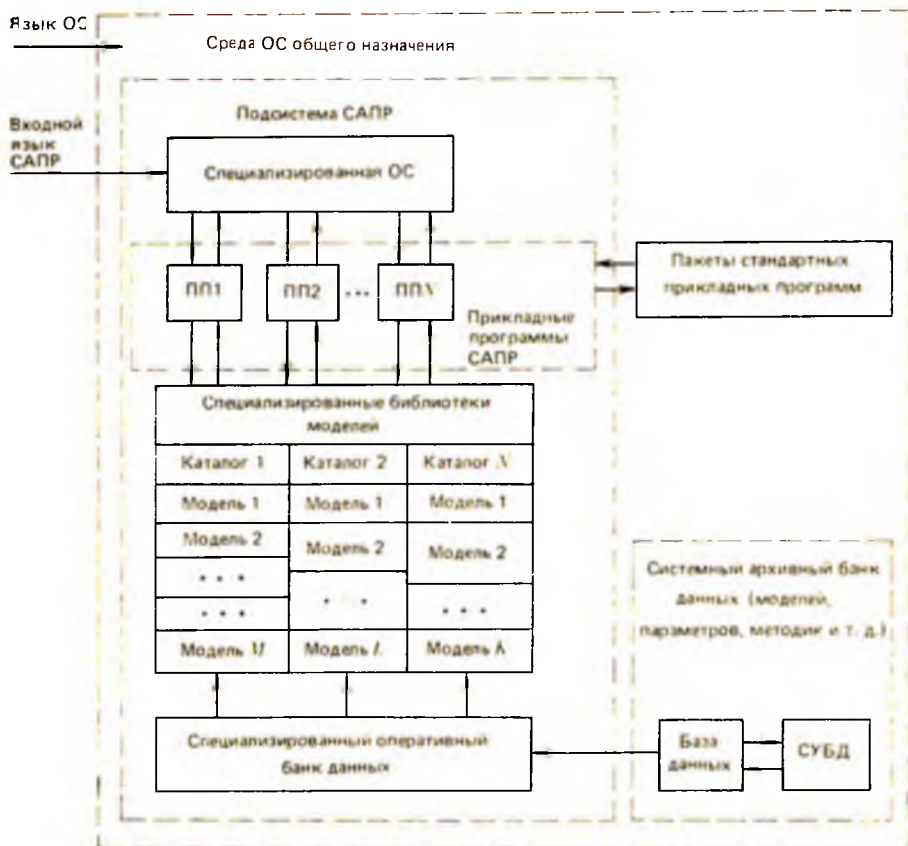


Рис. 2.9. Структура типовой программной подсистемы САПР

Например, при ограниченном объеме оперативной памяти, не вмещающем сразу все модули (подпрограммы) САПР, можно на языке управления заданиями скомпоновать оверлейную структуру САПР с использованием всех модулей (однако при этом частые обменные операции между оперативной и дисковой памятью могут резко снизить скорость расчетов) либо обычную структуру без перекрытия по памяти, но с использованием не всех модулей.

Одним из пунктов задания, записанного на языке общей ОС, может быть инициация подсистемы САПР, в результате которой управление передается специализированной ОС (см. рис. 2.9).

В свою очередь, специализированная ОС работает в соответствии с заданием, записанным на входном языке проектирования. Пунктами этого задания могут быть ввод описаний РЭУ, задания на проектирование, вызов тех или иных предметных программ ПП1, ПП2 и т. д. В процессе функционирования этих программ допустимы обращения к стандартным программам и к специализированным библиотекам моделей, параметры которых выбираются из специализированной базы данных, оперативно обслуживающей данную подсистему САПР и ее модули. Пополнение специализированной базы данных из системной базы должно предшествовать началу работы САПР и выполняется с помощью директив на языке СУБД ОС.

Остановимся подробнее на языке управления работой программ, называемом также входным языком СОС.

Структура входного языка СОС зависит от сложности и принципа построения САПР. Если подсистемы САПР работают в жестко заданной последовательности, то нагрузка на СОС весьма мала и ее функции может с успехом выполнять общая ОС, иницирующая первую подсистему САПР, вслед за которой автоматически включаются в работу в определенной последовательности остальные подсистемы САПР. Такой режим работы САПР называется автоматическим [2]. Он предполагает фиксированное заранее сопряжение информационных стыковочных массивов между отдельными подсистемами САПР. Иногда устанавливается несколько возможных последовательностей работы подсистем САПР.

Если необходимо обеспечить произвольную последовательность работы подсистем САПР, то управляющий входной язык СОС усложняется. В этом случае в нем должны быть средства описания заданной последовательности работы подсистем САПР, а в самой СОС — средства сопряжения данных между различными подсистемами (стыковочный интерфейс данных). Простейшим вариантом входного языка такой СОС является *управляющий вектор*, каждая позиция которого соответствует определенной подсистеме САПР или режиму ее работы. Записывая в позиции вектора условные знаки или числа, можно задать любую последовательность работы подсистем САПР или отдельных программ в подсистеме.

В состав технического обеспечения САПР входят ЭВМ и периферийное вспомогательное оборудование, обеспечивающее удобство взаимодействия проектировщика и САПР.

2.6.1. Электронные вычислительные машины. В настоящее время не выпускаются специализированные ЭВМ, предназначенные только для решения задач проектирования, поэтому в САПР используются ЭВМ общего назначения серий ЕС, СМ, «Электроника». К ЭВМ для САПР предъявляются требования высокого быстродействия (100 тыс. опер./с и более) и большой емкости памяти (сотни килобайт), поэтому маломощные вычислительные устройства (простейшие микропроцессоры, микрокалькуляторы и др.) в САПР не используются.

Обычно решение задач автоматизированного проектирования на физическом уровне требует высокоточных численных расчетов, связанных с выполнением условий сходимости при итерационных процессах. Поэтому для решения таких задач в САПР необходимо применять ЭВМ с достаточно длинными машинными словами (не менее 4 байт). Это затрудняет использование, например, для схемотехнических расчетов ЭВМ с длиной слова 2 байта.

Искусственное увеличение длины слова программными средствами сокращает объем памяти и замедляет выполнение расчетов. В то же время решение, например, дискретных задач проектирования или задач, не связанных с численным решением уравнений, характерных для информационного уровня, вполне возможно на малых и средних ЭВМ с коротким машинным словом.

2.6.2. Периферийные средства. К ним относят устройства графического ввода, чертежные автоматы, координатографы, дисплеи и АРМ.

Устройства графического ввода выполняют преобразование графической информации (ГИ) в цифровую форму, состоящее из операций считывания и кодирования. Считывание ГИ — это определение координат элементов чертежей (точек, линий, фрагментов), т. е. их привязка к выбранной системе координат, а кодирование — это переработка найденных координат в цифровую форму.

Чертежные автоматы служат для вычерчивания чертежей по заданной цифровой информации. Эти устройства позволяют перевести результаты проектирования, полученные в виде цифровых кодов, в графическую форму и облегчают изготовление технической документации.

Координатографы являются разновидностью чертежных автоматов и отличаются высокой точностью вычерчивания. Поэтому их используют при разработке фотооригиналов подложек микросхем, печатных плат и т. д. Изображение в отличие от чертежных автоматов наносится путем вырезания, гравировки и т. д. по заданной ЭВМ программе.

Дисплеи — это устройства отображения информации на экране электронно-лучевой трубки. В зависимости от содержания ин-

формации различают алфавитно-цифровые (символьные) и графические дисплеи.

Автоматизированные рабочие места представляют собой комплексы для организации диалога между ЭВМ и проектировщиком с широким использованием графической информации. В состав АРМ обычно входят мини-ЭВМ, алфавитно-цифровой или графический дисплей, устройство графического ввода и чертежный автомат. АРМ может работать либо в автономном режиме для решения небольших задач, либо в комплексе с большой ЭВМ для организации диалога и редактирования информации.

В настоящее время промышленностью выпускаются АРМ, ориентированные на выполнение определенных видов проектирования: АРМ-Р — для радиоинженеров, АРМ-М — для инженеров машиностроителей и т. д. Наибольшее применение АРМ находят при решении конструкторских задач, требующих большого объема графических работ.

2.7. ОРГАНИЗАЦИОННОЕ И МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Внедрение и эксплуатация САПР представляет не только сложную научно-техническую, но и организационную проблему, требующую создания специальных подразделений, ответственных за текущее состояние САПР, ее модернизацию, правильное функционирование и т. д.

Совокупность правил, инструкций и документов, регламентирующих состав групп обслуживания САПР, их обязанности и взаимоотношения, образует *организационное обеспечение* САПР. В основном этот вид обеспечения связан с экономическими и правовыми проблемами, однако от него часто зависит судьба новых САПР. Не вдаваясь в детали, отметим только, что такая динамичная часть САПР, как библиотека моделей, должна содержаться в необходимом состоянии группой информационной поддержки САПР непосредственно в организации, эксплуатирующей САПР.

Методическое обеспечение САПР — это описания программ, баз данных, языков проектирования и различные инструкции по использованию всех видов обеспечения САПР.

ГЛАВА 3

СТРУКТУРНОЕ МОДЕЛИРОВАНИЕ

3.1. ПОСТАНОВКА ЗАДАЧИ

Одним из первых этапов проектирования РЭА является проектирование ее структуры. На этом этапе объект проектирования представляется в виде структурной схемы, т. е. совокупности взаимно-связанных достаточно крупных завершенных блоков. При

этом иногда физические каналы связи между блоками рассматриваются как отдельные блоки с соответствующими параметрами, а графические связи между блоками в структурной схеме считаются идеальными каналами связи.

Исходными данными при проектировании структурной схемы обычно являются:

тип, параметры и характеристики отдельных блоков структуры, например надежность, точность, пропускная способность, время срабатывания и т. п.;

требования к выходным параметрам и характеристикам всей структурной схемы — производительности, стоимости, надежности, точности и т. д.

Задача проектирования структурной схемы заключается в том, чтобы, варьируя типами, параметрами и связями блоков, найти такую структуру, которая обладала бы заданными выходными параметрами и характеристиками. Ограничениями при проектировании структуры могут быть номенклатура блоков, их предельно допустимые характеристики, число связей и др.

Проектирование структуры начинается с синтеза ее исходного варианта. Этот синтез выполняется квалифицированными разработчиками на основе имеющегося у них опыта разработки аналогичных структур.

Например, при разработке структурной схемы информационно-вычислительной системы исходят из анализа решаемых системой задач, пропускной способности каналов связи, скорости работы внешних устройств, оперативной памяти, арифметического устройства и т. д. При этом предложенная структура должна удовлетворять некоторым поставленным требованиям по производительности, стоимости, надежности и т. п.

Синтезированная структурная схема должна быть проанализирована. Малое время, отводимое на проектирование систем, большая сложность и стоимость не позволяют производить анализ на опытных макетах системы, поэтому анализ структурных схем производится либо аналитическими методами, либо методами имитационного моделирования на ЭВМ.

3.2. ОСНОВНЫЕ СПОСОБЫ СТРУКТУРНОГО МОДЕЛИРОВАНИЯ

3.2.1. Аналитическое моделирование. При анализе системы работа каждого блока может быть описана в виде аналитического выражения, соответствующего решаемой задаче. Например, при определении вероятности безотказной работы системы каждый из блоков характеризуется собственной вероятностью безотказной работы. Вообще аналитическое решение задачи может быть получено при условии, что функции, описывающие поведение отдельных блоков, хорошо известны, не имеют точек разрывов, а отклонения параметров блоков невелики.

Методика аналитического способа моделирования состоит в составлении математического описания системы с точки зрения рассматриваемых характеристик, вычисления этих характеристик при значениях параметров, соответствующих выбранной структуре, и оценке полученных значений. Так, вероятность отказа за некоторое время t системы, состоящей из параллельно и последовательно соединенных блоков, для каждого из которых заданы плотности распределения вероятности отказов $f(t)$, может быть вычислена с учетом следующих соотношений.

При заданной плотности распределения $f(t)$ вероятность отказа блока за время t определяется выражением $Q(t) = \int_0^t f(t) dt$.

Для параллельно соединенных блоков (дублирующих друг друга), имеющих вероятности отказов $Q_1(t)$ и $Q_2(t)$, суммарная вероятность отказа определяется как $Q_{\text{пар}}(t) = Q_1(t) Q_2(t)$; для последовательно соединенных (не дублирующих) блоков — как $Q_{\text{посл}}(t) = Q_1(t) + Q_2(t) - Q_1(t) Q_2(t)$.

С учетом этих соотношений для системы может быть найдено общее выражение для вычисления отказа за время t , по которому и определяется вероятность отказа за конкретно заданное время.

Использование аналитических моделей позволяет затратить значительно меньше труда на получение решения, чем при использовании имитационного моделирования. Однако случайный характер явлений, происходящих в элементах системы (шумы, отказы, время срабатывания), и внешних воздействий, сложность системы в целом и нелинейности элементов затрудняют составление аналитических моделей. Аналитические методы применяются для простых систем и элементов, однако даже в этих случаях рассчитываются лишь линейные стационарные и нелинейные безынерционные системы. Как правило, применение аналитических моделей требует принятия некоторых упрощающих предположений. Кроме того, при использовании аналитических моделей изменения, вносимые в систему, приводят к существенной переработке описывающих их аналитических выражений.

3.2.2. Имитационное моделирование. В настоящее время наиболее часто при анализе структурных схем используются методы имитационного моделирования. Это вызвано тем, что в отличие от аналитической имитационная модель, будучи алгоритмической по способу представления, может быть создана практически для любой системы, а внесение изменений в имитационную модель существенно проще, чем в аналитическую.

Для имитационного моделирования разрабатывается соответствующая модель имитируемой системы. Модель строится так, чтобы отразить исследуемые характеристики системы (надежность, точность, производительность). Построение модели системы осуществляется на основе ее описания. Описание системы представляется в виде схем, текстов, формул, таблиц экспериментальных данных, характеризующих предлагаемую структуру и функциони-

рование системы, и содержит также характеристики внешних воздействий и окружающей среды.

В качестве блоков системы рассматриваются устройства или группы устройств, параметры которых характеризуют отдельные аспекты ее функционирования: продолжительность работы — в моделях массового обслуживания, исправность — в моделях исследования надежности и т. п. В модели реализуется описание каждого из этих блоков. Обычно каждый блок системы предварительно исследуется автономно и на основании этого исследования составляется упрощенное описание. Описание некоторых блоков может быть задано стохастически.

После разработки модели системы и реализации ее на ЭВМ в виде программы производится статистический эксперимент (рис. 3.1), состоящий из совокупности отдельных имитационных экспериментов, в каждом из которых

имитируются, с одной стороны, входные сигналы, поступающие на систему, а с другой, — состояния и поведение отдельных устройств и блоков системы. Например, моделируются входные сигналы с учетом их возможных случайных изменений, исправное либо неисправное состояние блока в каждый текущий момент времени, поток заявок на обслуживание и т. д.

Процесс имитационного моделирования может заключаться (в зависимости от целей) в моделировании прохождения сигнала через систему, в определении состояния каждого из блоков системы (исправен — неисправен) и системы в целом, в расчете ошибок или помех, возникающих при прохождении сигнала и т. д. Каждый

имитационный эксперимент моделирует некоторую случайную систему из генеральной совокупности.

Обычно при структурном проектировании имеют дело с двумя типами задач. В этих задачах либо определяется качество функционирования системы (надежность, точность, пропускная способность), либо система рассматривается с позиций теории массового обслуживания и при моделировании определяется наличие очередей запросов к устройствам, простои устройств, коэффициент использования устройств.

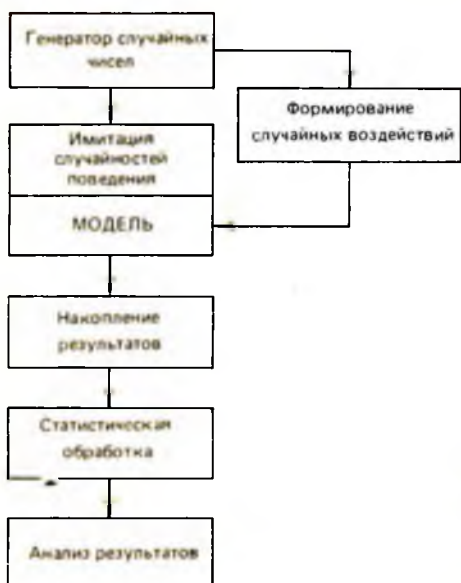


Рис. 3.1. Структура системы имитационного моделирования

Методы имитационного моделирования используются при решении задач обоих типов. Различия в решении задач заключаются в способах моделирования работы отдельных устройств. В задачах первого типа предполагается достаточно точное аналитическое (либо алгоритмическое) описание работы каждого из блоков системы с точки зрения решаемой задачи. В задачах второго типа каждый из блоков системы описывается статистическими характеристиками, определяющими время его работы (математическое ожидание времени срабатывания, дисперсия и т. п.).

Работа проектируемой системы для задач первого и второго типов анализируется путем многократного расчета реакций системы на случайные внешние воздействия или случайные вариации параметров ее устройств. Статистическая обработка результатов, полученных при моделировании, дает оценки характеристик моделируемой системы, на основе которых выполняется коррекция схемы и модели. Схематично последовательность операций структурного проектирования представлена на рис. 3.2.



Рис. 3.2. Общая схема процесса проектирования структуры

3.3. ТИПОВЫЕ ЗАДАЧИ СТРУКТУРНОГО МОДЕЛИРОВАНИЯ

3.3.1. Исследование надежности системы. Надежность системы зависит от надежности отдельных блоков и от структуры системы.

Пусть имеется структурная схема системы (рис. 3.3). Работа каждого блока описывается оператором, формирующим на выходе блока случайный процесс

$$\eta_i(t) = \begin{cases} 1, & \text{если в момент } t \text{ } i\text{-й блок системы исправен;} \\ 0, & \text{если в момент } t \text{ } i\text{-й блок системы неисправен.} \end{cases}$$

Характеристики $\eta_i(t)$ каждого блока находятся экспериментально либо по результатам расчета.

При определении работоспособности всей системы в целом определяется работоспособность каждого из ее блоков и затем по структурной схеме просчитывается надежность системы. При этом

последовательное соединение блоков i, j соответствует логическому умножению (конъюнкции) $\eta_i(t) \wedge \eta_j(t)$, а параллельное — логическому сложению (дизъюнкции) $\eta_i(t) \vee \eta_j(t)$.

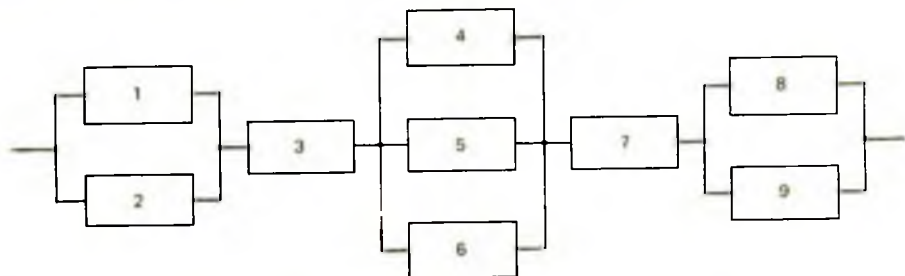


Рис. 3.3. Структурная схема системы для определения надежности

Для схемы на рис. 3.3 выражение для вычисления состояния системы выглядит следующим образом:

$$v(t) = (\eta_1(t) \vee \eta_2(t)) \wedge \eta_3(t) \wedge (\eta_4(t) \vee \vee \eta_5(t) \vee \eta_6(t)) \wedge \eta_7(t) \wedge (\eta_8(t) \vee \eta_9(t)). \quad (3.1)$$

Здесь единичное значение $v(t)$ соответствует исправности системы.

Программа реализации модели сводится к цифровой имитации процессов $\eta_i(t)$, вычислению $v(t)$ в каждый момент t по (3.1) и статистической обработке значений $v(t)$ на заданном интервале времени T . В результате многократного повторения эксперимента, имитирующего работу системы в течение заданного времени T , получают информацию о поведении системы в виде значений «исправна — неисправна» для каждого моделируемого момента. Далее по этим результатам определяют статистические характеристики системы, такие как среднее время наработки на отказ, вероятность исправного состояния в течение некоторого времени T и т. п.

3.3.2. Моделирование производительности системы. Пусть система состоит из трех устройств, имеющих времена обслуживания t_{01}, t_{02}, t_{03} .

На систему поступает поток заявок, интервал Δt между которыми случаен и подчиняется статистическому закону $\varphi(t)$. Пусть способ обслуживания заявок, называемый обычно дисциплиной обслуживания, состоит в том, что заявка поступает последовательно на каждое из трех устройств и обслуживается любым первым свободным из них, если же свободных устройств нет, заявка пропадает. Необходимо найти вероятность P обслуживания заявок за время T .

Процесс имитационного моделирования поясняется на рис. 3.4. Момент появления заявок t_i имитируется путем моделирования интервалов Δt_i по закону $\varphi(\Delta t)$ и прибавления их к текущему

времени t . Пусть первая заявка появилась в момент t_1 , вторая — в момент $t_2 = t_1 + \Delta t_1$, где Δt_1 — случайная величина, третья — в момент $t_3 = t_2 + \Delta t_2$ и т. д. В соответствии с рис. 3.4 заявки 5 и 8 пропадут. Моделируя процесс обслуживания заявок на интервале T , повторяя n раз моделирование и рассчитывая в каждом эксперименте оценку вероятности P_i обслуживания заявок как отношение обслуженных заявок к общему числу заявок, можно найти искомую величину $P = \sum_{i=1}^n P_i / n$.

Изменяя дисциплину обслуживания, например ставя заявки в очередь, можно повысить вероятность P .

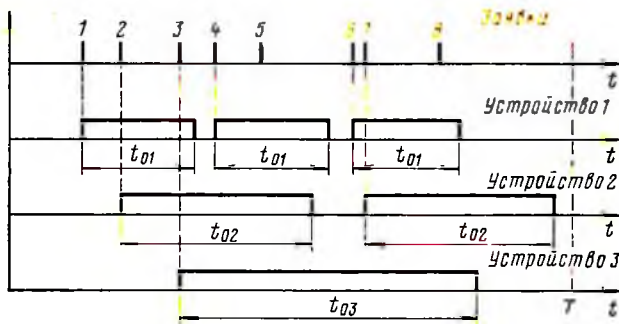


Рис. 3.4. Пример обслуживания заявок

В качестве устройств могут выступать отдельные вычислительные машины или процессоры вычислительной системы, накопители на магнитных дисках (НМД), обслуживающие запросы от процессора, коммутаторы телефонных станций, отвечающие на запросы абонентов, операторы РЛС и т. п.

3.4. МОДЕЛИ БЛОКОВ И СИГНАЛОВ

3.4.1. Общие сведения о моделях. Модели блоков и сигналов для структурного проектирования достаточно разнообразны и зависят от типа решаемых задач. Эти задачи можно разделить на три основных типа: моделирование точности работы, моделирование надежности (работоспособности) и моделирование обслуживания (пропускная способность, коэффициент загрузки или простоя и т. д.).

При моделировании точности работы модели блоков структуры представляют собой аналитические выражения или алгоритмы для вычисления ошибок на выходе блока как функций случайного соотношения его отдельных параметров и характеристик внешней среды. Вид этих моделей зависит от конкретного типа моделируемой системы и способа ее функционирования, но, как правило, в них учитываются статистические характеристики ошибок.

При моделировании надежности системы модели блоков обычно представляют собой случайные величины или случайные процессы в виде последовательности цифр, каждая из которых соответствует определенному состоянию моделируемого блока, например 0 — неисправен, 1 — исправен, 2 — неопределенное состояние, 3 — исправен, но работает в недопустимом режиме и т. д. При этом каждое состояние появляется с определенной вероятностью.

При моделировании процессов обслуживания системы модели блоков представляются, как правило, набором временных параметров, характеризующих блок, — времени подготовки обслуживания, «отдыха» после обслуживания и т. д. При этом каждое из времен может быть случайной величиной.

Модели сигналов во всех трех типах задач представляют собой случайные воздействия, формируемые с помощью генераторов случайных величин.

Таким образом, независимо от типа задач общим для моделей блоков и сигналов является отражение в них статистических свойств моделируемых объектов. Поэтому их основу составляют генераторы случайных величин, рассматриваемые далее.

3.4.2. Основные характеристики случайных величин, моделируемых на ЭВМ. Случайные величины в системах имитационного моделирования могут формироваться тремя способами:

1) из таблиц случайных чисел, 2) физическими датчиками случайных чисел, 3) программами получения случайных чисел.

Таблицы случайных чисел, полученных в результате многократного выполнения каких-либо физических опытов, например бросания игральной кости, пуска рулетки и т. п., напечатаны в специальных справочниках и могут быть записаны в память ЭВМ, используемой для имитационного моделирования. Таблицы представляют собой последовательности цифр 0, 1, ..., 9, имеющих равную вероятность появления. Использование таблиц достаточно просто. Если в ходе вычислений требуется случайная цифра N , то берется любая цифра из таблицы, если требуется случайное число x , то из таблицы берется n очередных цифр, например $x = a_1, a_2, \dots, a_n$. Цифры из таблицы можно выбирать подряд, начиная с любого места, и двигаться в любом направлении.

Использование таблиц случайных чисел гарантирует качество чисел, однако для хранения их в памяти ЭВМ требуется много места, что не всегда допустимо, кроме того, ввод их в ЭВМ требует больших затрат ручного труда.

Случайные величины могут быть получены с помощью *физических датчиков*. Чаше всего для построения датчиков используются шумящие радиоэлектронные приборы (диоды, тиратроны и т. п.). Схема устройства формирования последовательности случайных цифр 0 и 1 приведена на рис. 3.5. Генератор шума непрерывно формирует флюктуирующее напряжение. Импульсы этого напряжения, превысившие некоторый порог ограничения, подаются на счетный вход триггера. Съем результатов с триггера произ-

водится периодически с некоторым интервалом T . В течение этого интервала на триггер придет некоторое число импульсов. Если оно нечетно, то состояние триггера изменится на противоположное, если четно — останется прежним. Таким образом, формируются равновероятные цифры 0, 1, из последовательности которых можно составить случайные числа произвольной разрядности.

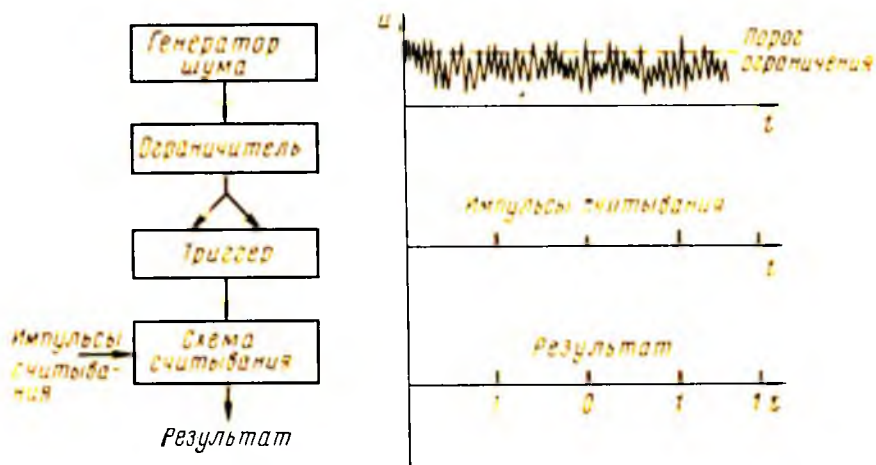


Рис. 3.5. Структура генератора случайных чисел и их последовательность

Физические датчики случайных чисел имеют большую скорость работы и генерируют «хорошие» последовательности случайных цифр (и чисел). Недостатком их является усложнение аппаратной части ЭВМ, а также невозможность повторения последовательности чисел, что затрудняет отладку программ.

Случайные числа могут быть получены в вычислительных машинах с помощью специальных программ, формирующих каждое следующее случайное число из предыдущего в результате некоторых функциональных преобразований:

$$\xi_{n+1} = F(\xi_n). \quad (3.2)$$

Получаемые с помощью программ последовательности чисел на самом деле не являются случайными, так как формируются с применением не содержащих случайности алгоритмов и могут быть повторены. Однако последовательности этих чисел весьма похожи на действительно случайные числа, а пригодность чисел для применения их в вычислительных процессах определяется в конечном счете не процессом их получения, а тем, удовлетворяют ли они некоторым принятым критериям случайности.

Числа $\xi_0, \xi_1, \xi_2, \dots$, которые вычисляются по какой-либо заданной формуле и могут быть использованы вместо случайных чисел при решении некоторых задач, называются псевдослучайными числами.

Возможность использования псевдослучайных чисел вместо случайных определяется в результате проверки их специальными тестами. Анализ равномерно распределенных псевдослучайных чисел, например, может быть произведен подсчетом количества нулей и единиц в двоичном представлении числа (их должно быть примерно равное количество), количества различных групп (00, 11, 01, 10, 000, 111, 001 и т. п.) в числе, их взаимосвязи и частоты повторения и других проверок. Однако реализация проверки случайности чисел на основе их двоичного представления достаточно сложна. На практике для проверки последовательностей псевдослучайных чисел применяются тесты, анализирующие их значение.

Наиболее часто используются следующие тесты:

тест соответствия случайных чисел принятому закону распределения, например соответствия последовательности чисел равномерному закону распределения;

тест серий, проверяющий равномерность и независимость групп (пар, троек и др.) следующих друг за другом случайных чисел;

тест интервалов, проверяющий частоту появления отдельных одинаковых случайных чисел и расстояние (количество чисел) между ними.

Для более детальной проверки используются дополнительные тесты [10, 12]. Считается, что если последовательность удовлетворяет тестам, то она пригодна для решения задач. В действительности соответствие тестам не гарантирует пригодности чисел для решения любых задач. Успешное проведение контрольных расчетов при отладке программ может явиться лучшим доказательством пригодности чисел.

Последовательности псевдослучайных чисел, полученные на ЭВМ с помощью выражений вида (3.2), являются периодически. Действительно, так как в представлении чисел любой ЭВМ используется конечное число разрядов, то в машинном слове с использованием этих разрядов может быть записано конечное число

M различных чисел и, следовательно, при формировании последовательности чисел рано или поздно одно из них совпадет с ранее вычисленным числом ξ_i . После этого формируемые числа

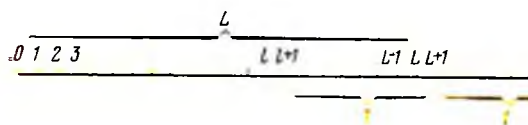


Рис. 3.6. Аперриодичность и период последовательностей псевдослучайных чисел

будут совпадать с уже имеющимися и следовать в том же порядке, т. е. $\xi_{i+1} = \xi_{L+i}$ при $i = 1, 2, \dots$ (рис. 3.6).

Из рис. 3.6 видно, что в последовательности чисел выделяют два участка — аперриодичности, на котором не наблюдается повторение числовых последовательностей, и периодических последовательностей. Количество чисел, содержащихся в отрезке аперриодичности, называется длиной отрезка аперриодичности L , а множество чисел $\xi_{i+1}, \xi_{i+2}, \dots, \xi_L$ — периодом последовательности T .

Таким образом, отрезок аperiodичности состоит из различных чисел. Для расчетов берутся числа из отрезка аperiodичности. Использование чисел, находящихся за отрезком аperiodичности, приводит к повторению расчетов с уже использовавшимися исходными данными.

Для имитационного моделирования требуются случайные числа с различными законами распределения — нормальным, равномерным, экспоненциальным и другими. Основой для их генерации является последовательность случайных чисел с равномерным законом распределения на интервале $(0 \dots 1)$. Способы получения случайных чисел с различными законами распределения из последовательности чисел с равномерным законом распределения будут рассмотрены в пп. 3.4.4—3.4.6.

3.4.3. Моделирование равномерного распределения. Наиболее часто для генерации равномерно распределенных случайных чисел применяется алгоритм вида

$$\xi_{n+1} = K\xi_n \bmod M, \quad (3.3)$$

где K — специально подобранное большое целое число, а операция умножения $K\xi_n$ производится по модулю M , т. е. если результат $K\xi_n \geq M$, то в качестве ξ_{n+1} берется остаток от деления $K\xi_n / M$. Числа ξ_i — целые. В качестве M берется максимальное целое число, размещаемое в машинном слове ЭВМ. Так, для вычислительных машин серии ЕС ЭВМ $M = 2^{31} - 1$ (длина машинного слова равна 32 разрядам, из которых один разряд используется для кодирования знака числа).

Функция вида (3.3) представляет собой множество прямых линий в квадрате с длиной стороны, равной M . Наклон и количество линий определяется величиной K . На рис. 3.7 показан график функции $\xi_{n+1} = 12\xi_n \bmod M$ и процедура получения ξ_{n+1} по ξ_n . Числа ξ_i , формируемые процедурой (3.3), располагаются на изображенных линиях. Настоящие случайные числа ϵ_i , равномерно распределенные в интервале $(0, \dots, M)$, если их представить на графике в виде точек с координатами $(\epsilon_i, \epsilon_{i+1})$, будут равномерно покрывать указанный квадрат. Отсюда следует, что для получения «хороших» случайных чисел с помощью процедур вида (3.3) надо, чтобы график этой функции представлялся большим числом линий и «плотно заполнил» квадрат, т. е. K должно быть достаточно большим.



Рис. 3.7. График функции вида $\xi_{n+1} = 12\xi_n \bmod M$

Дополнив процедуру (3.3) преобразованием

$$\xi_{n+1}^* = \xi_{n+1} / M, \quad (3.4)$$

получим случайные числа, равномерно распределенные на интервале $(0 \dots 1)$, являющиеся базовыми для формирования случайных чисел с произвольным законом распределения.

Пример 3.1. Рассмотрим подпрограмму RANDU генерации псевдослучайных чисел, которая широко используется в вычислительных машинах серии ЕС ЭВМ. В подпрограмме RANDU [4] применяются алгоритмы (3.3) и (3.4) с $K=65539$ и $M=2147483647$, т. е. $M=2^{31}-1$. Отрезок аperiodичности последовательности случайных чисел, генерируемых программой, равен примерно 2^{29} . При первом обращении к программе ей задается начальное число ξ_0 , которое рекомендуется выбирать целым, нечетным. Программа генерирует два числа: ξ_i — целое в интервале $1 \dots 2147483647$, и ξ_i^* — действительное нормированное в интервале $0 \dots 1$. При необходимости генерировать следующее число производится повторное обращение к программе, причем в качестве исходного берется число ξ_i , полученное при предыдущем обращении.

Программа написана на языке Фортран (рис. 3.8). Операция взятия модуля произведения $K\xi_i$ выполняется автоматически за счет переполнения разрядной сетки представления целых чисел. Обращение к программе осуществляется оператором CALL RANDU (IX, IY, ZFL), где IX — входное случайное число, IY — выходное случайное число, целое в интервале $1 \dots 2147483647$, ZFL — выходное случайное число с плавающей запятой в интервале $(0 \dots 1)$.

```
SUBROUTINE RANDU (IX, IY, ZFL)
  IY=IX*65539
  IF (IY) 5,6,6
5  IY=IY+2147483647+1
6  ZFL=IY
  ZFL=ZFL*.4656613E-9
  RETURN
END
```

Рис. 3.8. Подпрограмма RANDU

3.4.4. Моделирование нормального распределения. Нормальное распределение является одним из важнейших непрерывных распределений. Для выработки нормально распределенных случайных чисел разработано несколько методов. Все они базируются на использовании равномерно распределенных случайных чисел.

Один из часто применяемых методов основан на центральной предельной теореме теории вероятностей, согласно которой сумма независимых случайных величин ε_i с произвольными законами распределения и мало различающимися дисперсиями образует последовательность $\gamma = \sum_{i=1}^n \varepsilon_i$ случайных величин с законом распределения, стремящимся к нормальному при $n \rightarrow \infty$. В качестве ε_i можно использовать равномерно распределенные случайные величины на интервале $(0 \dots 1)$. На практике уже при $n \geq 8$ получается последовательность γ с законом распределения, достаточно близким к нормальному. Математическое ожидание таких распределений $M(\gamma) = n/2$, а дисперсия $D(\gamma) = n/12$. Нормированное рас-

пределение с $M(\gamma)=0$ и $D(\gamma)=1$ можно получить, воспользовавшись следующим алгоритмом:

$$\gamma = \frac{1}{n} \left(\sum_{i=1}^n \xi_i - \frac{n}{2} \right),$$

где ξ_i — случайные числа, равномерно распределенные на интервале $(0 \dots 1)$. При $n=12$

$$\gamma = \sum_{i=1}^{12} \xi_i - 6.$$

Другим методом получения нормально распределенных случайных чисел является метод полярных координат [12]. Согласно этому методу числа вычисляются по формулам

$$\gamma_1 = \sqrt{-2 \ln(\xi_1)} \sin(2\pi\xi_2),$$

$$\gamma_2 = \sqrt{-2 \ln(\xi_1)} \cos(2\pi\xi_2),$$

где ξ_1, ξ_2 — два числа с равномерным распределением на $(0 \dots 1)$. Получающаяся последовательность чисел нормирована, т. е. $M=0, D=1$.

Погрешности метода полярных координат значительно меньше, чем предыдущего метода, однако затраты времени ЭВМ на формирование чисел больше, так как требуется вычислять $\ln, \sin, \cos$.

Рассмотренные методы дают представление о способах генерации чисел с нормальным распределением. Более детальное изложение методов дано в [10, 12]. Выбор конкретного метода определяется требованиями решаемой задачи (точность, время решения) и характеристиками используемой ЭВМ (производительность, объем памяти, система команд и т. д.).

3.4.5. Моделирование дискретного распределения. Если необходимо получить ряд дискретных случайных величин $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ с вероятностями появления $P_1, P_2, \dots, P_n$, то, используя равномерно распределенную последовательность случайных чисел ξ_i , можно это сделать следующим образом.

Разделим интервал $(0 \dots 1)$ на n отрезков δ_i длина каждого из которых равна P_i (заметим, что $\sum \delta_i = 1$). Для каждого ξ_i определим интервал δ_i , в который оно попадает, и тем самым — событие, которое произойдет, т. е. определим ε_i в соответствии с условием

$$\varepsilon = \begin{cases} \varepsilon_1, & \text{если } 0 \leq \xi_k < \delta_1, \\ \varepsilon_2, & \text{если } \delta_1 \leq \xi_k < \delta_2, \\ \dots & \dots \\ \varepsilon_n, & \text{если } \delta_{n-1} \leq \xi_k < \delta_n. \end{cases}$$

Пример 3.2. Получим равномерно распределенную последовательность десятичных цифр. Вероятность появления каждой из цифр 0, 1, 2, ..., 9 равна

$p_{(0)}=p_{(1)}=p_{(2)}=\dots=p_{(9)}=0,1$. следовательно, интервал $(0 \dots 1)$ необходимо разделить на отрезки δ_i длиной 0,1. В процессе генерации каждое равномерно распределенное число ξ_i анализируется и определяется интервал δ_i , в который оно попадает. Тем самым определяется текущее значение K в последовательности.

Пример 3.3. Для моделирования случайного события, имеющего вероятность P , интервал $(0 \dots 1)$ делится на два участка: P и $1-P$. Далее генерируются равномерно распределенные случайные числа и определяется их попадание в соответствующий участок. Если попадание произошло в участке P , то событие имело место, т. е. имеет значение 1, если в участок $1-P$, то событие не произошло, значение его равно 0.

3.4.6. Моделирование произвольного распределения. Одним из наиболее общих методов моделирования произвольных законов распределения на основе равномерного является метод обратной функции. Метод основан на следующей теореме. Если случайная величина x имеет плотность распределения вероятности $p(x)$, то случайная величина $\xi = F(x) = \int_{-\infty}^x p(x) d(x)$ распределена равномерно в интервале $(0 \dots 1)$ независимо от вида $p(x)$ (рис. 3.9). Отсюда вытекает способ моделирования случайных чисел x_i с про-

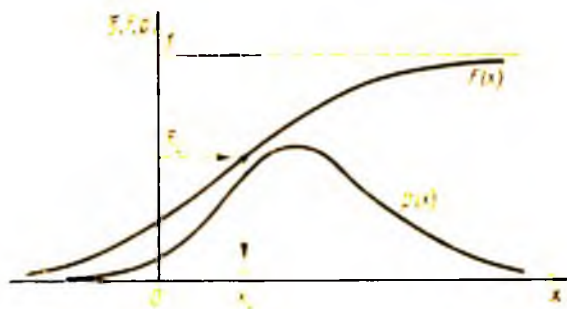


Рис. 3.9. Метод обратной функции

извольной плотностью распределения вероятности $p(x)$.

1. Моделируется равномерно распределенная случайная величина ξ_i из интервала $(0 \dots 1)$.

2. Решается интегральное уравнение

$$\xi_i = \int_{-\infty}^{x_i} p(x) dx \quad (3.5)$$

относительно верхнего предела x_i . Значение x_i будет случайным числом из совокупности чисел, имеющих плотность распределения вероятности $p(x)$. Если $p(x)=0$ при $x \rightarrow x_0$, то нижний предел $-\infty$ можно заменить на x_0 .

Графически эта процедура представлена на рис. 3.9. Здесь для каждой случайной величины ξ_i , равномерно распределенной на интервале $(0 \dots 1)$, находится соответствующая величина x_i , у которой плотность распределения вероятности связана с $F(x)$ по (3.5).

Пример 3.4. Рассмотрим моделирование экспоненциального распределения. Для экспоненциальной случайной величины x_i , распределенной при $x_0 < x < \infty$ по закону $p(x) = ae^{-a(x-x_0)}$, в соответствии с (3.5) имеем

$$\xi_i = \int_{x_0}^{x_i} a e^{-a(x-x_0)} dx,$$

откуда

$$x_i = x_0 - (1/a) \ln(1 - \xi_i).$$

Практически метод обратной функции применим не для всех распределений, так как в некоторых случаях обратная функция не может быть найдена, а в других она приводит к слишком сложным и громоздким вычислениям. В этом случае можно воспользоваться методом Неймана генерирования непрерывных случайных величин с произвольным законом распределения, который заключается в следующем.

Если случайная величина x определена на конечном интервале $(a \dots b)$ и ее плотность распределения вероятности ограничена: $p(x) \leq M_0$ (рис. 3.10), то получить значения x_i можно следующим образом.

1. Генерируются два значения ξ' и ξ'' случайной величины ξ , равномерно распределенной на $(0 \dots 1)$, и строится случайная точка $C(\eta', \eta'')$ с координатами $\eta' = a + \xi'(b-a)$, $\eta'' = \xi'' M_0$.

2. Если точка C лежит под кривой $y = p(x)$, то полагается $x_i = \eta'$, если же точка C лежит над кривой $y = p(x)$, то пара значений (ξ', ξ'') отбрасывается и выбирается новая пара значений. Очевидно, чем больше значение $y = p(x)$, тем чаще испытания будут успешными ($\xi'' < p(x)$) и соответственно больше моделируемая плотность вероятности, что и требуется.



Рис. 3.10. Метод Неймана

3.5. РЕАЛИЗАЦИЯ ПРОГРАММ СТРУКТУРНОГО МОДЕЛИРОВАНИЯ

Задачи структурного моделирования, заключающиеся в определении таких характеристик, как точность, надежность, чувствительность, полоса пропускания системы и т. п., т. е. задачи определения технических характеристик РЭА обычно решаются с использованием универсальных процедурно-ориентированных алгоритмических языков типа Фортран, ПЛ/1.

Задачи моделирования структур с целью определения пропускной способности и производительности систем, простоев и занятости устройств, т. е. задачи, решаемые с позиций систем массового обслуживания, достаточно специфичны. Так, в них необходимо учитывать время моделирования, моделировать внешнюю среду путем генерации потока запросов, моделировать реакции устройств,

работающих параллельно в системе, создавать очереди к устройствам, занимать и освобождать устройства, учитывать статистические данные о работе системы и т. д. Реализация перечисленных операций с использованием универсальных алгоритмических языков затруднительна. Поэтому для задач имитационного моделирования разработаны специальные проблемные языки. К ним относятся Симула, Симскрипт, GPSS, SMPL и другие.

Эти специализированные языки предназначены только для моделирования. Фактически за этими названиями стоят системы моделирования с собственными языками описания задач. Так, в них имеются встроенные счетчики модельного времени, встроенные генераторы случайных сигналов и воздействий, программы сбора и обработки статистических данных и т. д. Эти системы представляют собой сложные программные комплексы, обычно превосходящие по сложности системы программирования, использующие универсальные алгоритмические языки Фортран, ПЛ/1.

3.5.1. Принципы организации моделирования. При моделировании структур систем с целью выяснения динамики их работы используются понятия: процесс, активность, событие.

Под *процессом* понимают описание алгоритма работы некоторой части системы в терминах работы ее блоков. В свою очередь работа каждого блока задается *активностью*. Блок активен, если к нему был запрос и он находится в состоянии обработки запроса. Основной характеристикой активности является время. При моделировании работы блока это время задается или может быть определено. Процесс является логически связанным набором активностей. Выделение понятий активности и процесса определяется уровнем рассмотрения системы. Так, при рассмотрении на более высоком уровне процесс может рассматриваться как активность, а при рассмотрении с большей степенью детализации активность может рассматриваться как процесс и состоять из набора других активностей.

Например, в ЭВМ выполнение операции обмена информацией с накопителями на магнитных дисках (НМД) можно рассматривать как процесс, включающий в себя три активности: установки головок считывания записи, задержки времени на время вращения носителя с нужной записью и передачи данных. Процесс более высокого уровня — выполнение программы — может содержать активности вычисления и обмена информацией с диском. Программа в свою очередь может рассматриваться как составная часть, т. е. активность задания.

Активность (т. е. выполнение каких-либо операций) инициируется в результате совершения событий. *Событие* — это изменение состояния какого-либо объекта системы либо запрос извне системы. Окончание работы какого-либо блока — окончание активности — является событием, которое может возбудить другие активности.

Процессы, активности и события являются основными элементами, с помощью которых можно описать динамическое поведение системы.

Пример 3.5. Рассмотрим систему, в состав которой входят два НМД и устройство управления вместе с каналом, через который происходит обмен данными между НМД и оперативной памятью (рис. 3.11). Запросы от процессора могут быть направлены к любому из двух НМД. При появлении запроса определяется, свободен ли запрашиваемый НМД, и если это так, то начинается позиционирование его головок в требуемое положение. Если НМД занят, запрос помещается в очередь. После завершения позиционирования головок определяется, свободен ли канал. Если канал занят, запрос помещается в очередь. Если канал свободен, он начинает выполнять операцию передачи информации и остается занятым до окончания операции. По окончании передачи НМД и канал освобождаются, и начинается обработка запросов, находящихся в очереди.

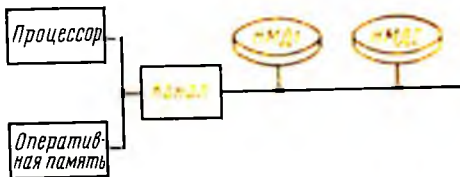


Рис. 3.11 Пример структуры ЭВМ с НМД

Таким образом, обращение к НМД (запрос) инициирует процесс, состоящий из двух активностей: активности позиционирования и активности канала. Время позиционирования определяется временем перемещения блока головок на требуемую дорожку; время активности канала складывается из времени поворота диска в нужное положение и времени передачи данных.

На рис. 3.12 показана одна из возможных последовательностей активностей при обращении к НМД. Здесь E_1 — запросы процессора к НМД, A_1 и A_2 — активности установки головок и работы канала соответственно, E_2 и E_3 — события, соответствующие завершению активностей. Сплошной линией на рисунке показаны состояния активностей, пунктирной — ожидание выполнения необходимых условий для перехода в эти состояния. Запрос к НМД1 событие E_1 инициирует установку головок (активность A_1), а конец ее (событие E_2) инициирует работу канала (активность A_2). Повторный запрос к НМД1 во время A_1 помещается в очередь и удовлетворяется лишь по окончании A_2 по первому запросу. Выполнение A_2 по второму запросу после установки головок НМД1 начинается лишь после того, как освободится канал, занятый работой с НМД2.

Структурное моделирование динамики поведения системы сводится к следующим основным действиям.

1. Моделирование времени поступления внешних запросов. Момент их появления определяется в соответствии с известными ста-

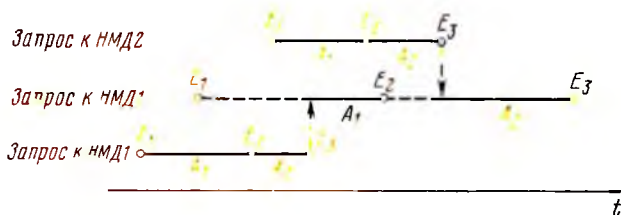


Рис. 3.12. Пример запросов к НМД и их обработки

тистическими характеристиками внешних процессов. Например, запросы от процессора к дисковой памяти имеют обычно экспоненциальное распределение с математическим ожиданием, определяемым характером решаемых задач и производительностью процессора.

2. Моделирование действий, соответствующих событиям, происходящим в системе (определение реакции на внешние запросы и на окончание активностей). К этим действиям относятся освождение и занятие отдельных блоков системы, определение выполнения условий, требуемых для инициации следующих активностей, инициации активностей и др.

3. Моделирование интервалов времени активных состояний блоков системы. Время активного состояния блока вычисляется на вероятностной или детерминированной основе в зависимости от принципов работы блока.

4. Наблюдение за поведением системы и накопление статистических сведений в течение заданного интервала времени моделирования. Определяется время простоя блоков, наличие и размеры очередей к ним, процент занятости во времени и т. п.

Важным моментом в программах моделирования является способ слежения за временем и его учет. В ранних программах время постепенно увеличивалось на некоторую достаточно малую величину Δt , при этом каждый раз анализировалось состояние моделируемой системы. В настоящее время обычно используется событийный принцип, который заключается в том, что состояние моделируемой системы анализируется в моменты возникновения событий в модели. Технически это реализуется с применением списка будущих событий.

В этот список в процессе моделирования упорядоченно (в порядке увеличения времени) записывается каждое событие, которое должно произойти в системе, и момент времени, когда оно произойдет. Время же появления событий определяется длительностью выполнения активностей, либо при помощи моделирования внешних запросов. В том и другом случае оно либо известно, либо вычисляется на основании характеристик процессов. Текущее событие, обрабатываемое системой, считывается из вершины списка. Это событие инициирует какие-либо активности. Для этих активностей вычисляются моменты завершения, которые и заносятся в список будущих событий. Затем считывается следующее событие из списка, выполняются необходимые действия и т. д. В моменты считывания событий происходит изменение показаний счетчика модельного времени, который всегда показывает время наступления последнего обработанного события.

3.5.2. Особенности языков структурного моделирования. Языки структурного моделирования строятся с использованием понятий процессов, активностей и событий. В зависимости от подхода к описанию рассматриваемых объектов их можно разделить на языки, ориентированные на активности, события или процессы. Из языков, ориентированных на события, наибольшее распространение

ние получил Симскрипт. Языки Симула, GPSS, ASPOL ориентированы на процессы.

В языках, ориентированных на события, моделирующая программа организована в виде совокупности процедур событий. Каждая процедура состоит из набора операций, выполняющихся после завершения какой-либо активности, т. е. при возникновении определенных событий в системе.

На рис. 3.13 представлена структура программы, ориентированной на события. Программа состоит из иницирующей секции, управляющей секции и ряда событийных секций. В иницирующей секции описывается структура моделируемой системы, производится начальная установка времени, задается интервал времени моделирования, осуществляется запуск программы моделирования.

Управляющая секция устанавливает, какое событие следует первым по графику, выбирает первый элемент из списка событий и передает управление соответствующей событийной секции.

Событийная секция выполняет действия, определенные концом предыдущей активности, выявляет, какие активности должны и могут выполняться в текущий момент, и планирует события, возникающие при завершении этих активностей. После выполнения событийной секции управление передается в управляющую секцию.

Пример 3.6. Рассмотрим структуру ориентированной на события программы моделирования системы из примера 3.5.

Моделирование проводится с целью определения соответствия скорости работы системы (канала и НМД) и последовательности запросов, поступающих от процессора. Предполагается, что статистические параметры запросов и обрабатывающих устройств известны. В процессе моделирования выявляется наличие очередей запросов как ко всей системе, так и к отдельным ее устройствам, максимальный размер очередей, простои системы и ее компонентов, их средние и максимальные значения и т. д.

Структура программы соответствует рис. 3.13. В модели рассматриваются пять типов событий: событие 1 — запрос к системе от процессора; событие 2 — запрос к конкретному НМД; событие 3 — завершение позиционирования головок считывания-записи; событие 4 — окончание обмена информацией; событие 5 — завершение моделирования. Рассмотрим действия, выполняемые каждой программной секцией, применительно к данной задаче.

В иницирующей секции указывается состав моделируемой системы — два дисковых накопителя и канал передачи информации; указывается возможность

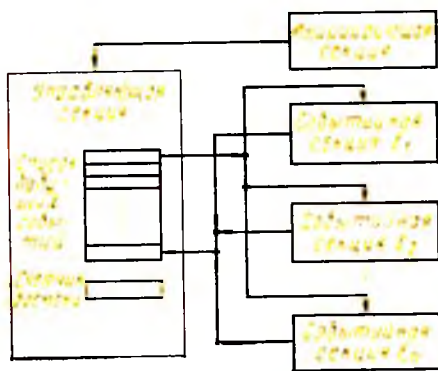


Рис. 3.13. Структура событийно-организованной программы

образования очередей запросов к каждому из устройств; в список будущих событий записывается событие 1 — первый запрос к системе и событие 5; для каждого из событий указывается время его возникновения.

Событийная секция 1 обрабатывает запрос к системе. В этой же секции генерируется следующий запрос к системе, который помещается в список будущих событий, в этот же список заносится событие 2. Событие 2 инициирует установку блока головок НМД на нужную дорожку.

Событийная секция 2 определяет состояние запрашиваемого НМД, если диск вращается свободно, то занимает его, а если занят, то ставит запрос в очередь. При занятии НМД генерируется время установки головок и в список событий записывается событие 3 с полученным временем.

Событийная секция 3 обрабатывает ситуацию, возникающую при завершении позиционирования головок, и выполняет следующие действия. Если канал свободен, он резервируется за текущим запросом, и в список событий заносится событие 4, время возникновения которого вычисляется в этой же секции. Если канал занят, то запрос помещается в очередь.

Событийная секция 4 соответствует окончанию обмена информацией. В этой секции освобождаются ранее зарезервированные НМД и канал, анализируются очереди к каналу и НМД, и если в любой из них имеются запросы, то выбирается запрос, находящийся в вершине списка событий, и для него планируется соответствующее событие.

Событие 5, предварительно занесенное в список событий во время инициации программы, завершает работу программы и распечатывает всю накопленную статистическую информацию о работе моделируемой системы.

Управляющая секция выбирает событие из вершины списка будущих событий, записывает текущее время и передает управление соответствующей событийной секции.

Рассмотренный пример демонстрирует принципы работы программ.

В языках, ориентированных на процессы, таких как Симула, GPSS, моделирующая программа организуется в виде набора описаний процессов. Для каждого типа процесса, который может протекать в системе, имеется собственная процедура. Фактически описание процесса — это подпрограмма, в которой определяется последовательность выполнения действий, моделируются интервалы времени работы устройств, производится занятие и освобождение устройств. Каждая из подпрограмм процессов построена таким образом, что может выполняться одновременно для нескольких различных процессов одного типа. Например, если в моделируемой системе имеется несколько подсистем, каждая из которых включает в себя каналы и НМД, то для моделирования их работы используется одна и та же подпрограмма процесса.

Подпрограммы процессов — это реентерабельные программы, которые могут выполняться параллельно для различных процессов. Учет модельного времени здесь, как и в предыдущем случае, ведется с помощью списка будущих событий. Генерация событий и моментов времени их возникновения осуществляется процедурами процессов. При моделировании активности выполнение про-

цедуры процесса задерживается на время активности и продолжается далее по истечении заданного времени.

Сравнивая языки, ориентированные на процессы, и языки, ориентированные на события, следует отметить, что с помощью первых проще получить адекватность поведения модели и системы. Кроме того, структура программы-модели более соответствует структуре моделируемой системы, что весьма важно при моделировании сложных систем. Объем программ получается меньше. Однако разработка самой программы моделирования проще для событийно-ориентированного языка.

ГЛАВА 4

ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ

4.1. ПОСТАНОВКА ЗАДАЧИ

Одним из важных вопросов, которые приходится решать при проектировании сложных радиоэлектронных устройств (РЭУ), является исследование процесса преобразования сигнала по мере его прохождения от входа к выходу РЭУ. Это исследование можно проводить на ЭВМ с разной степенью точности — либо весьма точно, что требует больших затрат машинного времени, либо достаточно приближенно, но быстро. В первом случае РЭУ моделируется на основе физического подхода методами схемотехнического (электрофизического) моделирования, а во втором — на основе информационного подхода методами функционального (электроинформационного) моделирования.

Суть функционального моделирования (ФМ) состоит в разбиении РЭУ на отдельные функциональные блоки (элементы 1...5 на рис. 4.1), каждый из которых выполняет то или иное функциональное преобразование сигнала (усиление, ограничение, интегрирование и т. д.), и расчете формы сигнала и его основных параметров в каждой точке полученной функциональной схемы.

Под *формой сигнала* понимается либо зависимость сигнала от времени $x(t)$ при моделировании во временной области, либо эквивалентное представление сигнала в виде изображения по Лапласу $x(p)$ или зависимости от комплексной частоты $j\omega$ при моделировании в частотной области.

Основным требованием при ФМ является высокая скорость моделирования, необходимая для того, чтобы за короткое время можно было исследовать большое число различных вариантов функциональных схем.

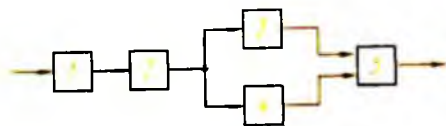


Рис. 4.1. Представление устройства для функционального моделирования

Поскольку этап проектирования функциональной схемы относится к начальным этапам общего процесса проектирования, то высокой точности от ФМ не требуется.

Первым основным допущением, характерным для ФМ, является *развязка* отдельных блоков функциональной схемы, т. е. независимость характеристик отдельных блоков от режима работы других блоков. Условие развязки блоков эквивалентно выполнению условий $R_{вх} = \infty$, $R_{вых} = 0$ для каждого из блоков. Вследствие этого преобразование сигнала зависит только от характеристик вход-выход каждого блока, а не от их взаимного влияния друг на друга.

Вторым основным допущением, характерным для ФМ, является допущение об *однаправленности* элементов, т. е. сигнал на выходе любого элемента не влияет на сигнал на его входе. Это позволяет считать, что сигнал в функциональных схемах распространяется *однаправленно* — от входа к выходу каждого элемента.

Заметим, что при составлении функциональных схем нельзя допускать объединения (замыкания) выходов различных элементов, например элементов 3 и 4 на рис. 4.1, поскольку это приводит к неопределенности выходного сигнала.

Характерным примером задач, решаемых методами ФМ, является исследование поведения системы автоматического регулирования, функциональные схемы которых состоят из типовых звеньев — дифференцирующих, интегрирующих, форсирующих, чистой задержки и т. д. Другим примером может служить исследование временных диаграмм работы аналого-цифровых и логических схем методами логического моделирования. Вследствие специфических особенностей и важности логического моделирования как одного из видов функционального моделирования его рассмотрению посвящена гл. 6.

Наконец, в ряде случаев к ФМ прибегают потому, что исследуемые процессы весьма сложны и их моделирование, например на основе теории цепей, требует слишком много времени даже при небольшом числе уравнений. Характерным примером служит исследование процессов в радиочастотных устройствах (рис. 4.2). Если проводить расчет высокочастотного модулированного колебания (точки 1, 2, 3, ...) на уровне электрических цепей, для кото-

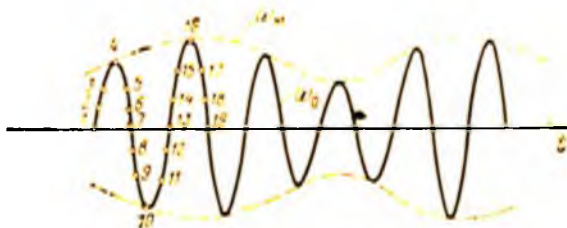


Рис. 4.2. Результаты моделирования радиочастотного процесса на несущей частоте (точки 1, 2, 3, ...) и на огибающей (точки 4, 16, ...)

рого необходимо получить численное решение уравнений схемы в каждой временной точке t_n , то моделирование потребует огромного количества времени.

Так, если несущая частота $\omega_0 = 10^6$ Гц, частота модуляции $\omega_m = 1$ кГц, а для достоверного воспроизведения каждого высокочастотного периода взять $N_{\omega_0} = 10$ точек, то для моделирования лишь одного периода модуляции потребуется N точек: $N = (\omega_0 / \omega_m) N_{\omega_0} = (10^6 / 10^3) 10 = 10^4$. Если время однократного расчета модели схемы во временной области принять равным 1 с, то расчет N точек займет 10^4 с ≈ 170 мин, что обычно неприемлемо. Поэтому целесообразно рассчитывать только те точки, в которых содержится информация об огибающей (точки 4, 16, ... на рис. 4.2). Однако расчет этих точек на электрическом уровне также весьма трудоемок, в связи с чем моделирование радиочастотных устройств во временной области чаще всего выполняется на функциональном уровне.

4.2. БАЗОВЫЕ ЭЛЕМЕНТЫ ФУНКЦИОНАЛЬНЫХ СХЕМ

Несмотря на многообразие функциональных элементов их можно свести к следующим четырем основным типам, которые будем называть *базовыми*: генераторы сигналов, безынерционные элементы, инерционные линейные элементы, инерционные нелинейные элементы (рис. 4.3). Основной характеристикой элемента при ФМ является его функция преобразования, связывающая входной и выходной сигналы. Рассмотрим вид этих функций для указанных четырех типов базовых элементов.

4.2.1. Генератор сигналов. Этот тип элементов включает две разновидности — независимые генераторы, задающие сигнал $x(t)$ на входе функциональной схемы, и управляемые генераторы, формирующие ту или иную форму сигнала $x(t)$ в зависимости от управляющего воздействия u . Функция преобразования управляемого генератора имеет вид

$$x(t) = \begin{cases} x_1(t) & \text{при } u = u_1, \\ \vdots & \vdots \\ x_n(t) & \text{при } u = u_n. \end{cases} \quad (4.1)$$



Рис. 4.3. Базовые элементы функциональных схем:
 а — генератор сигнала; б — безынерционный элемент; в, г, д — линейные инерционные элементы; е — нелинейный инерционный элемент; ж, з — эквивалентное представление нелинейного инерционного элемента

Типичным примером управляемых генераторов являются различные пороговые устройства, выходной сигнал которых зависит от превышения уровнем входного сигнала некоторого порогового значения.

4.2.2. Безынерционный линейный или нелинейный элемент. Функция преобразования этого элемента представляет собой линейную или нелинейную функцию f , связывающую входной сигнал x и выходной сигнал y :

$$y = f(x). \quad (4.2)$$

Важным частным случаем является безынерционный элемент с памятью, например триггер. Его функция преобразования неоднозначна по отношению к входному сигналу x и зависит не только от x , но и от состояния элемента s : $y = f(x, s)$.

Безынерционный нелинейный элемент позволяет на основе (4.2) преобразовать форму входного сигнала $x(t)$ в любую форму выходного сигнала $u(t) = f(x(t))$, например ограничивать амплитуду $x(t)$, формировать из импульса $x(t)$ импульс другой формы $y(t)$. Основные типы безынерционных элементов приведены в табл. 4.1.

Разнообразие типов безынерционных элементов определяет широту их применения. Чаще всего их используют для построения статических временных диаграмм устройств (без учета их динамических свойств, т. е. инерционности) с целью проверки правильности функционирования при изменении параметров сигнала и устройства в заданном диапазоне.

4.2.3. Инерционный линейный элемент. Его функция преобразования во временной области — это переходная характеристика $h(t)$, а в частотной области — коэффициент передачи $K(p)$. В первом случае

$$y(t) = \int_0^t x(\tau) h(t - \tau) d\tau, \quad (4.3a)$$

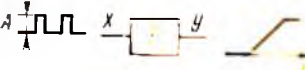
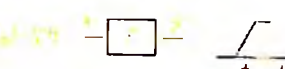
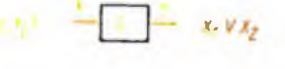
а во втором

$$y(p) = K(p)x(p). \quad (4.3b)$$

К инерционным линейным элементам относится весьма широкий класс устройств, например различные типы частотных фильтров, линейные высокочастотные, полосовые и низкочастотные усилители, операционные усилители, выполняющие функции суммирования, интегрирования, дифференцирования и другие линейные операции и т. д.

4.2.4. Инерционный нелинейный элемент. Его функция преобразования есть некоторый нелинейный оператор $A(x)$, например дифференциальное уравнение, ставящий в соответствие каждой реализации $x(t)$ реализацию $y(t)$. Дифференциальные уравнения являются наиболее общей и универсальной формой представления моделей инерционных элементов любого типа — линейных и нелинейных. Эта форма представления удобна для автоматизации функци-

Таблица 4.1

Тип элемента	Выполняемое преобразование	Примечание
Аналого-аналоговый		$y = f(x)$
Аналого-импульсный		$\tau = f(x(t))$ — широтно-импульсная модуляция
Импульсно-аналоговый		$y(t) = k(A)t$
Аналого-цифровой		$N_y(t) = kx(t)$
Цифро-аналоговый		$y(t) = kN_x(t)$
Аналого-логический		$y = 1, 1/2, 0$ — значения логической переменной
Логико-аналоговый		y_1, y_3 — электрические уровни, $y_2 = y_2(t)$ — форма сигнала y
Цифровой		$N_y = f(N_x)$
Логический		$y = L(x_1, x_2)$, L — логическая функция
Логико-цифровой		$N_y = L(x_1, x_2)$
Логико-импульсный		импульсы на выходе при $L(x_1, x_2) = 1$

онального проектирования на ЭВМ, так как легко вписывается в аппарат численных методов, используемый для функционального моделирования. В большинстве случаев реальный инерционный элемент можно описать системой дифференциальных уравнений вида

$$\frac{dy_i(t)}{dt} + g_i(y(t)) = \frac{dx_i(t)}{dt} + f_i(x(t)), \quad i = \overline{1, n}, \quad (4.4)$$

где $x(t)$, $y(t)$ — входные и выходные сигналы элемента; n — порядок системы.

В общем случае дифференциальное уравнение нелинейного инерционного элемента может быть любого порядка и содержать любые нелинейности, поэтому вместо (4.4) его можно представить в более общем виде

$$\begin{aligned} & F_i(y^{(k)}(t), y^{(k-1)}(t), \dots, y(t), t) = \\ & = \Phi_i(x^{(p)}(t), x^{(p-1)}(t), \dots, x(t), t), \end{aligned} \quad (4.5)$$

где k, p — максимальный порядок производных; $i = \overline{1, n}$.

В тех случаях, когда составление модели инерционного нелинейного элемента в форме дифференциальных уравнений затруднительно, этот элемент приближенно моделируют последовательным соединением инерционного линейного элемента и безынерционного нелинейного элемента (рис. 4.3, ж, з), так что соответственно имеем

$$\begin{aligned} y(t) &= f\left(\int_0^t x(\tau) h(t-\tau) d\tau\right), \\ y(t) &= \int_0^t f(x(\tau)) h(t-\tau) d\tau. \end{aligned}$$

Заметим, что выходной сигнал существенно зависит от последовательности включения нелинейного и инерционного элементов. Отметим также что схема на рис. 4.3, ж имеет большую вычислительную погрешность, чем схема на рис. 4.3, з, так как сигнал на выходе инерционного элемента включает ошибку численного расчета интеграла, которая затем усиливается в нелинейном элементе. Это не означает, однако, в силу первого замечания, что модель рис. 4.3, з всегда более точная, так как точность моделирования определяется не только вычислительной погрешностью, но и точностью модели в целом.

Для обобщенного описания структуры моделей инерционных нелинейных элементов введем операторы: T — моделирования временных характеристик, H — моделирования нелинейных свойств, L — моделирования линейных свойств, а также знаки: « $\cdot$ » — для обозначения совместного (параллельного) действия операторов и « $+$ » — для обозначения последовательного действия этих операторов. Тогда структура модели на рис. 4.3, ж может быть описана как $M = LT + H$, на рис. 4.3, з — как $M = H + LT$, а исходная недекомпозированная модель — как $M = LTH$, где M — оператор моделирования всего элемента.

Очевидно, этот способ можно применить для описания более сложных структур нелинейных инерционных элементов, в которых приведенные операторы могут относиться к входной, преобразующей и выходной частям. Например, структура $M = L_{\text{вх}} T_{\text{вх}} + H_{\text{пр}} + L_{\text{вых}} T_{\text{вых}}$ состоит из последовательного соединения модели линейной инерционной входной части $L_{\text{вх}} T_{\text{вх}}$, модели нелинейной безынерционной преобразующей (функциональной) части $H_{\text{пр}}$ и модели линейной инерционной выходной части $L_{\text{вых}} T_{\text{вых}}$. Структуру

ра $M = \mathcal{L}_{\text{вх}} + T_{\text{пр}} H_{\text{пр}} + \mathcal{L}_{\text{вых}}$ содержит последовательно включенные модели безынерционной линейной входной части, инерционной нелинейной преобразующей части и безынерционной линейной выходной части.

4.3. АЛГОРИТМЫ МОДЕЛИРОВАНИЯ БАЗОВЫХ БЕЗЫНЕРЦИОННЫХ ЭЛЕМЕНТОВ

Под моделированием элемента понимается вычисление его выходной величины y по заданному значению сигнала x . Рассмотрим сначала методы моделирования отдельных базовых элементов, после чего перейдем к методам моделирования структур из этих элементов.

4.3.1. Генераторы сигналов. Моделирование генератора сигнала $x(t)$ заключается в вычислении заданной функции $x(t)$ в известные моменты времени t_n . В результате непрерывная функция $x(t)$ заменяется дискретной решетчатой функцией $x_n = x(t_n)$.

Если вычисление производится часто и каждый расчет x_n требует значительного времени, то для экономии времени можно при вычислении новых x_n использовать известные старые значения $x_{n-1}, x_{n-2}, \dots$, заменяя сложную функцию $x_n = x(t_n)$ более простой рекуррентной функцией $x_n = f(x_{n-1}, x_{n-2}, \dots)$ [15, 16].

Пример 4.1. Пусть нужно смоделировать сигнал $x(t) = ke^{at}$. Для моделирования используем следующее рекуррентное соотношение:

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

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

$$x_n = A x_{n-1}, \quad x_0 = k,$$

избегая вычисления в каждой точке нового значения экспоненты.

Аналогично, при моделировании сигналов $x_1(t) = \sin at$, $x_2(t) = \cos at$, представляя $\sin at_n = \sin a(t_{n-1} + \Delta t)$, $\cos at_n = \cos a(t_{n-1} + \Delta t)$, легко получить следующие рекуррентные формулы:

$$x_{1,n} = A_1 x_{1,n-1} + A_2 x_{2,n-1},$$

$$x_{2,n} = A_1 x_{2,n-1} - A_2 x_{1,n-1},$$

где $A_1 = \cos a\Delta t$, $A_2 = \sin a\Delta t$, $x_{1,0} = 0$, $x_{2,0} = 1$. Сигналы $x_{1,n}$ и $x_{2,n}$ моделируются здесь попеременно.

4.3.2. Безынерционные статические элементы. К этим элементам относятся элементы из табл. 4.1 с преобразующими функциями статического типа, т. е. когда преобразующая функция $y = f(x)$ может быть вычислена в любой момент времени, в отличие от безынерционных динамических элементов, в которых значение y может быть определено лишь по прошествии некоторого интервала времени, т. е. является дискретной величиной.

Обычно с помощью статических безынерционных элементов моделируются различного рода нелинейности характеристик реальных элементов — смесителей, детекторов, ограничителей, ключей и др. Моделирование элементов без памяти сводится к вычислению функции $y=f(x)$ для любого заданного значения x , а моделирование элемента с памятью — к вычислению одной из функций

$$y = \begin{cases} f_1(x) & \text{при } S = S_1, \\ f_2(x) & \text{при } S = S_2. \end{cases}$$

Например, для элемента с памятью — триггера — функция вычисляется либо по верхней части гистерезисной кривой (рис. 4.4), если триггер находится в устойчивом состоянии $S=1$, либо по нижней части, если $S=0$.

4.3.3. Безынерционные динамические элементы. Выходная величина y в этих элементах зависит не только от x , но и от интервала времени Δt . Например, для элемента логико-цифрового типа — схемы совпадения — на выходе формируется последовательный цифровой код $N=k\Delta t$ при $x=1$, и код отсутствует, т. е. $N=0$ при $x=0$. Хотя элемент безынерционный, но для получения выходного кода нужно время.

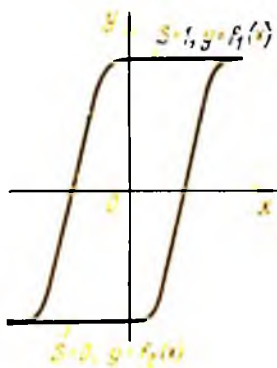


Рис. 4.4. Характеристика статического безынерционного элемента с памятью

К этому же типу элементов относятся импульсные, аналого-импульсные и импульсно-аналоговые элементы, в которых форма выходного сигнала $y(t)$ ставится в соответствие форме входного сигнала $x(t)$ по закону: если $x(t)=x_i(t)$, то $y(t)=y_i(t)$. В этом случае для определения $y_i(t)$ необходимо получить весь сигнал $x(t)$ на некотором интервале $t_1—t_2$ и проанализировать его на соответствие какому-либо $x_i(t)$.

Соответствующий сигнал $y_i(t)$ считается действующим с момента t_1 , т. е. задержка элемента отсутствует, в связи с чем элемент считается безынерционным. Такие элементы удобно использовать при моделировании функциональных схем методом полного сигнала, описанным далее.

В отличие от статических безынерционных элементов, которые можно назвать преобразователями отдельных мгновенных значений сигнала, динамические безынерционные элементы можно называть преобразователями параметров и формы полного сигнала, заданного множеством значений. Преобразующей характеристикой статического элемента является функция, ставящая в соответствие числу число, а преобразующей характеристикой динамического элемента — оператор, ставящий в соответствие функции функцию, или функционал, ставящий в соответствие функции число.

4.4. АЛГОРИТМЫ МОДЕЛИРОВАНИЯ БАЗОВЫХ ИНЕРЦИОННЫХ ЛИНЕЙНЫХ ЭЛЕМЕНТОВ

Способы моделирования этих элементов зависят от способа их задания — переходной характеристикой $h(t)$, комплексным коэффициентом передачи $K(j\omega)$ или операторным коэффициентом передачи $K(p)$, а также от области, в которой выполняется моделирование — временной или частотной, т. е. от вида входного сигнала $x(t)$ или $x(j\omega)$. Рассмотрим основные из этих способов.

4.4.1. Моделирование элементов, заданных переходной характеристикой $h(t)$, во временной области. Эти элементы моделируются на ЭВМ путем численного расчета интеграла свертки (4.3а). Обычно численное интегрирование выражения (4.3а) выполняется *методом скользящего суммирования* [15]. Суть метода состоит в следующем (рис. 4.5). Разбивая заданный интервал интегрирования $(0, t)$ и интервал текущего времени интегрирования $(0, \tau)$ на отрезки Δt , получаем

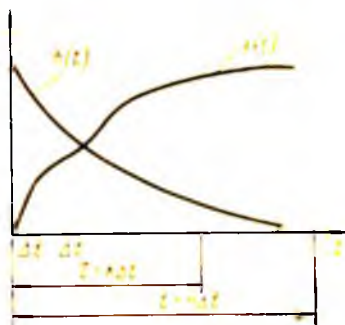


Рис. 4.5. К моделированию инерционного линейного элемента во временной области

$$t = n\Delta t, \tau = k\Delta t, t - \tau = (n - k)\Delta t.$$

Полагая на каждом из отрезков Δt значения $x(t)$ и $h(t - \tau)$ постоянными, что соответствует численному интегрированию (4.3а) методом прямоугольников, получаем при $d\tau \approx \Delta t$ вместо интеграла (4.3а) его дискретную аппроксимацию в виде суммы

$$y_n = \sum_{k=0}^n x(k\Delta t) h((n-k)\Delta t) \Delta t = \Delta t \sum_{k=0}^n x_k h_{n-k}. \quad (4.6)$$

Таким образом, вычисление выходного сигнала $y_{n-1}, y_n, y_{n+1}, \dots$ в каждый момент времени $t_{n-1}, t_n, t_{n+1}, \dots$ сводится к вычислению сумм

$$y_{n-1} = \Delta t (x_0 h_{n-1} + x_1 h_{n-2} + \dots + x_{n-1} h_0),$$

$$y_n = \Delta t (x_0 h_n + x_1 h_{n-1} + \dots + x_n h_0),$$

$$y_{n+1} = \Delta t (x_0 h_{n+1} + x_1 h_n + \dots + x_{n+1} h_0)$$

методом скользящего суммирования значений x_k со скользящим весом h_{n-k} .

Выражение (4.6) называют *дискретной сверткой* интеграла (4.3а). Очевидно, применяя вместо прямоугольников другие виды аппроксимации выражений $x(t)$, $h(t)$ на интервалах Δt , можно получить более точные формулы дискретной свертки, соответствующие вычислению интеграла методом трапеции, парабол (Симпсона) и др.

4.4.2. Моделирование элементов, заданных обыкновенным дифференциальным уравнением (ОДУ). Эти элементы моделируются во временной области на основе *дискретизации* и *алгебраизации* ОДУ вида (4.4). Дискретизация состоит в замене производных их конечно-разностными аналогами, например

$$\left. \frac{dy}{dt} \right|_{t=t_{n+1}} - \left. \frac{dy}{dt} \right|_{t=t_n} \approx \frac{y_{n+1} - y_n}{\Delta t}, \quad \left. \frac{dy}{dt} \right|_{t=t_n} \approx \frac{y_{n+1} - y_{n-1}}{2\Delta t}$$

или в более общем виде

$$\frac{dy}{dt} \approx \sum_{i=n-k}^{n+1} \alpha_i y_i$$

где α_i — коэффициенты, соответствующие различным способам дискретизации производных.

Аналогично дискретизируются производные dx/dt . В результате *дискретизации производных* уравнение (4.4) превращается в *алгебраическое уравнение* относительно y_{n+1} :

$$\alpha_{n+1} y_{n+1} + \sum_{i=n-k}^n \alpha_i y_i + g(y_p) = \sum_{i=n-k}^{n+1} \beta_i x_i + f(x_p).$$

Дискретизация производной dx/dt может выполняться либо в точке t_{n+1} , либо в какой-либо предыдущей точке. Функции $g(y_p)$, dx/dt , $f(x_p)$ должны при этом вычисляться в той же точке.

Можно показать, что если дискретизированная производная y' , а также функции $g(y_p)$, dx/dt , $f(x_p)$ вычисляются для точки t_{n+1} ($p = n+1$), то получившееся уравнение будет соответствовать неявному методу численного решения исходного ОДУ (4.4), а если для предыдущей точки t_{n+1-k} ($k = 1, 2, \dots$) — явному методу.

В первом случае вычисление y_{n+1} при известных $y_n, y_{n-1}, \dots, y_{n-k}, x_{n+1}, \dots, x_{n-k}$ требует решения уравнения

$$\alpha_{n+1} y_{n+1} + \sum_{i=n-k}^n \alpha_i y_i + g(y_{n+1}) = \sum_{i=n-k}^{n+1} \beta_i x_i + f(x_{n+1}) \quad (4.7)$$

относительно y_{n+1} в общем случае каким-либо численным методом, а во втором случае y_{n+1} легко выразить в явном виде:

$$y_{n+1} = \frac{1}{\alpha_{n+1}} \left(\sum_{i=n-k}^{n+1} \beta_i x_i + f(x_n) - \sum_{i=n-k}^n \alpha_i y_i - g(y_n) \right). \quad (4.8)$$

Заметим в заключение, что моделирование элементов, описываемых интегральными зависимостями или уравнениями в частных производных, также может быть выполнено методом дискретизации и алгебраизации ОДУ.

4.4.3. Моделирование линейных инерционных элементов, заданных комплексным коэффициентом передачи $K(j\omega)$, в частотной области. Для этого нужно представить входной сигнал $X(j\omega)$, коэффициент передачи $K(j\omega)$ и выходной сигнал $Y(j\omega)$ в виде комплексных чисел в алгебраической или показательной форме. Для алгебраической формы, полагая $X(j\omega) = X_1(\omega) + jX_2(\omega)$, $K(j\omega) =$

$= K_1(\omega) + jK_2(\omega)$, $Y(j\omega) = Y_1(\omega) + jY_2(\omega)$ и учитывая, что $Y(j\omega) = K(j\omega)X(j\omega)$, получаем

$$Y_1(\omega) = X_1(\omega)K_1(\omega) - X_2(\omega)K_2(\omega),$$

$$Y_2(\omega) = X_1(\omega)K_2(\omega) + X_2(\omega)K_1(\omega).$$

Если используется показательная форма $X(j\omega) = X(\omega)e^{j\varphi_x(\omega)}$, $K(j\omega) = K(\omega)e^{j\varphi_k(\omega)}$, то $Y(j\omega) = X(\omega)K(\omega)e^{j(\varphi_x(\omega) + \varphi_k(\omega))}$. При этом

$$K(\omega) = (K_1^2(\omega) + K_2^2(\omega))^{1/2}; \quad \varphi_k(\omega) = \arctg(K_2(\omega)/K_1(\omega));$$

аналогично рассчитываются $X(\omega)$ и $\varphi_x(\omega)$.

Очевидно, фактически вместо $Y(j\omega)$ вычисляются

$$Y(\omega) = X(\omega)K(\omega) \text{ и } \varphi_y(\omega) = \varphi_x(\omega) + \varphi_k(\omega).$$

Переход от показательной формы к алгебраической выполняется по формулам

$$K_1(\omega) = K(\omega) \cos \varphi_k(\omega); \quad K_2(\omega) = K(\omega) \sin \varphi_k(\omega).$$

При задании вместо $K(j\omega)$ операторного коэффициента передачи $K(p)$ изложенная методика определения $Y(j\omega)$ остается справедливой, если положить $p = j\omega$ и, следовательно, $K(p) = K(j\omega)$.

Пример 4.2. Рассмотрим возможные способы моделирования линейного инерционного элемента, описываемого следующими эквивалентными способами:

$$1) h(t) = k(1 - e^{-t/T}); \quad 2) K(j\omega) = \frac{k}{1 + j\omega T}; \quad 3) T \frac{dy(t)}{dt} + y(t) = kx(t).$$

В первом случае

$$y(t_n) = \Delta t \sum_{k=0}^n x(t_k) k(1 - e^{-t_n - k/T}).$$

Во втором случае

$$\begin{aligned} K_1(\omega) &= k/(1 + \omega^2 T^2), \quad K_2(\omega) = -k\omega T/(1 + \omega^2 T^2), \quad Y(j\omega) = Y_1(\omega) + jY_2(\omega) = \\ &= [X_1(\omega)k/(1 + \omega^2 T^2) + X_2(\omega)k\omega T/(1 + \omega^2 T^2)] + j[-X_1(\omega)k\omega T/(1 + \omega^2 T^2) + \\ &\quad + X_2(\omega)k/(1 + \omega^2 T^2)]. \end{aligned}$$

В третьем случае, полагая

$$dy/dt|_{t=t_{n+1}} \approx (y_{n+1} - y_n)/\Delta t, \quad x(t) = x_{n+1}, \quad y(t) = y_{n+1},$$

получаем алгоритм расчета y_{n+1} неявным методом Эйлера

$$T(y_{n+1} - y_n)/\Delta t + y_{n+1} = kx_{n+1}.$$

Поскольку уравнение линейное, то вместо численного решения относительно y_{n+1} его можно решить аналитически:

$$y_{n+1} = (\Delta t k x_{n+1} + T y_n)/(T + \Delta t).$$

Если положить

$$dy/dt|_{t=t_n} \approx (y_{n+1} - y_n)/\Delta t, \quad y(t) = y_n, \quad x(t) = x_n,$$

то получим алгоритм расчета y_{n+1} явным методом Эйлера

$$y_{n+1} = y_n + \Delta t (kx_n - y_n) / T.$$

В данном случае дискретизации производной в точках t_{n+1} и t_n одинаковы, поэтому различие алгоритмов обусловлено лишь использованием значений $y(t)$, $x(t)$ для разных моментов времени. В общем случае (при использовании более чем двух точек) численные производные в точках t_{n+1} и t_n различны, например для четырех точек t_{n+1} , t_n , t_{n-1} , t_{n-2}

$$dy/dt|_{t=t_{n+1}} = (-2y_{n-2} + 9y_{n-1} - 18y_n + 11y_{n+1}) / 6\Delta t,$$

$$dy/dt|_{t=t_n} = (y_{n-2} - 6y_{n-1} + 3y_n + 2y_{n+1}) / 6\Delta t.$$

4.4.4. Моделирование элементов, заданных комплексным коэффициентом передачи $K(j\omega)$ или операторным коэффициентом передачи $K(p)$, во временной области [15, 16]. Пусть имеем входной непрерывный сигнал $x(t)$. При моделировании на ЭВМ $x(t)$ вычисляется лишь в отдельные моменты времени t_n . Можно считать, что непрерывный сигнал $x(t)$ заменяется дискретными значениями $x_n = x(t_n)$. В промежутках между дискретами, т. е. при $t_n < t < t_{n+1}$, значение сигнала не определено, поэтому его можно доопределить различными способами, например, полагая его постоянным на интервале $t_n - t_{n+1}$ и равным x_n . Тогда можно считать, что на вход инерционного элемента поступает не непрерывный сигнал, а последовательность скачков, и «непрерывный» элемент с коэффициентом передачи $K(p)$ и непрерывным входным сигналом $x(t)$ эквивалентен дискретному элементу с дискретным коэффициентом передачи $K^*(z)$ и дискретным сигналом x_n .

Если известен способ перехода от $K(p)$ к $K^*(z)$ и если K — дробнорациональная функция, т. е.

$$K^*(z) = \frac{a_0 + a_1 z + a_2 z^2 + \dots + a_k z^k}{1 + b_1 z + b_2 z^2 + \dots + b_m z^m}, \quad (4.9)$$

то можно показать, что выходной сигнал $y(t_n)$ связан с входным сигналом x_n рекуррентной формулой

$$y_n = \sum_{i=0}^k a_i x_{n-i} - \sum_{i=1}^m b_i y_{n-i}. \quad (4.10)$$

Эта формула позволяет найти новое значение y_n по известной старой m — предыстории (m точек) для y и k — предыстории для x .

Существуют различные частные способы перехода от $K(p)$ к $K^*(z)$, однако большинство из них требует значения полюсов функции $K(p) = A(p)/B(p)$, что в свою очередь требует решения уравнения $B(p) = 0$. Это не всегда легко сделать, поэтому рассмотрим один из наиболее универсальных способов определения $K^*(z)$, свободный от этого недостатка [15].

Пусть

$$K(p) = \frac{A_0 + A_1 p + \dots + A_r p^r}{B_0 + B_1 p + \dots + B_k p^k} \quad (4.11)$$

Разделив числитель и знаменатель на p^k , получим представление $K(p)$ совокупностью интегрирующих звеньев различного порядка:

$$K(p) = \frac{A_0/p^k + A_1/p^{k-1} + \dots + A_r/p^{k-r}}{B_0/p^k + B_1/p^{k-1} + \dots + B_k} \quad (4.12)$$

Поскольку все полюса звеньев равны нулю, то для каждого звена i -го порядка легко найти соответствие $K_i(p)$ и $K^*_i(z)$, используя один из упомянутых выше частных способов, например, метод Тастина, состоящий в следующем.

Для интегрирующего звена с коэффициентом передачи $K_1(p) = 1/p$ вход и выход связаны соотношением

$$y(t) = \int x(t) dt.$$

Используя общую формулу численного интегрирования

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} x(t) dt$$

и полагая в ней $x(t) \approx (x(t_{n+1}) + x(t_n))/2$, что соответствует интегрированию методом трапеций, получаем

$$y_{n+1} = y_n + \Delta t (x_{n+1} + x_n)/2.$$

Для этого рекуррентного уравнения на основе соответствия формул (4.10) и (4.9) можно написать выражение $K^*_1(z)$, соответствующее $K_1(p) = 1/p$:

$$K^*_1(z) = \frac{\Delta t}{2} \frac{1+z}{1-z}.$$

Поскольку $K_i(p) = 1/p^i = (1/p)^i = (K_1(p))^i$, то дискретный коэффициент передачи интегрирующего звена i -го порядка

$$K^*_i(z) = (K^*_1(z))^i = \left(\frac{\Delta t}{2} \frac{1+z}{1-z} \right)^i. \quad (4.13)$$

Заменяя в (4.12) $K_1(p)$ на $K^*_1(z)$, приводя получившееся выражение к виду (4.9) и переходя к (4.10), получаем рекуррентное выражение для вычисления сигналов y_n на выходе линейного инерционного элемента с коэффициентом передачи $K(p)$ достаточно общего вида (4.11).

Пример 4.3. Пусть элемент имеет коэффициент передачи $K(p) = (1 + pT)/p^2$. Разделив числитель и знаменатель на p^2 : $K(p) = 1/p^2 + T/p$. Используя (4.13), получаем

$$K^*(z) = \left(\frac{\Delta t}{2} \frac{1+z}{1-z} \right)^2 + T \frac{\Delta t}{2} \frac{1+z}{1-z} = \frac{a_0 + a_1 z + a_2 z^2}{1 + b_1 z + b_2 z^2},$$

где

$$a_0 = \frac{\Delta t^2 + \Delta t T}{4}; \quad a_1 = \frac{\Delta t^2}{4}; \quad a_2 = \frac{\Delta t^2 - T \Delta t}{4}; \quad b_1 = -2; \quad b_2 = 1.$$

Отсюда в соответствии с (4.10) следует рекуррентная формула для вычисления выходного сигнала в моменты времени t_n :

$$y_n = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} - b_1 y_{n-1} - b_2 y_{n-2}.$$

4.4.5. Моделирование узкополосных линейных инерционных элементов [15, 16]. К элементам этого типа относятся различные частотно-избирательные устройства — резонансные усилители, фильтры и т. д. Выделение этих устройств в особый класс, требующий специальных методов моделирования, вызвано спецификой их импульсной характеристики и входного сигнала.

Импульсная характеристика (реакция на δ -функцию) имеет вид (рис. 4.6)

$$h(t) = H(t) \cos(\omega_p t + \varphi_h(t)) = \operatorname{Re} \mathbf{H}(t) e^{j\omega_p t}, \quad (4.14)$$

где $\mathbf{H}(t) = H(t) e^{j\varphi_h(t)}$ — комплексная огибающая импульсной характеристики, частота изменения амплитуды $H(t)$ и фазы $\varphi_h(t)$ которой существенно меньше резонансной частоты ω_p .

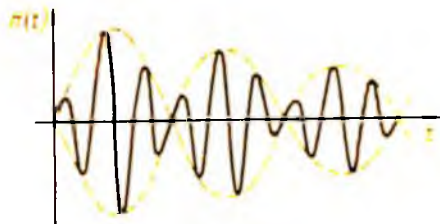


Рис. 4.6. Импульсная характеристика узкополосного линейного элемента

Входной сигнал обычно является высокочастотным модулированным колебанием на частоте $\omega_c = \omega_p + \Delta\omega$, где $\Delta\omega \ll \ll \omega_p$ — расстройка несущей частоты входного сигнала ω_c относительно резонансной частоты избирательного элемента ω_p . Модуляция сигнала по амплитуде или фазе (по закону $u = u(t)$) приводит к появлению

в сигнале комплексной огибающей, так что запись сигнала в комплексной форме имеет вид

$$\mathbf{X}(u, t) = S_1(u, t) e^{j\omega_c t} = S_1(u, t) e^{j\Delta\omega t} e^{j\omega_p t} = S_x(u, t) e^{j\omega_p t} e^{j\omega_p t}. \quad (4.15)$$

Здесь
$$S_x(u, t) = S_x(u, t) e^{j(\varphi_c(u, t) + \varphi_0 + \Omega t)} \quad (4.16)$$

— комплексная огибающая сигнала на входе; $S_x(u, t)$, $\varphi_c(u, t)$ — законы медленного изменения амплитуды и фазы огибающей входного сигнала; φ_0 — начальная фаза. Далее будем полагать $\varphi_x(u, t) = \varphi_c(u, t) + \varphi_0 + \Omega t$.

Прямой способ моделирования частотно-избирательных элементов состоит в вычислении мгновенных значений быстроосциллирующего (с частотой ω_c) выходного сигнала одним из рассмотренных выше способов, например, используя формулу обычной свертки для функции $h(t)$ и $x(t) = \operatorname{Re} \mathbf{X}(u, t)$. Однако при этом шаг дискретизации сигнала Δt придется выбирать соизмеримым с периодом частоты ω_c , что обычно приводит к огромному числу точек расчета, необходимому для воспроизведения медленного информационного процесса $u(t)$.

Поскольку вся информация об этом процессе заключена в комплексной огибающей, то для экономии времени моделирования целесообразно рассчитывать мгновенные значения не всего сигнала (4.15), а только его комплексной огибающей (4.16). Разработанный для этого метод называется *методом комплексных огибающих* и состоит в следующем. Если $S_x(t) = S(t)e^{j\Delta\omega t}$ — комплексная огибающая расстроенного на $\Delta\omega$ входного модулированного сигнала, а $H(t) = H(t)e^{j\varphi_h(t)}$ — комплексная импульсная характеристика элемента, то при $\Delta\omega \ll \omega_p$ комплексная огибающая сигнала на его выходе определяется приближенной формулой

$$S_y(t) = \frac{1}{2} \int_0^t H(\tau) S_x(t-\tau) d\tau, \quad (4.17)$$

являющейся комплексным аналогом формулы (4.3а).

Подставляя в (4.17)

$$H(\tau) = H_1(\tau) + jH_2(\tau), \quad S_x(t-\tau) = S_{x1}(t-\tau) + jS_{x2}(t-\tau)$$

и перемножая слагаемые, после преобразования произведений по методу скользящего суммирования (см. п. 4.3.3) можно получить формулы *дискретной комплексной свертки* для вычисления $S_y(t_n) = S_{y1,n} + jS_{y2,n}$:

$$\begin{aligned} S_{y1,n} = & \frac{\Delta t}{2} \left(\sum_{k=0}^n H_1(k \Delta t) S_{x1}((n-k) \Delta t) + \right. \\ & \left. - \sum_{k=0}^n H_2(k \Delta t) S_{x2}((n-k) \Delta t) \right), \\ S_{y2,n} = & \frac{\Delta t}{2} \left(\sum_{k=0}^n H_1(k \Delta t) S_{x2}((n-k) \Delta t) + \right. \\ & \left. + \sum_{k=0}^n H_2(k \Delta t) S_{x1}((n-k) \Delta t) \right). \end{aligned} \quad (4.18)$$

В общем случае дискретная комплексная свертка требует в четыре раза больше операций, чем вещественная. В частных случаях количество операций сокращается. Так, при только амплитудной модуляции, когда $\Delta\omega = 0$ и $\varphi_c(t) \equiv 0$, имеем $S_{x1} = S_x$, $S_{x2} = 0$ и в (4.18) остается только по одной сумме. Аналогично, если $\varphi_h(t) \equiv 0$, то $H = H_1$, $H_2 \equiv 0$ и тоже остается по одной сумме. При $\varphi_c(t) \equiv \varphi_h(t) \equiv 0$ имеем $S_{x2} = 0$, $H_2 = 0$ и (4.18) превращаются в (4.6) вещественной свертки.

Кроме комплексной свертки для расчета комплексной огибающей разработаны и другие, более быстрые способы. Среди них можно отметить способ рекуррентных комплексных разностных уравнений, являющийся комплексным аналогом рассмотренного ранее способа (4.9)—(4.10) для сигнала общего вида. В данном случае формула (4.9) описывает не полную, а укороченную дискретную передаточную функцию $K^*(z)$, соответствующую укороченной передаточной функции $K_y(p)$ узкополосного элемента. Функцию $K_y(p)$ можно получить, зная $K(j\Delta\omega)$, где $\Delta\omega = \omega_c - \omega_p$ и заменяя $j\Delta\omega$ на p .

Например, для усилителя промежуточной частоты (УПЧ) с n одиночными настроенными в резонанс контурами, каждый из которых имеет резонансный коэффициент усиления K_p и полосу пропускания $\Delta\omega_p$ на уровне 0,7, можно записать

$$K(j\Delta\omega) = \frac{K_p}{(1 + j\Delta\omega/\Delta\omega_p)^n}, \quad K(p) = \frac{K_p}{(1 + p/\Delta\omega_p)^n},$$

после чего перейти от $K(p)$ к $K^*(z)$ и далее получить рекуррентную разностную формулу по методике, описанной в п. 4.3.3 (формулы (4.11)–(4.13)) с учетом комплексного характера сигналов S_x , S_y и коэффициентов уравнения a_i , b_i . Детальное изложение приведенных выше и других способов моделирования узкополосных частотно-избирательных элементов дано в [15, 16].

4.5. ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ НЕЛИНЕЙНЫХ ИНЕРЦИОННЫХ ЭЛЕМЕНТОВ

Моделирование элементов этого типа возможно двумя способами. Первый способ состоит в описании элемента нелинейным дифференциальным уравнением вида (4.4) или (4.5), после чего это уравнение можно решить численными методами на основе дискретизации производных и алгебраизации получившегося уравнения (см. п. 4.4.2).

Однако этот способ может потребовать больших затрат времени, так как он связан с решением уравнений. Более быстрым является второй способ, основанный на представлении нелинейного инерционного элемента упрощенной функцией нелинейного преобразования или упрощенной функциональной схемой замещения. Ниже рассматриваются особенности реализации этого способа.

4.5.1. Моделирование нелинейных преобразований в радиочастотных устройствах. При моделировании нелинейных преобразований в радиочастотных устройствах — модуляторах, преобразователях частоты, ограничителях амплитуды, детекторах — возможны два подхода — идеальное и реальное моделирование.

Идеальное моделирование не учитывает особенностей реальных характеристик преобразующих устройств, например нелинейности характеристик в смесителях частоты, зависимости свойств детектора от амплитуды входного сигнала и т. д., вследствие чего все побочные эффекты, сопровождающие реальное преобразование (появление ненужных гармоник в преобразователях, снижение амплитуды огибающей в детекторах и др.) отбрасываются. Таким образом, идеальное моделирование преобразующего нелинейного элемента сводится к выполнению им заданной математической операции над входным сигналом, результат которой рассматривается как выходной сигнал.

Поясним это на примере моделирования идеального детектирования (рис. 4.7). Пусть на вход преддетекторного фильтра поступает высокочастотное модулированное колебание с известным законом модуляции, а значит, известно уравнение комплексной

огibaющей. Методом комплексной огibaющей (4.18) найдем мгновенные значения квадратурных составляющих $S_{y1,n}$ и $S_{y2,n}$ на выходе фильтра. Операция идеального детектирования тогда будет заключаться в вычислении мгновенных значений амплитуды или фазы огibaющей по известным значениям $S_{y1,n}$ и $S_{y2,n}$ по формулам 4 из табл. 4.2 типовых идеальных нелинейных преобразований в радиоустройствах. При этом сами процессы детектирования, протекающие в детекторе, не моделируются и в формулах свойства реального детектора не учитываются.

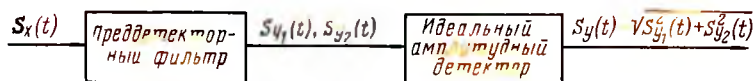


Рис. 4.7. Схема идеализированного функционального детектирования

Следует иметь в виду, что некоторые формулы численной реализации типовых нелинейных преобразований в радиочастотных цепях (1в, 4б, 4в) получены простейшими способами (интегрирование методом прямоугольников, двухточечная формула численного дифференцирования), поэтому они содержат вычислительную погрешность. Эта погрешность приводит к отклонениям от идеального моделирования, но ее можно уменьшить, применяя более точные численные методы.

Реальное функциональное моделирование нелинейных преобразований должно проводиться с учетом свойств преобразующего элемента и функциональная схема для моделирования строится на основе реальной структуры элемента. Например, если амплитудный детектор построен по схеме рис. 4.8,а, то его функциональную схему можно представить в виде, показанном на рис. 4.8,в, где f — характеристика диода, $K(p)$ — коэффициент передачи RC -цепи нагрузки.

Заметим, что структура функциональной схемы детектора не обязательно должна отражать структуру физической схемы, так как одной и той же функциональной схеме (рис. 4.8,в) могут соответствовать разные физические схемы (рис. 4.8,а,б). Необяза-

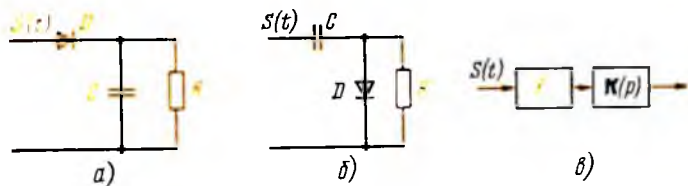


Рис. 4.8. К понятию реального функционального детектирования:
а, б — принципиальные схемы амплитудных детекторов; в — функциональная схема замещения

88 Таблица 4.2

Тип преобразования	Формула непрерывного преобразования	Формула численной реализации
1. Модуляция: а) амплитудная б) фазовая в) частотная	$x(t) = A(t) \sin(\omega_c t + \varphi_0)$ $x(t) = A_0 \sin(\omega_c t + \varphi_c(t) + \varphi_0)$ $x(t) = A_0 \sin(\omega_c t + \int_0^t \Omega(t) dt + \varphi_0)$	$x_n = A_n \sin(\omega_c t_n + \varphi_0)$ $x_n = A_0 \sin(\omega_c t_n + \varphi_{c,n} + \varphi_0)$ $x_n = A_0 \sin(\omega_c t_n + \Delta t \sum_{k=0}^n \Omega_k + \varphi_0)$
2. Преобразование частоты: а) идеальное б) реальное	$y(t, \omega_1 \pm \omega_2) = x(t, \omega_1)$ $y((\omega_1 \pm \omega_2)t) = f(x_1(\omega_1 t), x_2(\omega_2 t))$	$y_n(\omega_1 \pm \omega_2) = x_n(\omega_1)$ $y_n(\omega_1 \pm \omega_2) = f(x_{1,n}(\omega_1), x_{2,n}(\omega_2))$
3. Усиление с ограничением амплитуды на уровне A	$y(t) = \begin{cases} k(x(t)), & x < A; \\ A, & x \geq A \end{cases}$	$y_n = \begin{cases} k x_n, & x < A; \\ A, & x \geq A \end{cases}$
4. Детектирование: а) амплитудное б) фазовое в) частотное	$S_y(t) = \sqrt{S_{x1}^2(t) + S_{x2}^2(t)}$ $\varphi_y(t) = \arctg(S_{x2}(t)/S_{x1}(t))$ $\varphi_y(t) = \int_0^t \Omega(t) dt,$ $\Omega(t) = \frac{d}{dt} \arctg \frac{S_{x2}(t)}{S_{x1}(t)} =$ $= \frac{S_{x1}(t) \dot{S}_{x2}(t) - \dot{S}_{x1}(t) S_{x2}(t)}{S_{x1}^2(t) + S_{x2}^2(t)}$	$S_{y,n} = \sqrt{S_{x1,n}^2 + S_{x2,n}^2}$ $\varphi_y(t) = \arctg(S_{x2,n}/S_{x1,n})$ $\varphi_{y,n} = \Omega_n \Delta t + \varphi_{y,n-1},$ $\Omega_n = \frac{S_{x2,n} S_{x1,n-1} - S_{x1,n} S_{x2,n-1}}{\Delta t (S_{x1,n}^2 + S_{x2,n}^2)}$

тельность соответствия функциональной и физической схем — характерная черта функционального моделирования, целью которого является воспроизведение последовательности процессов преобразования сигнала, не всегда совпадающей с последовательностью расположения реальных физических элементов схемы.

В качестве примеров реального функционального моделирования различных нелинейных радиочастотных элементов представлением их в виде последовательного соединения безынерционного нелинейного преобразователя f входного сигнала x и инерционного линейного элемента с переходной характеристикой $h(t)$ могут служить структурные схемы и их математические описания, приведенные в табл. 4.3 [40]. Вычисление интегралов в табл. 4.3 можно выполнить методом скользящего суммирования (4.6).

4.5.2. Моделирование нелинейных преобразований в переключательных устройствах. Подобно радиочастотным устройствам переключательные схемы (ключи, релаксационные спусковые и сравнивающие схемы, триггеры), а также нелинейные аналоговые схемы (схемы операционных усилителей разных типов), точная математическая модель которых представляет нелинейное дифференциальное уравнение, можно моделировать на двух функциональных уровнях — идеальном и реальном.

Идеальное функциональное моделирование сводится к точному (до вычислительной погрешности) воспроизведению выполняемых элементом функций. При этом моделируемая схема заменяется безынерционным статическим или динамическим элементом, идеально (без учета задержки и нелинейных искажений, вносимых элементом) выполняющим заданное преобразование — усиление аналогового сигнала, формирование импульса заданной формы и т. д. Например, идеальное функциональное моделирование для операционного интегрирующего усилителя — это, по возможности, точное численное интегрирование входного сигнала, для ключевой логической схемы — воспроизведение скачка сигнала на выходе без задержки по отношению к скачку на входе, для мультивибратора — формирование последовательности идеально прямоугольных импульсов и т. д.

Реальное функциональное моделирование переключательных устройств должно проводиться с учетом таких свойств этих устройств, как формирование фронтов конечной длительности, задержка выходного сигнала по отношению к входному, неидеальная форма рабочей части импульса — вершины в прямоугольных импульсах, линейной части в пилообразных импульсах и т. д. При этом построение функциональной схемы замещения для реального моделирования нелинейного устройства может выполняться, как и для радиочастотных элементов, двумя способами.

Первый способ состоит в разбиении полного преобразования сигнала на отдельные элементарные преобразования и отображении каждого из них соответствующим функциональным элементом. В этом случае переключательный или импульсный элемент можно представить обычно последовательностью трех простых

Тип элемента	Схема замещения	Формула моделирования
Усилитель высокой частоты, регулируемый системой АРУ		$y(t) = K(u_{APY}) \int_0^{\infty} x(t-\tau) h^*(\tau) d\tau$
Усилитель низкой частоты		$y(t) = K \int_0^{\infty} x(t-\tau) h(\tau) d\tau$
Преобразователь частоты или синхронный детектор		$y(t) = K \int_0^{\infty} x(t-\tau) u_{\text{гет}}(t-\tau) h^*(\tau) d\tau$
Резонансный ограничитель амплитуды		$y(t) = \int_{-\infty}^t G[x(\tau)] h^*(t-\tau) d\tau$
Умножитель частоты (n — коэффициент умножения)		$y(t) = \int_{-\infty}^t G[x(\tau)] h^*(n)(t-\tau) d\tau$
Амплитудный детектор		$y(t) = \int_{-\infty}^t G[x(\tau)] h(t-\tau) d\tau$
Амплитудно-фазовый детектор		$z(t) = y_1(t) - y_2(t),$
Частотный детектор с расстроенными контурами		$y_i(t) = \int_{-\infty}^t G_i[x_i(\tau)] h(t-\tau) d\tau, \quad i = 1, 2,$ $x_1(t) = y(t) + x(t), \quad x_2(t) = y(t) - x(t)$ $z(t) = z_1(t) - z_2(t),$ $z_i(t) = \int_{-\infty}^t G_i[y_i(\tau)] h_{дi}(t-\tau) d\tau,$ $y_i(t) = \int_{-\infty}^t x(\tau) h_i(t-\tau) d\tau, \quad i = 1, 2$

Примечание: $h^*(t) = H(t) \cos(\omega_0 t + \varphi(t))$ — нормированная импульсная характеристика с огибающей $H(t)$.

элементов — элемента для моделирования изменения уровня сигнала от $x(t)$ до $x_1(t) < x(t)$ за счет конечного значения входного сопротивления $R_{вх}$, безынерционного элемента для моделирования функционального преобразования формы сигнала $y=f(x_1(t))$ и элемента чистой задержки τ для моделирования инерционных свойств (рис. 4.9,а). Таким образом, цепь элементарных преобразований имеет вид

$$x(t) \xrightarrow{1} x_1(t) \xrightarrow{2} f(x_1(t)) \xrightarrow{3} f(x_1(t-\tau)),$$

где цифры 1, 2, 3 обозначают номера простых элементов, реализующих эти преобразования.

Очевидно, способы определения отдельных параметров преобразования — уровня $x_1(t)$ (зависящего или независящего от $R_{вых}$ предыдущей схемы), формы сигнала $f(x_1(t))$ (идеализированной или реальной, постоянной или зависящей от формы $x_1(t)$), задержки τ (постоянной или зависящей от параметров формы сигнала, например, крутизны фронтов и т. д.) — могут быть самыми различными и порождать различные по точности функциональные схемы замещения. Очевидно также, что структура этой схемы не отражает структуры самого элемента.

Второй способ реального функционального моделирования сложного переключательного устройства состоит в разбиении его на более простые части, каждая из которых может быть заменена простым функциональным элементом. Структура получающейся функциональной схемы замещения отражает структуру самого устройства. Обычно разбиение по этому способу состоит в разделении устройства на три последовательные части, моделирующие соответственно инерционные входные цепи, функциональное преобразование в активных цепях и инерционные выходные цепи (рис. 4.9,б).

Таким образом, при реальном функциональном моделировании нелинейных радиочастотных, переключательных и других уст-

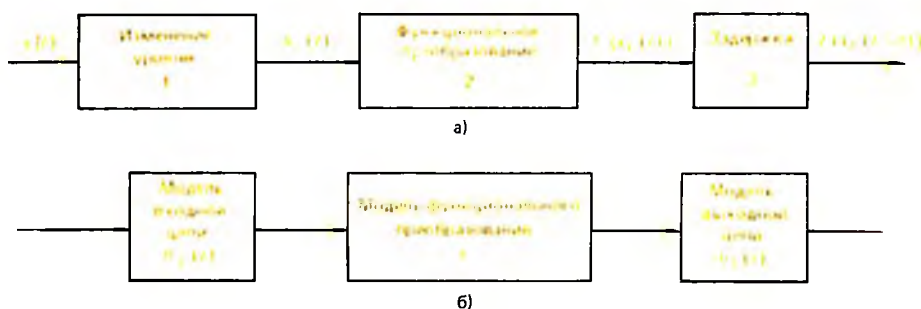


Рис. 4.9. Схемы функционального моделирования переключательных и аналоговых элементов:

а — функциональная модель преобразования сигнала; б — функциональная модель структуры элемента

роЙств необходимо для каждого преобразующего элемента строить свою функциональную схему, элементы которой отражают либо отдельные составляющие процесса преобразования — нелинейное безынерционное преобразование, линейную фильтрацию, задержки и т. д., либо физическую структуру преобразующего элемента, так что каждый элемент функциональной схемы выполняет такое же преобразование сигнала, как и соответствующий элемент физической структуры. Реальное функциональное моделирование указанными способами, особенно вторым, можно рассматривать как один из видов схмотехнического макро моделирования (см. гл. 11).

В заключение оценим вкратце области целесообразного применения идеального и реального функционального моделирования. Идеальная и реальная функциональные модели соотносятся между собой как макро модель и модель — макро модель быстрее реализуется, но имеет более низкую точность чем модель. В связи с этим идеальные функциональные модели следует применять в тех случаях, когда необходимо минимальное время моделирования, например при расчетах оптимальной структуры устройства на основе статистических методов или при многовариантных расчетах очень больших функциональных схем. Если же функциональная схема невелика и имеется достаточное время для ее моделирования, то целесообразно использовать реальные функциональные модели, позволяющие оценить погрешности преобразования сигнала и выявить требования к параметрам отдельных элементов.

4.6. ВЫЧИСЛИТЕЛЬНЫЕ АСПЕКТЫ МОДЕЛИРОВАНИЯ БАЗОВЫХ ЭЛЕМЕНТОВ

Применение тех или иных моделей базовых элементов налагает определенные ограничения на алгоритмы расчета как самих элементов, так и составленных из них функциональных схем. Среди этих ограничений можно выделить следующие.

4.6.1. Возможность вычисления производных. В некоторых численных методах расчета функциональных схем используются производные от преобразующих функций элементов. Численные методы, не использующие производных, называются методами нулевого порядка (методы простых итераций, Зайделя и др.); методы, использующие первые производные, называются методами первого порядка (метод Ньютона) и т. д.

Приведенные выше типы базовых элементов и способы их моделирования накладывают определенные ограничения на порядок численных методов, которые можно использовать для моделирования функциональных схем из этих элементов. Например, методы ненулевого порядка можно применять только для расчета функциональных схем из элементов типа аналог — аналог, допускающих расчет не только функции $y(t_n) = f(xt_n)$ для любого момента времени, но и ее производных. Отсюда вытекает требование непрерывности функции $y = f(x)$, которому не удовлетворяют, напри-

мер, элементы цифрового типа. К тому же для логических функций понятие производной вообще не определено. Кроме того, если даже $y=f(x)$ — непрерывная функция, но может быть вычислена только алгоритмически, например на основе разностных рекуррентных уравнений, то производные y' вычисляются только численно, через приращения, что вызывает дополнительные вычислительные сложности и затраты времени.

Таким образом, специфика типов переменных функциональных схем и алгоритмов расчета их элементов ограничивает применение численных методов первого-второго порядков и приводит к тому, что для моделирования функциональных схем применяются почти всегда только методы нулевого порядка, чаще всего метод простых итераций.

4.6.2. Возможность расчета временных диаграмм и переходных процессов с переменным шагом. Эта возможность ограничена при расчете функциональных схем с инерционными элементами и сигналами, рассчитываемыми на основе рекуррентных разностных соотношений, полученных в предположении постоянного шага вычислений. Устранение этого ограничения в принципе возможно, но требует разработки более универсальных рекуррентных соотношений, допускающих вычисление y_n на основе значений y_{n-1} , y_{n-2} , ..., вычисленных через неравные промежутки времени.

4.6.3. Устойчивость вычислений. Как известно, численные методы решения дифференциальных уравнений, основанные на рекуррентных соотношениях, могут быть неустойчивыми (расходятся при любом шаге Δt), относительно устойчивыми (расходятся при $\Delta t > T_{i\text{ мин}}$, где $T_{i\text{ мин}}$ — минимальная из постоянных времени элемента) и абсолютно устойчивыми (не расходятся при любых Δt). Например, явный метод Эйлера $x_{n+1} = x_n + \Delta t f(x_n, t_n)$ для решения дифференциального уравнения $\dot{x}(t) = f(x, t)$ относится к относительно устойчивым, а неявный метод Эйлера $x_{n+1} = x_n + \Delta t f(x_{n+1}, t_{n+1})$ — к абсолютно устойчивым, но требующим дополнительных итераций для вычисления x_{n+1} при каждом t_{n+1} .

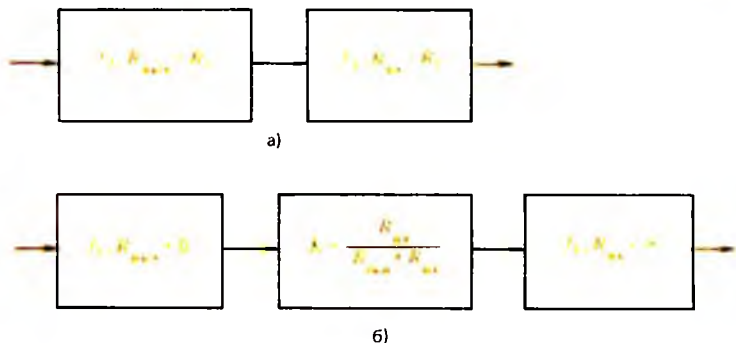


Рис. 4.10. Учет взаимного влияния элементов:
а — исходная функциональная схема; б — функциональная схема, учитывающая взаимное влияние элементов

Если общий алгоритм расчета функциональной схемы включает решение дифференциальных уравнений ее элементов, то этот алгоритм может расходиться при больших шагах Δt из-за расходимости алгоритмов расчета элементов. В связи с этим при использовании явного метода Эйлера для расчета элементов шаг Δt должен быть ограничен, а если это заметно увеличивает время расчета, нужно использовать неявный метод Эйлера. Применение более точных методов высокого порядка для функционального моделирования вряд ли целесообразно из-за невысокой точности моделей элементов.

4.6.4. Учет взаимного влияния элементов. Одна из основных погрешностей ФМ обязана отсутствию учета взаимного влияния элементов из-за допущения о развязке элементов. В ряде случаев (при больших $R_{\text{вых}}$ и малых $R_{\text{вх}}$ элементов) эта погрешность становится недопустимо большой. В этих случаях целесообразно вводить в состав функциональной схемы специальные согласующие элементы (рис. 4.10), позволяющие учесть изменения уровней сигналов из-за взаимного влияния элементов.

4.7. ПОСТРОЕНИЕ ФУНКЦИОНАЛЬНЫХ СХЕМ

Построение функциональных схем (ФС) сложных устройств проводится в общем случае в два этапа. На первом этапе каждый реальный элемент устройства представляется соответствующим элементом ФС. Для безынерционных и линейных инерционных элементов, выполняющих достаточно простые преобразования сигнала, а также для идеализированных функциональных элементов построение ФС на этом заканчивается. Если же элемент или выполняемое им преобразование сложны, то необходим второй этап построения ФС — этап детализации элемента или преобразования. На этом этапе детализируемый элемент разделяется на более мелкие образующие его элементы, а при детализации преобразования сигнала оно представляется последовательностью более простых преобразований.

Например, ФС супергетеродинного приемника (рис. 4.11,а) после первого этапа составления имеет вид, изображенный на рис. 4.11,б, предназначенный для идеализированного моделирования. Процесс моделирования состоит в том, что по высокочастотному сигналу $x(\omega_c)$ с известной огибающей рассчитывается огибающая сигнала на выходе усилителя высокой частоты с учетом его частотных свойств (методом комплексной огибающей или рекуррентных дискретных комплексных уравнений). Далее выполняется идеальное преобразование частоты путем ее сдвига на величину ω_r и снова рассчитывается огибающая S_x сигнала промежуточной частоты на выходе УПЧ. Детектирование также выполняется идеально и состоит в расчете параметров модулирующего сигнала (амплитуды, если детектор амплитудный, или фазы для фазового детектора) по рассчитанным квадратурным составляющим S_{x1} , S_{x2} на выходе УПЧ на основе формул из табл. 4.2.

Если это идеализированное моделирование не дает удовлетворительной точности, то отдельные наиболее сложные преобразования или элементы можно детализировать. Так, детализация преобразования частоты смесителем приводит к представлению процесса преобразования частоты функциональной схемой из трех

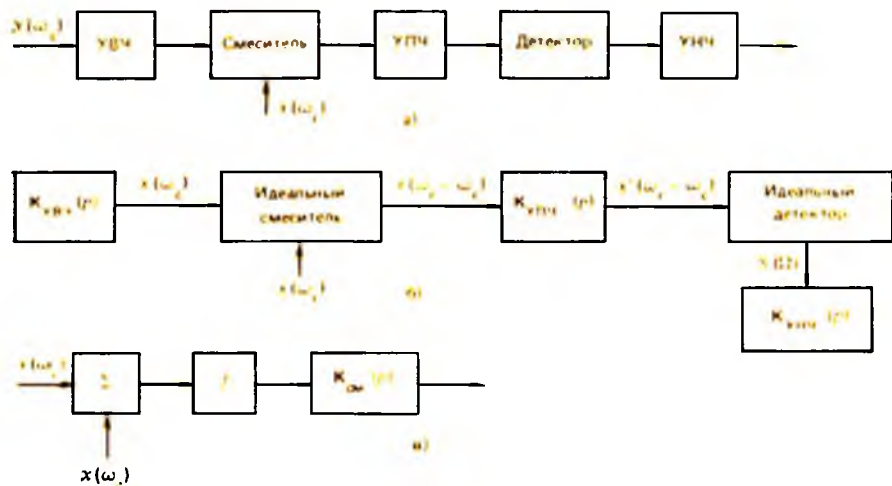


Рис. 4.11. Структурная схема супергетеродинного радиоприемника (а), идеализированная функциональная схема (б) и детализированная функциональная схема смесителя (в)

элементов на рис. 4.11,в, где f — характеристика нелинейного элемента смесителя, а $K_{см}(p)$ — элемент, характеризующий его частотные свойства. Можно считать эти три элемента функциональной схемой замещения смесителя, эквивалентной ему в смысле выполняемых преобразований (но не в смысле структуры). Аналогично можно детализировать идеальный детектор. Очевидно, моделирование приемника на основе детализированных схем смесителя и детектора даст другие, более точные сигналы на выходе детектора, учитывающие нелинейные и инерционные свойства элементов.

Рассмотрим примеры построения ФС.

1. Система автоматической регулировки усиления (АРУ) [17] (рис. 4.12) состоит из усилителя с регулируемым коэффициентом усиления, детектора и фильтра, преобразующего амплитуду низкочастотных колебаний в напряжение, управляющее усилением усилителя. При функциональном моделировании усилитель и детектор представляются каждый нелинейным безынерционным элементом и линейной инерционной частью.

Нелинейность f_1 характеризует зависимость $K_\omega = f(u_\phi)$ коэффициента усиления на частоте сигнала ω от управляющего напряжения на выходе фильтра. Нелинейность f_2 характеризует нелинейные свойства детектирующего диода.

выходе число $N = kt$, если на входе логическая переменная $x=1$); 3 — элемент цифрового типа (счетчик, преобразующий последовательный код N в параллельный двоичный N_x), 4 — элемент цифро-аналогового типа, преобразующий код N_x в аналоговую переменную $u_3 = f(N_x)$; 5 — пороговый элемент аналого-логического типа, преобразующий разность $u_x - u_3$ в логическую переменную Δ . В представленном виде ФС позволяет смоделировать статические временные диаграммы работы преобразователя, т. е. без учета инерционности его элементов.

Если скорость изменения сигнала u_x возрастет и станет соизмеримой со скоростью срабатывания отдельных элементов преобразователя, то моделирование по этой схеме не даст достоверных результатов. В этом случае необходимо моделировать преобразователь с учетом его инерционности, обусловленной конечным временем срабатывания элементов, т. е. в динамическом режиме. Для этого нужны более точные модели элементов. Например, в ФС замещения каждого элемента можно добавить конечную задержку — либо постоянную, либо зависящую от параметров входного сигнала. Так, задержку срабатывания триггера можно считать постоянной, а задержку образования кода N_x на выходе счетчика пропорциональной значению кода N на входе.

Приведенные примеры показывают, что, во-первых, построение ФС не является формальным процессом и требует от составителя достаточно высокой квалификации и, во-вторых, одно и то же устройство можно моделировать разными ФС замещения, обеспечивающими различную точность моделирования.

4.8. АЛГОРИТМЫ МОДЕЛИРОВАНИЯ ТИПОВЫХ СТРУКТУР ФУНКЦИОНАЛЬНЫХ СХЕМ

Большинство существующих ФС можно свести к типовым структурам (рис. 4.14). Рассмотрим основные алгоритмы моделирования этих структур.

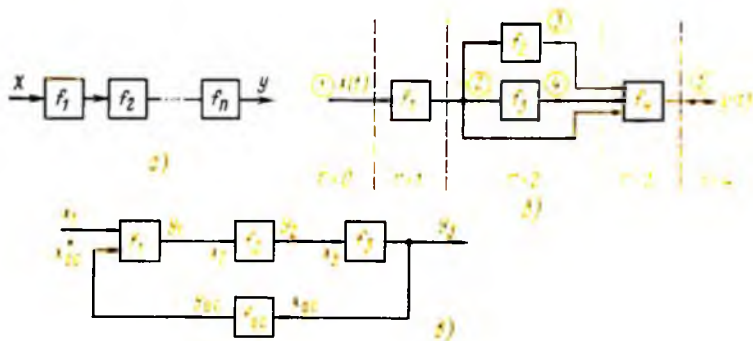


Рис. 4.14. Основные типы структур функциональных схем:
а — простая последовательная структура; б — сложная последовательная структура; в — структура с обратной связью

4.8.1. Общие подходы к моделированию функциональных схем. Можно выделить два общих подхода при построении алгоритмов моделирования ФС.

1. *Формальный подход* состоит в рассмотрении ФС как обычной ненаправленной структуры, описываемой уравнениями связей элементов (топологическими уравнениями) и уравнениями самих элементов (компонентными уравнениями). Далее к этим уравнениям можно применять любые алгоритмы решения систем уравнений соответствующего класса.

Формальность изложенного подхода заключается в том, что при построении алгоритма моделирования не учитываются особенности ФМ — развязки и однонаправленность элементов.

2. *Причинно-следственный или имитационный подход* к построению алгоритмов ФМ учитывает эти особенности и состоит в организации процесса моделирования так, чтобы он имитировал последовательное прохождение сигналов от одного элемента к другому в соответствии с принципом: сначала вычисляется причина — входной сигнал, а затем следствие — сигнал на выходе элемента.

Обязательным условием реализации этого подхода является ранжирование ФС. Ранжировать схему — значит расположить ее элементы так, чтобы входами элементов i -го ранга ($i=0, 1, 2, \dots$) были выходы элементов только меньшего ранга. Практически ранжирование обеспечивает соблюдение причинно-следственного принципа моделирования ФС: причина (сигнал на входе элемента) не должна вычисляться позже следствия (реакция на сигнал). В ранжированной структуре (рис. 4.14,б) входной сигнал $x(t)$ — элемент нулевого ранга ($r=0$), элемент f_1 имеет ранг $r=1$, для элементов f_2, f_3 имеем $r=2$, а для элемента f_4 — $r=3$.

Моделирование в ранжированной структуре заключается в последовательном вычислении сигнала сначала в элементах нулевого ранга (входных сигналах), затем во всех элементах первого ранга и т. д. Таким образом, сигнал последовательно проходит через слои элементов каждого ранга.

Если моделировать структуру на рис. 4.14,б без ранжирования, то сигнал в т. 2 нужно рассматривать как входной сразу для трех элементов f_2, f_3, f_4 , при этом сигнал в т. 5 от действия только сигнала в т. 2 может существенно отличаться от сигнала в т. 5 от одновременного действия сигналов в т. 2, 3, 4. Следовательно, элемент можно моделировать лишь после вычисления сигналов на всех его входах, что и обеспечивается автоматически ранжированием. Обычно операция ранжирования структуры выполняется в САПР программным путем на основе информации о связях элементов структуры. В дальнейшем все последовательные структуры функциональных схем будем считать ранжированными.

4.8.2. Алгоритмы расчета безынерционных функциональных схем. Наиболее просто рассчитываются ранжированные ФС с последовательной структурой без обратных связей (рис. 4.14,а). Расчет сводится к вычислению суперпозиции функций $f_1, \dots, f_n$:

$$y_n = f_n(f_{n-1}(\dots(f_2(f_1(x)))\dots)) \quad (4.19)$$

и является естественной реализацией причинно-следственного подхода к моделированию ФС.

Рассмотрим методику расчета выходного сигнала в более сложной структуре с обратной связью (рис. 4.14,в) на основе формального и причинно-следственного подходов. Математическая модель этой ФС при использовании формального подхода в общем случае включает две подсистемы уравнений: компонентную и топологическую. Компонентная подсистема состоит из уравнений отдельных элементов, а топологическая подсистема — из уравнений, описывающих связи между элементами. Для структуры на рис. 4.14,в, в которой для определенности положим $n=3$, математическая модель имеет следующий вид.

<i>Компонентная подсистема</i>	<i>Топологическая подсистема</i>	
1) $y_1 = f_1(x_1, x_{oc}^*)$,	5) $x_2 = y_1$,	
2) $y_2 = f_2(x_2)$,	6) $x_3 = y_2$,	
3) $y_3 = f_3(x_3)$,	7) $x_{oc} = y_3$,	(4.20)
4) $y_{oc} = f_{oc}(x_{oc})$,	8) $x_{oc}^* = y_{oc}$.	

Полученные уравнения после преобразования к неявной форме

1) $y_1 - f_1(x_1, x_{oc}^*) = 0$,	5) $x_2 - y_1 = 0$,	
2) $y_2 - f_2(x_2) = 0$,	6) $x_3 - y_2 = 0$,	(4.21)
3) $y_3 - f_3(x_3) = 0$,	7) $x_{oc} - y_3 = 0$,	
4) $y_{oc} - f_{oc}(x_{oc}) = 0$	8) $x_{oc}^* - y_{oc} = 0$	

можно решать методом Ньютона относительно восьми неизвестных

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\partial f_1}{\partial x_{oc}^*} \\
 0 & 1 & 0 & 0 & -\frac{\partial f_2}{\partial x_2} & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & -\frac{\partial f_3}{\partial x_3} & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & -\frac{\partial f_{oc}}{\partial x_{oc}} & 0 \\
 -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1
 \end{bmatrix}^{(k)} \begin{bmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta y_3 \\ \Delta y_{oc} \\ \Delta x_2 \\ \Delta x_3 \\ \Delta x_{oc} \\ \Delta x_{oc}^* \end{bmatrix}^{(k)}$$

$$\begin{bmatrix} y_1 - f_1(x_1, x_{oc}^*) \\ y_2 - f_2(x_2) \\ y_3 - f_3(x_3) \\ y_{oc} - f_{oc}(x_{oc}) \\ x_2 - y_1 \\ x_1 - y_2 \\ x_{oc} - y_3 \\ x_{oc} - y_{oc} \end{bmatrix}^{(k)} \quad (4.22)$$

где k — индекс итерации; $\Delta y_1^{(k)} = y_1^{(k+1)} - y_1^{(k)}$ и т. д. — поправки.

Число уравнений в (4.22) можно сократить вдвое, если в (4.20) подставить 5) — 8) в 1) — 4) и тем самым исключить очевидные топологические уравнения.

Основное преимущество формального подхода — возможность автоматизации функционального моделирования на основе поэлементного описания ФС без ее предварительного ранжирования. По существу, это описание ничем не отличается от описания электрической схемы, что позволяет использовать для расчета ФС хорошо известные программы схемотехнического проектирования при условии включения в их библиотеку моделей уравнений функциональных элементов.

Недостатком формального подхода является сложность его реализации при составлении программ расчета, а также ограничения на тип элементов, если используются алгоритмы первого-второго порядка — для этих элементов должно иметь смысл понятие производной. В связи с этим на основе метода Ньютона нельзя рассчитывать ФС, например, с переменными логического типа.

Более простыми являются алгоритмы расчета ФС, основанные на причинно-следственном подходе. Рассмотрим один из них. Структура уравнений системы (4.20) характерна тем, что в результате последовательной подстановки друг в друга компонентных и топологических уравнений легко получить одно уравнение относительно любой из переменных в петле обратной связи:

$$y_1 - f_1(x_1, f_{oc}(f_3(f_2(y_1)))) = 0, \quad (4.23a)$$

$$y_2 - f_2(f_1(x_1, f_{oc}(f_3(y_2)))) = 0, \quad (4.23б)$$

$$y_3 - f_3(f_2(f_1(x_1, f_{oc}(y_3)))) = 0, \quad (4.23в)$$

$$x_{oc}^* - f_{oc}(f_3(f_2(f_1(x_{oc}^*, x_1)))) = 0. \quad (4.23г)$$

Решив одно из этих уравнений, по найденному значению неизвестной y_1 , y_2 , y_3 или x_{oc}^* легко найти, используя (4.20), все остальные переменные.

Проще всего решать (4.23) методом простых итераций. Например, для (4.23a) можно записать

$$y_1^{(k+1)} = f_1(x_1, f_{oc}(f_3(f_2(y_1^{(k)})))) \quad (4.24)$$

Как известно, сходимость решения одиночного уравнения методом простых итераций можно всегда обеспечить, применяя неслож-

ные вспомогательные приемы, например, записывая (4.23а) в виде

$$y_1^{(k+1)} = y_1^{(k)} - \lambda (y_1^{(k)} - f_1(x_1, f_{oc}(f_3(f_2(y_1^{(k)})))) \quad (4.25)$$

и выбирая достаточно малое значение λ .

Таким образом, при функциональном моделировании на основе причинно-следственного подхода каждая петля обратной связи порождает одно уравнение вида $x=f(x)$. Если все элементы в петле обратной связи — линейные, т. е. $y_i=kx_i$, то, как известно, это уравнение можно решить аналитически, в общем же случае уравнение получается нелинейным и его нужно решать численно.

В ФС с n петлями обратной связи модель схемы будет содержать n уравнений вида $x=f(x)$, при этом они могут решаться либо последовательно независимо друг от друга, как, например, два уравнения вида $y_1^{(k+1)}=f_1(x, f_{oc1}(y_1^{(k)}))$, $y_2^{(k+1)}=f_2(y_1, f_{oc2}(y_2^{(k)}))$ для схемы на рис. 4.15,а, где y_1 во втором уравнении есть результат решения первого, либо как система связанных уравнений для схемы рис. 4.15,б

$$y_1^{(k+1, p)}=f_1(x, f_{oc2}(y_2^{(p)}), f_{oc1}(y_1^{(k)})), y_2^{(p+1)}=f_2(y_1^{(p)}). \quad (4.26)$$

Особенность решения этой системы состоит в том, что первое уравнение решается до сходимости (получения $y_1^{(p)}$) на каждой итерации (при каждом p) второго уравнения. Таким образом, получается алгоритм типа «цикл в цикле».

Если последовательность значений $y_1^{(p)}$ ($p=1, 2, \dots$) при $y_2^{(p)}$ как параметре сходится, то для итераций по y_2 также можно обеспечить сходимость.

Поскольку значение $y_2^{(p)}$ изменяется после каждой внешней итерации, то организация полного цикла внутренних итераций для каждого $y_2^{(p)}$ может заметно увеличить общее время расчета.

В связи с этим возможен более простой способ организации итераций, соответствующий или обычному методу простых итераций

$$y_1^{(k+1)}=f_1(x, f_{oc2}(y_2^{(k)}), f_{oc1}(y_1^{(k)})), y_2^{(k+1)}=f_2(y_1^{(k)}) \quad (4.27)$$

или методу Зайделя, когда второе уравнение имеет вид $y_2^{(k+1)}=f_2(y_1^{(k+1)})$. Однако в этих алгоритмах труднее обеспечить сходимость. Для этого каждое уравнение нужно приводить к виду (4.25); при этом сходимость всей системы (4.27) может оказаться чрезвычайно медленной (тысячи итераций при плохой обусловленности матрицы Якоби правых частей системы).

Из рис. 4.15,а, б видно, что независимые уравнения получаются в случае независимых петель обратной связи, т. е. не имеющих общих элементов, если же такие элементы есть, то уравнения оказываются связанными, при этом каждая внутренняя петля обратной связи порождает свой внутренний цикл итераций. Число связанных уравнений равно числу «вложенных» друг в друга или имеющих общие участки петель обратной связи.

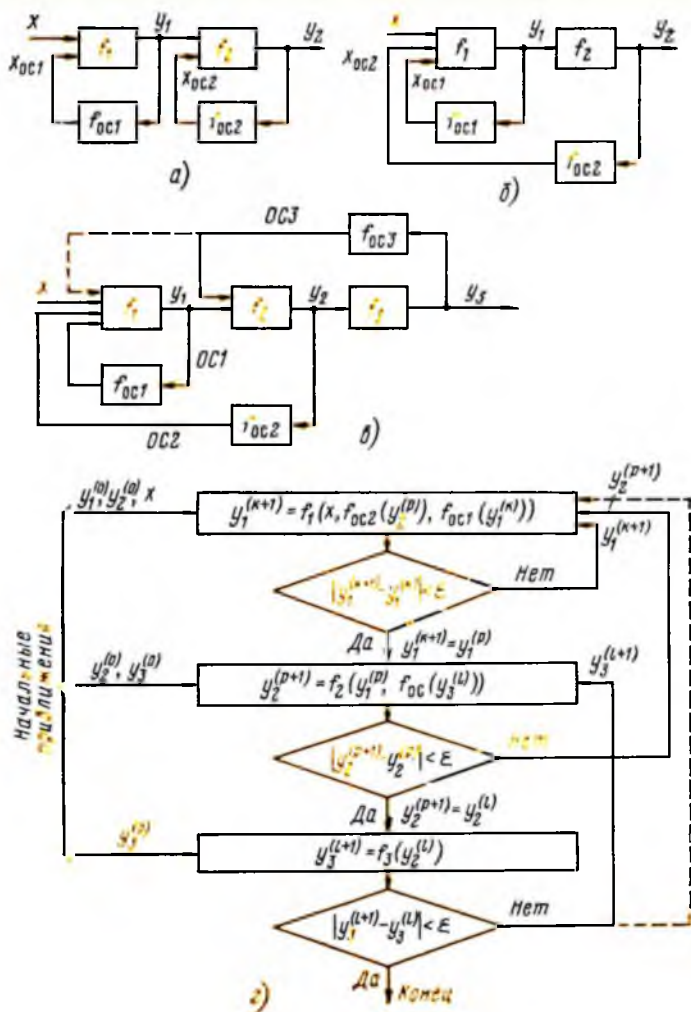


Рис. 4.15. Примеры структур функциональных схем с обратной связью (а, б, в) и алгоритм расчета структуры «в» (г)

Пример 4.4. Для функциональной схемы на рис. 4.15,в будем иметь три связанных уравнения:

$$\begin{aligned}
 y_1^{(k+1, p)} &= f_1(x, f_{oc2}(y_2^{(p)}), f_{oc1}(y_1^{(k)})), \\
 y_2^{(p+1, l)} &= f_2(y_1^{(p)}, f_{oc2}(y_2^{(l)})), \\
 y_3^{(l+1)} &= f_3(y_2^{(l)}).
 \end{aligned}
 \tag{4.28}$$

Алгоритм решения изображен на рис. 4.15,г. Как видно из рис. 4.15,в, структура имеет три петли обратной связи (назовем их ОС1, ОС2, ОС3).

При этом ОС1 вложена в ОС2, что дает двойной вложенный цикл — внутренний для ОС1 и внешний для ОС2, а ОС2 частично (через элемент f_2) вложена в ОС3, что дает второй двойной вложенный цикл — внутренний для ОС2 (с участием ОС1) и внешний для ОС3. Легко понять, что если бы петля ОС1 была как и ОС2 внутренней для ОС3, то получили бы алгоритм типа «тройной вложенный цикл», когда при невыполнении условия сходимости третьего уравнения выполнялось бы обращение не ко второму, а к первому уравнению. Этот случай показан на рис. 4.15, а, г, пунктиром.

Рассмотренная методика позволяет достаточно просто составлять вручную и рассчитывать на ЭВМ модели ФС, содержащих не слишком много элементов и петель обратной связи.

4.8.3. Расчет статических временных диаграмм. Пусть на вход ФС подан сигнал $x(t)$, изменяющийся настолько медленно, что инерционностью элементов ФС можно пренебречь и считать схему безынерционной. Получающиеся при этом зависимости $y(t)$ на выходе и во внутренних точках ФС назовем статическими временными диаграммами (СВД). Расчет СВД позволяет проверить правильность функционирования ФС и выявить ошибки структурного типа, вызванные, например, неверным соединением элементов ФС друг с другом.

Рассмотрим методику расчета СВД. Пусть сигнал $x(t)$ вычисляется в моменты времени $t_1, t_2, \dots$. Вычисление СВД может быть выполнено двумя способами.

Первый способ, который назовем *способом мгновенного сигнала*, состоит в расчете в каждый момент времени t_i значений $y_1(t_i), y_2(t_i), \dots, y_n(t_i)$ во всех точках ФС на основе одного из алгоритмов, рассмотренных в предыдущем разделе, после чего расчет повторяется для момента времени t_{i+1} и т. д.

Иногда более удобным является второй способ, который назовем *способом последовательного полного сигнала* (ППС). Этот способ, например, для схемы на рис. 4.14, а состоит в вычислении всех значений сигнала $y_1(t_1), y_1(t_2), \dots$ сначала на выходе элементов первого ранга, затем на их основе всех значений сигнала $y_2(t_1), y_2(t_2), \dots$ на выходе элементов второго ранга и т. д. Множество всех значений сигнала на входе или выходе элемента назовем полным сигналом.

Первый способ может быть реализован на основе как формального, так и причинно-следственного подхода, а второй способ — только на основе причинно-следственного подхода. В ФС с последовательной структурой расчет СВД более удобен способом ППС, позволяющим наглядно проследить преобразование сигнала по мере его прохождения по цепочке элементов, однако в ФС с обратными связями применение способа ППС имеет особенности, состоящие применительно к рис. 4.14, б в следующем:

1) сначала цепь обратной связи размыкается и полный сигнал рассчитывается на выходе разомкнутой цепи обратной связи (элемента 3 на рис. 4.1, в) при условии, что на входе ФС действует только полный сигнал x_1 ;

2) снова рассчитывается полный сигнал на выходе разомкнутой цепи обратной связи, но при условии, что на входе ФС действует не только полный сигнал x_1 , но и рассчитанный ранее полный сигнал обратной связи x_{oc}^* .

Описанный процесс повторяется до тех пор, пока не выполнится условие сходимости $\|x_j^{(i+1)}(t) - x_j^{(i)}(t)\| < \epsilon$, $j = 1, \dots, n$, где i — номер итерации; $x_j(t)$ — полный сигнал в j -й точке схемы; n — число точек. По существу, здесь реализуется метод простых итераций

$$x_{oc}^{(i+1)}(t) = f_{oc}(x_1(t), x_{oc}^{(i)}(t)), \quad (4.29)$$

но итерируются не отдельные значения, а вся форма сигнала. Вопросы сходимости для этого алгоритма, связанные с итерациями формы сигнала, остаются пока открытыми.

4.8.4. Алгоритмы расчета переходных процессов. Пусть в отличие от предыдущего случая в составе ФС имеются инерционные элементы (см. § 4.2), скорость реакции которых соизмерима со скоростью изменения входного сигнала, что вызывает дополнительные переходные процессы. Алгоритмы расчета сигнала во внутренних точках ФС и на ее выходе зависят от формы представления уравнений инерционных элементов и способов их решения. Рассмотрим наиболее типичные случаи.

1. *Инерционные элементы представлены дифференциальными уравнениями.* Алгоритм расчета ФС зависит от выбранного способа решения уравнений. Рассмотрим расчет переходных процессов в последовательной структуре без обратных связей.

Если выбран явный способ решения, то каждый инерционный элемент может быть представлен формулой типа (4.8), явно выражающей сигнал y_{n+1} на выходе элемента как функцию сигнала x_{n+1} и, возможно, предыдущих во времени известных значений $x_n, x_{n-1}, \dots, y_n, y_{n-1}, \dots$. Это представление соответствует уравнению безынерционного элемента, но справедливо только для одного момента времени t_{n+1} . Следовательно, в этом случае применимы все рассмотренные ранее алгоритмы расчета безынерционных ФС. Задавая на вход ФС сигнал $x(t)$ и рассчитывая выходной сигнал в моменты времени $t_1, t_2, \dots$ либо методом мгновенного сигнала, либо методом последовательного полного сигнала (см. п. 4.4.3), получаем диаграмму переходного процесса. Отличие от расчета безынерционных ФС состоит только в том, что нужно иметь начальные условия $y_i 0$ для решения дифференциального уравнения каждого элемента.

Пусть для численного решения дифференциальных уравнений элементов выбран неявный способ вида (4.7), который представлен в упрощенном виде

$$y_{i, n+i} = f_i(\alpha y_{i, n+1}, y_{i, n}, x_{n+1}), \quad (4.30)$$

где i — номер элемента; n — номер временного шага; α — коэффициент, зависящий от выбранного численного метода.

В этом случае реализация любого причинно-следственного алгоритма для ФС с последовательной структурой (рис. 4.14,а) будет заключаться не просто в последовательном вычислении функций $y_{1,n+1}=f(x_n)$, $y_{2,n+1}=f(y_{1,n})$ и т. д., как при использовании явных методов, а в последовательном решении уравнений вида (4.30) для каждого элемента либо при фиксированном n и разных $i=1, 2, \dots$, после чего переходим опять к первому элементу и следующему моменту времени (способ мгновенного сигнала), либо при фиксированном i , начиная с $i=1$, но разных $n=1, 2, \dots$, после чего переходим к следующему элементу $i=2$ и снова $n=1, 2, \dots$ и т. д. (способ ППС).

Легко заметить, что неявной формуле (4.30) решения дифференциального уравнения можно поставить в соответствие структуру с обратной связью (рис. 4.16,а—в), в которой элемент обратной связи α характеризует степень неявности метода. Однако в отличие от структурной обратной связи, являющейся неотъемле-

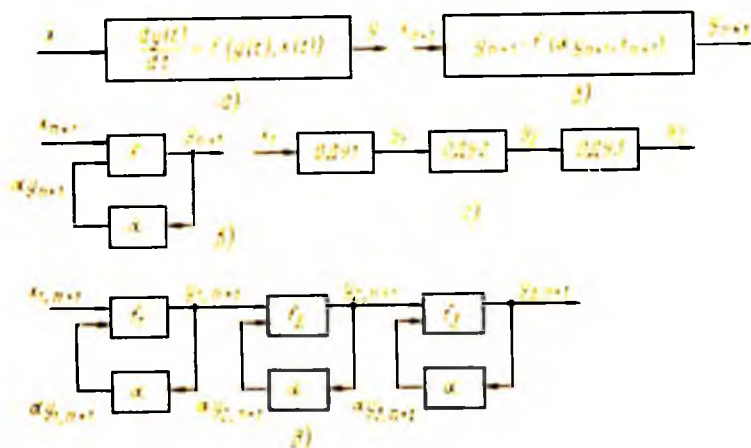


Рис. 4.16. К расчету инерционных элементов:

а — инерционный элемент (ИЭ); б — дискретизированное представление ИЭ для неявного метода; в — безынерционная структура, соответствующая расчету ИЭ неявным методом; г, д — последовательная инерционная функциональная структура и ее эквивалентное представление безынерционной структурой с местными обратными связями.

мой частью ФС, данная обратная связь обязана лишь принятому методу и может быть названа вычислительной. Изменяя величину α , можно регулировать степень устойчивости численного метода решения, от которой зависит допустимый шаг интегрирования дифференциального уравнения. Как известно, для явных методов этот шаг всегда ограничен числовой неустойчивостью решения, а для неявных методов это ограничение либо совсем отсутствует, как, например, для неявного метода Эйлера $y_{n+1}=y_n+h f(y_{n+1}, x_{n+1})$, либо проявляется лишь при очень больших шагах h .

Таким образом, с учетом вышеизложенного расчет сигналов в последовательной цепочке инерционных элементов (рис. 4.16,а) каким-либо неявным методом решения дифференциальных уравнений этих элементов в одной точке t_{n+1} эквивалентен расчету безынерционной структуры с обратными связями (рис. 4.16,б). Алгоритмы расчета таких структур уже рассматривались в п. 4.3.2.

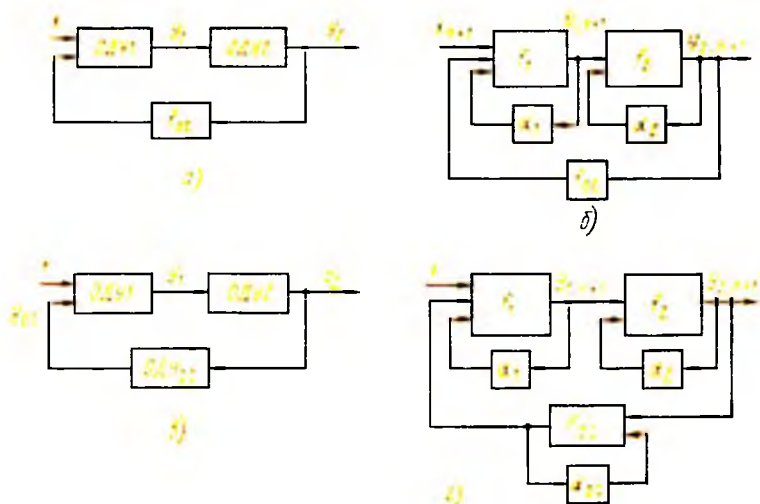


Рис. 4.17. Типовые структуры инерционных функциональных схем с безынерционной (а) и инерционной (б) обратной связью и эквивалентное представление схем безынерционными структурами (б, в)

Рассмотрим теперь алгоритмы расчета инерционных ФС с обратными связями двух типов — безынерционными (рис. 4.17,а) и инерционными (рис. 4.17,б). С учетом соответствия рисунков 4.16,а и в инерционные структуры на рис. 4.17,а, в после дискретизации дифференциальных уравнений можно заменить безынерционными структурами (рис. 4.17,б, в), где участки f_i с обратными связями α_i соответствуют решению дифференциальных уравнений инерционных элементов неявными итерационными методами.

Для схемы на рис. 4.17,б уравнения причинно-следственного типа для расчета $y_{1,n+1}$, $y_{2,n+1}$ в каждый момент t_{n+1} имеют вид

$$y_{1,n+1} = f_1(\alpha_1 y_{1,n+1}, y_{oc,n+1}, x_{n+1}),$$

$$y_{2,n+1} = f_2(y_{1,n+1}, \alpha_2 y_{2,n+1}).$$

Для схемы на рис. 4.17,в расчетные уравнения имеют вид

$$y_{1,n+1} = f_1(\alpha_1 y_{1,n+1}, y_{oc,n+1}, x_{n+1}),$$

$$y_{2,n+1} = f_2(y_{1,n+1}, \alpha_2 y_{2,n+1}),$$

$$y_{oc,n+1} = f_{oc}(y_{2,n+1}, \alpha_{oc} y_{oc,n+1}).$$

Полученные уравнения можно итерировать различными способами: методом простых итераций, методом Зайделя и т. д., что порождает, например, для схемы на рис. 4.17,б следующие алгоритмы:

Простые итерации:

$$\begin{aligned} y_{1,n+1}^{(k+1)} &= f_1(\alpha_1 y_{1,n+1}^{(k)}, f_{oc}(y_{2,n+1}^{(k)}), x_{n+1}), \\ y_{2,n+1}^{(k+1)} &= f_2(y_{1,n+1}^{(k)}, \alpha_2 y_{2,n+1}^{(k)}). \end{aligned}$$

Итерации по Зайделю:

$$\begin{aligned} y_{1,n+1}^{(k+1)} &= f_1(\alpha_1 y_{1,n+1}^{(k)}, f_{oc}(y_{2,n+1}^{(k)}), x_{n+1}), \\ y_{2,n+1}^{(k+1)} &= f_2(y_{1,n+1}^{(k+1)}, \alpha_2 y_{2,n+1}^{(k)}). \end{aligned}$$

Итерации с последовательной сходимостью:

$$\begin{aligned} y_{1,n+1}^{(k+1,p)} &= f_1(\alpha_1 y_{1,n+1}^{(k,p)}, f_{oc}(y_{2,n+1}^{(p)}), x_{n+1}), \\ y_{2,n+1}^{(k+1,p)} &= f_2(y_{1,n+1}^{(p)}, \alpha_2 y_{2,n+1}^{(p)}). \end{aligned}$$

В последнем алгоритме первое уравнение решается до сходимости на каждой итерации второго уравнения. Поскольку уравнения, как правило, нелинейные, сходимость их численного решения устанавливается обычно опытным путем в процессе расчета. Для упрощения расчета переходных процессов в ФС с обратными связями часто используют прием, называемый задержкой на один шаг или на один такт моделирования. В основе этого приема лежит допущение, что $y_{n+1} \approx y_n$, тем более верное, чем меньше временной шаг или такт. Это допущение позволяет избежать итерации по индексу k на каждом $(n+1)$ -м шаге, поскольку все величины, в том числе сигнал обратной связи, необходимые для расчета y_{n+1} , известны из предыдущего n -го шага.

Таким образом, обратная связь здесь оказывается запаздывающей на один шаг, что вносит определенные погрешности в результат, но упрощает расчет.

Пример 4.5. Для схемы на рис. 4.17,а будем использовать явные методы для расчета дифференциальных уравнений элементов и задержку на один шаг для расчета всей ФС. Тогда получим следующий алгоритм, не требующий решения уравнений на каждом шаге:

$$y_{1,n+1} = f_1(y_{1,n}, y_{oc,n}, x_n), \quad y_{2,n+1} = f_2(y_{2,n}, y_{1,n}).$$

Рассмотренные выше способы составления и решения уравнений операционных ФС основаны на причинно-следственном подходе, т. е. учитывают, что вход одного элемента одновременно является выходом другого. Таким образом, в этих уравнениях учтены связи элементов, т. е. топологические уравнения ФС. Учет этих связей выполнен неформальными методами, умозрительно, поэтому изложенную методику составления уравнений трудно автоматизировать. Кроме того, она применима вследствие своей неформальности лишь к ФС с достаточно простой структурой. Если же

структура сложная, с большим числом обратных связей, то расчет ФС целесообразно выполнять формальным методом, составляя подсистему топологических уравнений для связей элементов типа $x_i - y_j = 0$, где i, j — номера элементов, и подсистему компонентных уравнений, которые для инерционных элементов, заданных дифференциальными уравнениями, можно представить в неявном виде

$$F(dy/dt, y, dx/dt, x) = 0$$

или после дискретизации производных

$$F(y_{n+1}, y_n, \dots, x_{n+1}, x_n, \dots) = 0.$$

Общая система топологических и компонентных уравнений для каждого момента t_{n+1} имеет вид

$$\Phi(y_{n+1}, x_{n+1}) = 0$$

и содержит p компонентных уравнений (p — число элементов) и r топологических уравнений (r — число соединений элементов).

Далее эту систему целесообразно решать методом Ньютона, для которого обычно существует область сходимости в окрестности каждой точки y_{n+1}, x_{n+1} . Решение начинается при нулевых условиях y_0, x_0 и повторяется в каждой следующей точке y_{n+1}, x_{n+1} ($n=0, 1, \dots$).

2. *Инерционные элементы линейны и представлены переходными характеристиками $h_i(t)$.* Расчет переходных процессов в этом случае возможен только на основе причинно-следственного подхода, не требующего вычисления производных.

Алгоритмы расчета ФС без обратных связей очевидны и сводятся к последовательному вычислению мгновенного или полного сигнала на выходе каждого инерционного элемента в структуре рис. 4.14,а на основе формул типа (4.6). Если же ФС имеет обратные связи, то это приводит к итеративному алгоритму либо типа (4.18), если итерируется полный сигнал, либо обычного типа с итерациями до сходимости для каждого мгновенного значения сигнала. В обоих случаях алгоритмы расчета переходных процессов практически не отличаются от алгоритмов расчета статических временных диаграмм, за исключением вида компонентных уравнений.

3. *Инерционные элементы линейны и представлены комплексными коэффициентами передачи $K_i(j\omega)$ или $K_i(p)$.* Для ФС с последовательной структурой без обратных связей (рис. 4.14,а) результирующий коэффициент передачи

$$K_{\text{общ}}(p) = \prod_{i=1}^n K_i(p),$$

n — число последовательно включенных элементов.

Если ФС имеет цепь обратной связи с элементом $K_{\text{ос}}(p)$, охватывающую n последовательных элементов (рис. 4.14,б), то

$$K_{\text{общ}}(p) = \prod_{i=1}^n K_i(p) / (1 + K_{\text{ос}}(p) \prod_{i=1}^n K_i(p)).$$

Для ФС с n параллельно включенными элементами $K_i(p)$ общий коэффициент передачи для суммарного сигнала на выходе

$$K_{\text{общ}}(p) = \sum_{i=1}^n K_i(p).$$

Практически при моделировании на ЭВМ вместо оператора p , являющегося просто символом, неудобным для расчетов на ЭВМ, используют комплексную частоту $j\omega$, которой можно задавать различные численные значения ω .

Интересно отметить, что принцип развязки элементов, используемый в функциональном моделировании, позволяет применять в одной ФС разные способы представления ее элементов и моделировать их в соответствии с этими способами. Например, ФС может состоять из трех элементов, первый из которых представлен дифференциальным уравнением, второй — переходной характеристикой $h(t)$, а третий — комплексным коэффициентом передачи $K(j\omega)$. Моделирование преобразований сигнала $x(t)$ в этой цепочке можно выполнить следующим образом: численно решая дифференциальное уравнение первого элемента, найти полный сигнал на его выходе, выраженный множеством дискретных значений $y_i = y(t_i)$, $i = \overline{1, n}$, затем на основе формулы дискретной свертки (4.6) вычислить полный сигнал на выходе второго элемента, после чего можно рассчитать сигнал на выходе цепочки $y_{\text{вых}}$ на основе рекуррентных уравнений типа (4.10).

4.9. ПРОГРАММЫ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ

Характерной особенностью функционального моделирования является, с одной стороны, простота алгоритмов функционирования ФС, так как обычно выход одного элемента служит входом другого, а с другой стороны, большое разнообразие моделей функциональных элементов и способов их представления. Вследствие этой особенности наиболее рациональной формой для автоматизации ФМ следует считать программирование на языке высокого уровня типа Фортран или ПЛ/1 алгоритма функционирования ФС. Создание универсальных программ, автоматизирующих составление и решение уравнений ФС любой структуры и с любыми моделями элементов весьма сложно и трудоемко. Для простых ФС их уравнения и алгоритмы решения могут быть легко составлены на основе причинно-следственного подхода самим проектировщиком. В этом случае можно рекомендовать для написания программ моделирования отдельных ФС следующую методику.

1. Составляются модели для каждого элемента ФС и программируются в виде подпрограмм-функций, например, вида $F1(A, B, C, X1)$, $F2(D, E, F, G, X2)$ и т. д., где $F1$, $F2$ — имена преобразующих функций элементов 1, 2; $A, B, \dots$ — формальные параметры процедур, соответствующие параметрам элементов, $X1$, $X2$ —

формальные параметры, соответствующие входным сигналам элементов i , 2 и т. д.

2. Пишется моделирующая программа, в которой используются составленные подпрограммы элементов.

Таким образом, упрощенная структура программы, моделирующей, например, ФС из двух последовательных элементов, будет иметь вид

PROGRAM S	}	Программа моделирования ФС
X1 = ..		
Y1 = F1 (A, B, C, X1)		
X2 = Y1		
Y2 = F2 (D, E, F, G, X2)	}	Подпрограмма моделирования первого элемента
FUNCTION F1 (A, B, C, X1)		
END		
FUNCTION F2 (D, E, F, G, X2)	}	Подпрограмма моделирования второго элемента
END		
END		

Задавая различные значения входному сигналу $X1$, например, с помощью операторов DO и IF, получаем группу значений выходного сигнала $Y2$.

Основной сложностью моделирования ФС с обратными связями на основе причинно-следственного подхода является обеспечение сходимости итерационного решения систем уравнений.

Другой подход к моделированию ФС основан на использовании уже разработанных и получивших широкое признание программ автоматизации схемотехнического проектирования (АСХП) (см. гл. 7). Для этого каждый функциональный элемент ФС представляется электрической схемой замещения, включающей управ-

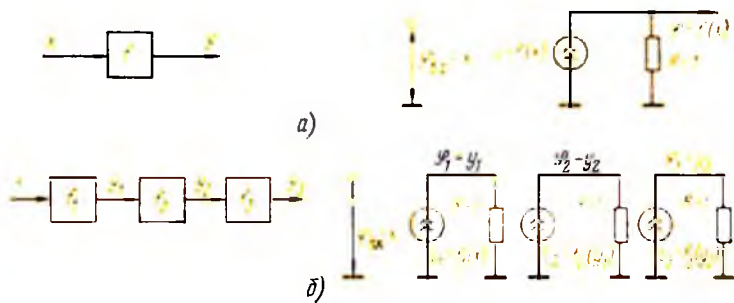


Рис. 4.18. Электрические схемы замещения функциональных элементов:

а — для одиночного элемента; б — для цепочки элементов

ляемые источники тока или напряжения. Вид схемы замещения и тип источников зависит от типа независимых переменных (базиса) программы.

Рассмотрим эти схемы замещения применительно к наиболее распространенному в современных программах АСХП базису узловых потенциалов. Моделирование безынерционного функционального элемента $y=f(x)$, имеющего $R_{вх}=\infty$, $R_{вых}=0$, выполняется с помощью схемы замещения на рис. 4.18,а, а цепочки элементов — с помощью схемы рис. 4.18,б. Выбор в этих схемах $R=1$ приводит к численному равенству потенциалов и токов $\varphi_i=f_i(\varphi_{i-1})$, и цепочка фактически выполняет преобразование входного потенциала $\varphi_{вх}=x$:

$$\varphi_1=f_1(\varphi_{вх}), \varphi_2=f_2(f_1(\varphi_{вх})), \varphi_3=f_3(f_2(f_1(\varphi_{вх}))).$$

Обычно в программах АСХП реализуется метод Ньютона, поэтому для цепочки на рис. 4.18,б этим методом в базисе узловых потенциалов будет решаться система трех уравнений с тремя неизвестными y_1, y_2, y_3 :

$$i_1 - \varphi_1/R = f_1(x) - y_1 = 0,$$

$$i_2 - \varphi_2/R = f_2(y_1) - y_2 = 0,$$

$$i_3 - \varphi_3/R = f_3(y_2) - y_3 = 0.$$

Моделирование инерционных функциональных элементов во временной области выполняется путем дискретизации и алгебраизации их уравнений, после чего полученное уравнение можно представить такой же схемой замещения, как и на рис. 4.18,а.

В качестве примера рассмотрим класс инерционных элементов ФС, описываемых дифференциальным уравнением вида [17]

$$a_1 \frac{dy}{dt} + a_0 y = b_1 \frac{dx}{dt} + b_0 x. \quad (4.31)$$

Разным значениям коэффициентов соответствуют различные типовые линейные звенья систем автоматического управления (САУ) — дифференцирующее ($a_1=0, b_0=0$); интегрирующее ($a_0=0, b_1=0$) и др. Дискретизация уравнения (4.31) позволяет привести его к виду $y_{n+1}=f(y_n, x_{n+1}, x_n)$, в котором y_n, x_{n+1}, x_n — известны, т. е. получаем уравнение вида $y=f(x)$, схема замещения которого уже рассматривалась.

Например, дискретизация (4.31) для явного метода Эйлера дает уравнение

$$a_1 \frac{y_{n+1} - y_n}{\Delta t} + a_0 y_n = b_1 \frac{x_{n+1} - x_n}{\Delta t} + b_0 x_n,$$

откуда получаем обобщенное уравнение типового инерционного звена

$$y_{n+1} = \frac{\Delta t}{a_1} \left(b_1 \frac{x_{n+1} - x_n}{\Delta t} + b_0 x_n + \left(\frac{a_1}{\Delta t} + a_0 \right) y_n \right).$$

Для абсолютно устойчивого неявного метода Эйлера дискретизация дает уравнение

$$a_1 \frac{y_{n+1} - y_n}{\Delta t} + a_0 y_{n+1} = b_1 \frac{x_{n+1} - x_n}{\Delta t} + b_0 x_{n+1}.$$

откуда получаем обобщенное дискретизированное уравнение звена

$$y_{n+1} = \left(b_1 \frac{x_{n+1} - x_n}{\Delta t} + b_0 x_{n+1} + \frac{a_1}{\Delta t} y_n \right) / \left(\frac{a_1}{\Delta t} + a_0 \right).$$

Очевидно, для повышения точности численного решения можно применять более сложные формулы дискретизации, например формулы дифференцирования назад (ФДН) [32].

Изложенные приемы расширяют область применения весьма трудоемких в разработке программ АСХП, позволяя рассчитывать с их помощью не только электрические, но и функциональные схемы различных систем управления и регулирования.

ГЛАВА 5

МОДЕЛИРОВАНИЕ НА РЕГИСТРОВОМ УРОВНЕ

5.1. ПОСТАНОВКА ЗАДАЧИ

В предыдущей главе рассматривалось функциональное моделирование РЭУ общего назначения. Среди этих устройств особое место как по степени важности и распространения, так и по способу функционирования и математическому аппарату моделирования занимает электронно-вычислительная аппаратура (ЭВА).

Разработка структурной схемы ЭВА выполняется общими способами, описанными в гл. 3. После выбора структурной схемы ЭВА переходят к более детальной разработке входящих в нее устройств, блоков, их алгоритмов работы и взаимодействия. Устройства при этом рассматриваются в виде совокупности регистров, счетчиков, блоков памяти, триггеров управления и т. п., а преобразование информации — на уровне машинных слов, соответствующих разрядности регистров, запоминающего и арифметического устройств. Внутренняя структура каждого из элементов — регистров, счетчиков, сумматоров — на данном этапе не детализируется. Схема устройства, составленная из указанных элементов, называется функциональной.

В настоящее время составление функциональных схем производится неавтоматизированными методами на основании накопленного опыта высококвалифицированными специалистами.

Кроме основной задачи — разработки структуры устройств ЭВА на уровне регистров, на этапе функционального проектирования решаются задачи разработки алгоритмов выполнения отдельных команд на уровне простейших операций преобразования и передачи информации, разработки алгоритмов обмена информацией между устройствами и т. п.

Естественной составной частью функционального проектирования является анализ разработанной функциональной схемы, ко-

торый обычно выполняется путем моделирования схемы на ЭВМ. Подобное моделирование часто называется моделированием на уровне регистровых передач.

5.2. ОСНОВНЫЕ ПРИНЦИПЫ МОДЕЛИРОВАНИЯ НА УРОВНЕ РЕГИСТРОВЫХ ПЕРЕДАЧ

При моделировании устройств на уровне регистровых передач спектр используемых элементов и их моделей достаточно широк. Основными элементами являются регистры, счетчики, сумматоры и т. п., однако используются и более сложные — устройства оперативной памяти, микропрограммной памяти и более простые — триггеры и логические элементы, реализующие функции И, ИЛИ, НЕ и т. п. Разнообразие элементов позволяет в зависимости от конкретных требований к процессу проектирования путем моделирования решать задачи проверки правильности функционирования устройств в целом и отладки программ, а также отдельных частей устройств на логическом уровне.

Модели элементов представляются выполняемыми ими функциями без раскрытия внутренней структуры, времен срабатывания, задержек и т. д. Токи, напряжения, пороги срабатывания устройств, нагрузочные способности не учитываются при моделировании на регистровом уровне. Например, моделью триггера является элементарное устройство, хранящее один бит информации, моделью n -разрядного регистра — устройство, хранящее n бит информации. Типы триггеров, составляющих регистр, и способ построения регистра не принимают во внимание.

Сигналы моделируются строками, состоящими из нулей и единиц. Эти строки соответствуют состоянию регистров, счетчиков, шин передачи информации.

При построении модели считается, что объект состоит из двух частей: операционного устройства и устройства управления (рис. 5.1), работа которых происходит по тактам. Тактирующие импульсы (синхроимпульсы) вырабатывает генератор синхросигналов. В



Рис. 5.1. Структура моделируемого устройства

каждом такте работы устройство управления формирует сигналы, инициирующие выполнение некоторых действий в операционном устройстве. Как правило, ЭВА работает по программе, хранящейся в памяти, и выполняемые в каждый конкретный момент времени действия определяются текущей командой программы.

Операционное устройство реализует элементарные действия по передаче, запоминанию и преобразованию информации, такие как перепись информации из одного регистра в другой, чтение и запись информации в оперативную память, установка заданных состояний триггеров управления, сдвиг информации в регистре на

один или несколько разрядов и т. п. Такие действия принято называть микрооперациями. Обычно в ЭВА каждой микрооперации соответствует собственный инициирующий сигнал.

Реализация каждой команды программы осуществляется за несколько синхротактов, в течение которых выполняется требуемая совокупность микроопераций. В каждой ЭВМ набор микроопераций строго определен, количество их в реальных ЭВМ составляет от нескольких десятков до нескольких сотен. Время выполнения микроопераций различных типов различно. Например, время переписи информации из регистра в регистр существенно меньше времени чтения информации из оперативной памяти.

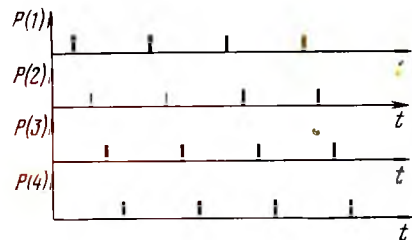


Рис. 5.2. Сигналы четырехфазного генератора синхросигналов

Частота импульсов генератора синхросигналов выбирается таким образом, чтобы за время синхротакта могла быть выполнена самая длительная микрооперация. Для возможности выполнения в течение одного синхротакта последовательности связанных коротких микроопераций генератор синхросигналов делают много-

фазным (рис. 5.2). При этом устройство управления вырабатывает инициирующие сигналы, связанные с различными выходами (фазами) генератора синхросигналов, которые и запускают требуемые микрооперации.

Изложенные принципы работы ЭВА реализуются в программах моделирования на регистровом уровне. Так, работа моделирующей программы протекает по синхротактам, внутри каждого из которых выполняется одна или несколько микроопераций.

Моделирование ЭВА на уровне регистровых передач заключается в покомандной имитации выполнения машинной программы, записанной в памяти этой ЭВА, и анализе преобразований представленной в двоичной форме информации, циркулирующей между функциональными устройствами. Моделируемая машинная программа, записанная на уровне машинных команд, играет роль исходных данных для моделирующей программы. Управление работой моделирующей программы осуществляется с помощью директив.

В описании модели ЭВА на регистровом уровне можно выделить две основные части: описание структуры объекта моделирования и операционное описание. В описании структуры указываются элементы, входящие в состав моделируемого объекта (регистры, счетчики, запоминающие устройства, генератор синхросигналов, шины передачи данных и т. п.). Операционное описание соответствует действиям, выполняемым в устройстве управления и операционном устройстве объекта. Часть операционного описания, соответствующая управляющему устройству, имитирует формирование инициирующих воздействий путем анализа кода операции

выполняемой команды, синхросигналов, триггеров управления и представляет собой множество условных операторов, которые передают управление в случае выполнения условия группе операторов, моделирующих микрооперации операционного устройства.

5.3. ЯЗЫКИ МОДЕЛИРОВАНИЯ НА УРОВНЕ РЕГИСТРОВЫХ ПЕРЕДАЧ

Для моделирования ЭВА на уровне регистровых передач могут применяться универсальные алгоритмические языки: ПЛ/1, Фортран и т. п., однако их использование характеризуется рядом неудобств — громоздкостью описаний, слабой связью структуры моделирующей программы и моделируемого устройства, большим объемом требуемой памяти и времени моделирования. Целесообразнее для этой цели применять специальные автономные языки моделирования ЭВА (см. гл. 2).

Одним из наиболее мощных языков, предназначенных для моделирования структур и процессов вычислительных машин, является язык APL [20]. Он содержит большое число различных операций и типов переменных (скаляры, векторы, матрицы и т. п.), средства для описания параллельно выполняемых операций, средства учета времени и синхронизации процессов. Язык удобен для описания алгоритмов работы ЭВМ, способ реализации которых еще не определен. Кроме этого, язык APL может использоваться и при программировании обычных задач вычислительного и информационно-логического характера. Однако универсальность языка затрудняет его изучение и использование при моделировании ЭВА.

Специально для описания ЭВА на регистровом уровне разработаны языки CDL (Computer Design Language) [21] и DDL (Digital Design Language) [11]. В этих языках имеются специальные операторы описания функциональных элементов ЭВА — регистров, сумматоров, дешифраторов и т. п., с их помощью задается структура моделируемого устройства, а работа устройства представляется на уровне элементарных действий над двоичными словами, таких как пересылка слов из устройства в устройство, сдвиг, инвертирование и др. Языки CDL и DDL во многом похожи друг на друга. Это относится и к идеологии построения, и к способам записи операторов. Различия касаются синтаксиса и набора операций. Модель устройства, представленная на одном из этих языков, отличается компактностью, наглядностью и соответствием структур программы и объекта. Языки удобны как для моделирования устройств, так и для описания их функционирования.

При проектировании вычислительных машин ЕС ЭВМ применялся язык ОСС-2 (Описание Структур и Схем) [19]. Язык ОСС-2 является многоуровневым, позволяющим описывать проектируемую ЭВМ, начиная с этапа разработки архитектуры и кончая принципиальной схемой.

5.4.1. Средства описания структуры. В описании модели на языке CDL выделяются две части: описание структуры моделируемого объекта и операционное описание.

Описание структуры объекта моделирования представляет собой перечисление всех функциональных устройств, имеющихсх в объекте, с указанием их некоторых характеристик. В языке CDL для описания структуры используются специальные описатели, в которых определяется тип и характеристики каждого из устройств, а также задается его имя. В табл. 5.1 приведены основные типы функциональных устройств ЭВА и их описатели, используемые в языке CDL.

Таблица 5.1

Тип устройства	Описатель
Регистр	REGISTER
Запоминающее устройство	MEMORY
Генератор синхросигналов	CLOCK
Шина передачи информации	BUS
Выводы (выходы) логических схем и устройств	TERMINAL
Дешифратор	DECODER
Ключи ручного управления	SWITCH

Рассмотрим правила описания функциональных устройств в моделирующих программах.

Для описания регистров используется описатель REGISTER, за которым записывается имя регистра и его разрядность. Например, шестнадцатиразрядный регистр, которому в программе присвоено имя А, описывается следующим образом:

REGISTER, A(0—15).

В языке CDL предусмотрена возможность объединения нескольких регистров или частей регистров в один регистр. Имеются средства для обозначения части регистра в качестве самостоятельного регистра. Например, описание

SUBREGISTER, R(OP) = A(0—5), R(ADDR) = A(8—15)

означает, что левым шести битам регистра присвоено имя R(OP), а его правым восьми битам — символическое имя R(ADDR).

В описании запоминающего устройства указывается длина ячейки памяти (количество разрядов), объем памяти, т. е. количество ячеек, и имя регистра, являющегося адресным для устройства. Например, описание

REGISTER, C(0—9)

MEMORY, M(C) = M(0—1023, 0—15)

определяет запоминающее устройство объемом 1024 ячейки. Длина ячеек 16 разрядов. Адресным регистром памяти является регистр С.

Описание ключей ручного управления определяет имя ключа и его состояния. Ключ может иметь одно (кнопка), два (тумблер) или несколько состояний. Возможные состояния ключа указываются в скобках. Например,

SWITCH, POWER (ON, OFF)

задает тумблер включения-выключения питания.

Дешифратор (логическая схема, преобразующая любое значение, содержащееся в регистре, в одно и только одно выходное значение) описывается именами его выходов и именем регистра, являющегося управляющим для дешифратора. Например, описание

DECODER, K (0—15) = F

указывает, что K(0—15) являются выходами дешифратора, соединенного с регистром F.

Выходы комбинационных схем, отдельные выходы регистров и дешифраторов могут описываться самостоятельно. Так, продолжая предыдущий пример, описание

TERMINAL, ADD = K(0)

будет присваивать имя ADD выводу K(0) дешифратора.

Генератор синхросигналов описывается заданием имени и количества фаз сигналов, выдаваемых генератором. Например, описание

CLOCK, P(4)

определяет генератор сигналов P, имеющий четыре фазы. Синхроимпульсы во всех четырех фазах имеют постоянную частоту следования и смещены относительно друг друга во времени. На рис. 5.2 дана временная диаграмма этих сигналов.

Шина передачи информации представляет собой набор проводников, которые могут подключаться к входам и выходам различных базовых элементов для передачи информации. В описании шины задается ее имя и количество содержащихся в ней цепей. Например,

BUS, OUT (0—15)

определяет шину OUT, содержащую 16 проводников.

5.4.2. Операционное описание. Описание структуры позволяет указать на наличие в моделируемом объекте тех или иных функциональных элементов и определить некоторые их характеристики. Операционное описание — это описание действий, выполняемых над содержимым регистров, ячеек запоминающих устройств, триггеров управления и т. д., соответствующих отдельным микрооперациям или их группам. Взаимосвязи элементов и способы обработки информации определяются описаниями операций. В опера-

циях также описываются некоторые или все условия, при которых устанавливаются эти взаимосвязи и выполняются действия. Операндами в операциях являются имена объявленных при описании структуры моделируемого объекта элементов, обозначающие входные или выходные контакты, и константы, представляющие собой фиксированные воздействия на входах элементов.

Одной из основных операций в вычислительных машинах является операция переписи информации с одного устройства в другое. Операция переписи информации в какое-либо запоминающее устройство (регистр, оперативную память, триггер) обозначается в языке CDL левой стрелкой. Например, если А и В регистры, имеющие одинаковую разрядность, то выражение $A \leftarrow B$ обозначает перепись информации из регистра В в регистр А.

В языке CDL имеется возможность описания различных логических операций над двоичными словами и отдельными разрядами слов, операций циклического и простого сдвига информации вправо и влево в регистрах, операции добавления и вычитания 1 из содержимого счетчика и т. п. Перечень основных операций приведен в табл. 5.2.

Т а б л и ц а 5.2

Операция	Обозначение
Логическое НЕ	$\neg$
Логическое И	$*$
Логическое ИЛИ	$+$
Арифметическое сложение	ADD
Арифметическое вычитание	SUB
Сдвиг вправо	SHR
Сдвиг влево	SHL
Циклический сдвиг вправо	CIR
Циклический сдвиг влево	CIL
Добавление 1 к счетчику	COUNTUP

Логические операции над двоичными словами производятся поразрядно. Например, если А, В и С — регистры, имеющие одинаковое количество разрядов, то операция $A \leftarrow \neg B + \neg C$ определяет действия поразрядного логического сложения инвертированных состояний регистров В и С и записи результата в регистр А.

В логических операциях и операциях переписи информации могут использоваться отдельные разряды регистров и шин либо группы разрядов. Например, если имеется описание

REGISTER, A(0—7), B(0—7), C(0—7),

то могут быть выполнены операции

$$A(0) \leftarrow B(0) * C(0),$$

$$A(1—7) \leftarrow B(1—7) + C(1—7).$$

В операциях могут использоваться числовые константы. Обычно используются десятичные константы. При выполнении операций они представляются в форме целых двоичных правоустановленных чисел. Например, если F и R — регистры, то запись $F \leftarrow 0$, $R \leftarrow 3$ указывает, что все разряды регистра F устанавливаются в 0, а в регистре R два крайних правых разряда устанавливаются в 1 ($3_{10} = 11_2$), а остальные устанавливаются в 0.

Допускается установка в определенное состояние некоторых разрядов регистра. Так, если C — шестнадцатиразрядный регистр, то запись

$$C(0) \leftarrow 0, C(8-15) \leftarrow 255$$

означает установку крайнего левого разряда в 0, а восьми правых разрядов в 1 (число 255 в двоичной форме имеет вид 11111111).

Операция добавления 1 к счетчику (регистру), часто используемая в ЭВМ, записывается

$$D \leftarrow \text{COUNTUP } D,$$

где D — регистр.

В языке предусмотрены две арифметические операции — сложения (ADD) и вычитания (SUB), которые соответственно суммируют или вычитают два двоичных числа без знака. Записываются эти операции следующим образом. Если A и R — регистры, то выражение

$$A \leftarrow A \text{ ADD } R$$

предполагает суммирование содержимого регистров A и R и запись результата в регистр A.

Для записи операций, реализуемых при выполнении некоторых условий, используется условный оператор. Используются две формы — полная и укороченная. Полная имеет вид IF... THEN... ELSE, а укороченная IF... THEN. Например если G — триггер, а A, B и C — регистры, то может быть записан оператор

$$\text{IF } (G=1) \text{ THEN } (A \leftarrow B) \text{ ELSE } (A \leftarrow C),$$

который определяет действие $A \leftarrow B$, если состояние G равно 1, и действие $A \leftarrow C$, если G равно 0.

Таким образом, проверяемое условие находится между словами IF и THEN и представляет собой операцию сравнения «=» (равно). Если условие выполняется, то выполняются операторы, стоящие между THEN и ELSE. В противном случае выполняются операторы, стоящие за словом ELSE, а если слова ELSE нет, то никакие действия не выполняются.

Операционное описание представляет совокупность некоторых групп операций, каждая из которых предназначена для реализации конкретных машинных команд ЭВА или частей команд. В реальном объекте операции, связанные с выполнением команд, инициируются сигналами устройства управления.

В моделирующей программе на языке CDL для имитации сигналов устройства управления используется понятие метка. Факти-

чески метка — это запись условий, при которых должны выполняться операции. Значением метки является 0 или 1. Метка записывается в виде логического выражения, заключенного в косые скобки. Следом за ней приводятся соответствующие ей операции, например

$$/K(1)*P(1)/ \quad A \leftarrow B, \quad C \leftarrow D.$$

В этой записи, если $K(1)$ — выход дешифратора кода операций, а $P(1)$ — выход первой фазы генератора синхросигналов, то в регистр A переписывается содержимое регистра B , а в регистр C — содержимое регистра D в момент, когда на выходах $K(1)$ и $P(1)$ будет присутствовать логическая 1, причем операции выполняются параллельно во времени.

В реальных объектах для формирования управляющих сигналов часто применяют дешифраторы, вырабатывающие сигналы в соответствии с кодом операции выполняемой команды, и многофазные генераторы синхросигналов. В этом случае метки (управляющие сигналы) в программах моделирования могут быть записаны следующим образом:

REGISTER,	$H(0-5)$
DECODER,	$K(0-63) = H$
CLOCK,	$P(0-2)$
$/K(0)*P(0)/$	
$/K(0)*P(1)/$	
$/K(0)*P(2)/$	

Здесь $K(0)$ — выход дешифратора. Аналогичная последовательность может быть записана и для других выходов дешифратора. Например, для $K(3)$:

$/K(3)*P(0)/$	
$/K(3)*P(1)/$	
$/K(3)*P(2)/$	

Таким образом, при использовании трехтактного генератора каждому выходу дешифратора (т. е. каждому коду операции) могут соответствовать три управляющих сигнала, подаваемых в такты $P(0)$, $P(1)$, $P(2)$.

Совокупность всех меток моделирующей программы, по существу, описывает работу устройства управления ЭВМ. Например, последовательность операций выборки информации из ячейки оперативной памяти в устройстве, содержащем счетчик D адреса команды, регистр C адреса запоминающего устройства M и трехтактный генератор синхросигналов P , записывается следующим образом:

$/F*P(0)/$	$C \leftarrow D$
$/F*P(1)/$	$R \leftarrow M(C).$

Здесь F — управляющий сигнал, разрешающий считывание из памяти. В такт $P(0)$ производится перепись адреса из счетчика

команд D в адресный регистр С памяти и иницируется процедура чтения. В такт Р(1) информация из ячейки запоминающего устройства переписывается в буферный регистр R.

5.5. ОПИСАНИЕ ЭВМ НА ЯЗЫКЕ CDL

В качестве примера рассмотрим описание на языке CDL одноадресной ЭВМ, структурная схема которой представлена на рис. 5.3.

В машине имеется запоминающее устройство (ЗУ), устройство управления (УУ) и арифметическое устройство (АУ). Запоминающее устройство состоит из памяти М, буферного регистра R и адресного регистра С, УУ — из управляющего регистра F, связанного с ним дешифратора К, и программного счетчика (счетчика команд) D, управляющего триггера пуска-останова G, трехфазного генератора синхросигналов Р и ключей (кнопок) управления POWER, START и STOP. Выходы дешифратора К(0—9) формируют управляющие сигналы выполнения команд. Когда в триггере G находится 1, машина находится в состоянии «Работа», в про-

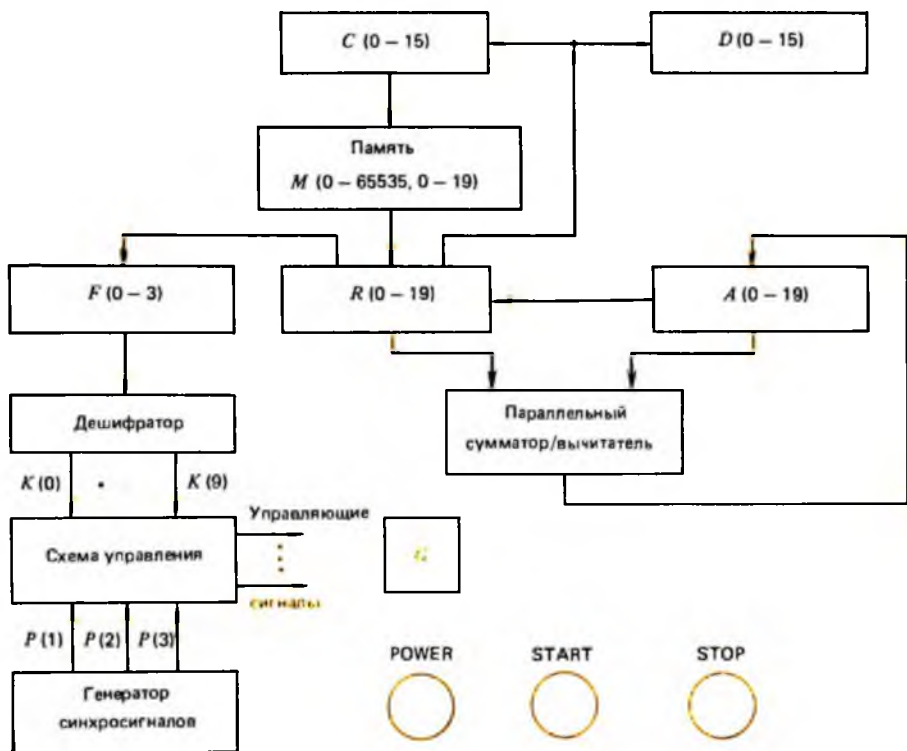


Рис. 5.3. Структурная схема ЭВМ

тивном случае она находится в состоянии «Ожидание». Кнопка POWER включает питание машины, кнопка START осуществляет пуск, а кнопка STOP — останов машины. Арифметическое устройство параллельного действия состоит из накапливающего регистра А, регистра R и комбинационного сумматора-вычитателя. Арифметическое устройство работает с целыми числами.

В ЭВМ используется машинное слово длиной 20 разрядов. Операнды — 20-разрядные целые числа, в левом разряде знак. Команды состоят из 4-битового поля кода операции и 16-битового поля адреса. Форматы команд и чисел приведены на рис. 5.4, а и б.

В машине могут выполняться девять команд: сложение (ADD), вычитание (SUB), условный переход (JOM), запись в ЗУ (STO), безусловный переход (JMP), сдвиг вправо (SHR), сдвиг влево (CIL), очистка сумматора (CLA) и останов (STP).

Каждая команда в рассматриваемой машине выполняется за два синхротакта. В первом такте осуществляется выборка команд

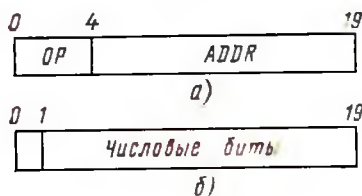


Рис. 5.4. Форматы команд и чисел

из ЗУ, дешифруется код операции, адрес операнда пересылается в регистр адреса ЗУ и формируется адрес следующей команды. Операции первого синхротакта одинаковы для всех команд. Во втором такте выполняются операции, реализующие конкретную команду. Для каждого типа команд этот набор операций различен. Так, при выполнении операции ADD из ЗУ считывается операнд (по адресу, заданному в команде) в регистр R, далее содержимое регистра R суммируется с содержимым регистра А и результат отсылается в регистр А. При выполнении других типов команд соответственно меняется состав операций. Конкретные действия при реализации каждого типа команд представлены в операционном описании ЭВМ (см. п. 5.5.2). Предполагается, что цикл работы ЗУ равен одному такту синхрогенератора Р. Инициация работы ЗУ осуществляется по сигналу первой фазы синхрогенератора Р(1), запись результата из ячейки памяти в буферный регистр R по сигналу второй фазы Р(2), а запись информации в ЗУ по сигналу третьей фазы Р(3).

5.5.1. Описание структуры ЭВМ. В соответствии с рис. 5.3 на языке CDL структура ЭВМ описывается следующим образом. Справа даны пояснения

REGISTER, R(0—19),
A(0—19),
C(0—15),
D(0—15),
F(0—3),
G

буферный регистр ЗУ
накапливающий регистр АУ
адресный регистр ЗУ
счетчик адреса команд ЗУ
управляющий регистр УУ
триггер пуска-останова

SUBREGISTER	$R(OP) = R(0-3),$	разряды кода операции
MEMORY,	$R(ADDR) = R(4-19)$	команды
DECODER,	$M(C) =$ $= M(0-65535, 0-19)$	разряды адреса команды
SWITCH,	$K(0-9) = F$	запоминающее устройство
TERMINAL,	$POWER(ON),$ $START(ON),$ $STOP(ON)$	дешифратор кода операции
		кнопка включения ЭВМ
		кнопка пуска ЭВМ
		кнопка останова
		далее следуют условные обозначения выводов дешифратора
	$ADD = K(0),$	операция сложения
	$SUB = K(1),$	операция вычитания
	$JOM = K(2),$	операция перехода по минусу
	$STO = K(3),$	запись в память
	$JMP = K(4),$	операция безусловного перехода
	$SHR = K(5),$	операция сдвига вправо на один разряд
	$CIL = K(6),$	операция циклического сдвига влево на один разряд
	$CLA = K(7),$	операция очистки накапливающего регистра
	$STP = K(8),$	останов
	$FETCH = K(9)$	операция выборки команды
CLOCK,	$P(1-3)$	трехфазный генератор синхросигналов

5.5.2. Операционное описание ЭВМ. Описание содержит несколько групп операций. Первая группа определяет действия, выполняемые при пуске машины, вторая — при выборке команд из ЗУ ЭВМ, последующие группы определяют действия, выполняемые для реализации различных команд.

Операции пуска-останова

/POWER(ON)/ $G \leftarrow 0, F \leftarrow 8$

включение питания, установка управляющего триггера G в состояние «Ожидание», установка начального состояния управляющего регистра в состояние STP

/START(ON) * $G \leftarrow 1$
*P(2)/

установка триггера G в состояние «Работа» (осу-

ществляется на фазе P(2), так как на фазе P(1) анализируется состояние G)
установка триггера G в состояние «Ожидание»

/STOP(ON)/ $G \leftarrow 0$

Операции выборки команды из ЗУ

*/FETCH*P(1)/* IF ($G=0$) THEN
 ($F \leftarrow 8$)
 ELSE ($C \leftarrow D$)

если в триггере управления G записан 0, то в управляющий регистр записывается код команды «Останов», если в G записана 1, то номер команды записывается в адресный регистр ЗУ и иницируется операция считывания из ЗУ

*/FETCH*P(2)/* $R \leftarrow M(C)$
*/FETCH*P(3)/* $F \leftarrow R(OP)$

считывание из ЗУ
код операции переписывается в управляющий регистр F

Далее следует описание операций выполнения различных команд. Каждая из команд (кроме команды «Останов») заканчивается операцией $F \leftarrow 9$, после чего дешифратор выдает сигнал FETCH (выборка), т. е. подготавливается выполнение следующей команды.

Операции команды ADD

*/ADD*P(1)/* $C \leftarrow R(ADDR)$
*/ADD*P(2)/* $R \leftarrow M(C)$
*/ADD*P(3)/* $A \leftarrow A + R, F \leftarrow 9$

считывание операнда из ЗУ
запись операнда в буферный регистр

Операции команды SUB

*/SUB*P(1)/* $C \leftarrow R(ADDR)$
*/SUB*P(2)/* $R \leftarrow M(C)$
*/SUB*P(3)/* $A \leftarrow A - R, F \leftarrow 9$

Операции команды записи в память

*/STO*P(1)/* $C \leftarrow R(ADDR)$
*/STO*P(2)/* $R \leftarrow A$
*/STO*P(3)/* $M(C) \leftarrow R, F \leftarrow 9$

запись адреса в адресный регистр

Операции команды CLA

*/CLA*P(1)/* $C \leftarrow R(ADDR)$
*/CLA*P(2)/* $R \leftarrow M(C), A \leftarrow 0$
*/CLA*P(3)/* $A \leftarrow A + R, F \leftarrow 9$

Операции команды STP

```

/STP*P(1)/   G←0
/STP*P(3)/   IF (G=0) THEN (G←0,
              D←0)
              ELSE F←9
    
```

проверка состояния триггера G необходима в момент P(3), так как в момент P(2) с помощью кнопки START возможна установка $G \leftarrow 1$

Остальные команды

```

/JMP*P(3)/   D←R(ADDR), F←9
/JOM*P(3)/   IF (A(0)) THEN
              D←R(ADDR), F←9
/SHP*P(3)/   A←SHR A, F←9
/CIL*P(3)/   A←CIL A, F←9
    
```

засылка адреса перехода в регистр

Приведенный пример демонстрирует общие принципы использования языка CDL для описания структуры и алгоритмов функционирования ЭВА.

5.6. ОСОБЕННОСТИ ОРГАНИЗАЦИИ МОДЕЛИРОВАНИЯ

Моделирование выполняется с использованием специальных систем. Структура системы моделирования представлена на рис. 5.5. Система содержит транслятор, переводящий описание модели с алгоритмического языка на машинный язык или некоторую промежуточную форму, и управляющую программу, организующую процесс моделирования.

Как уже говорилось, целью моделирования является проверка правильности алгоритмов выполнения отдельных команд, проверка

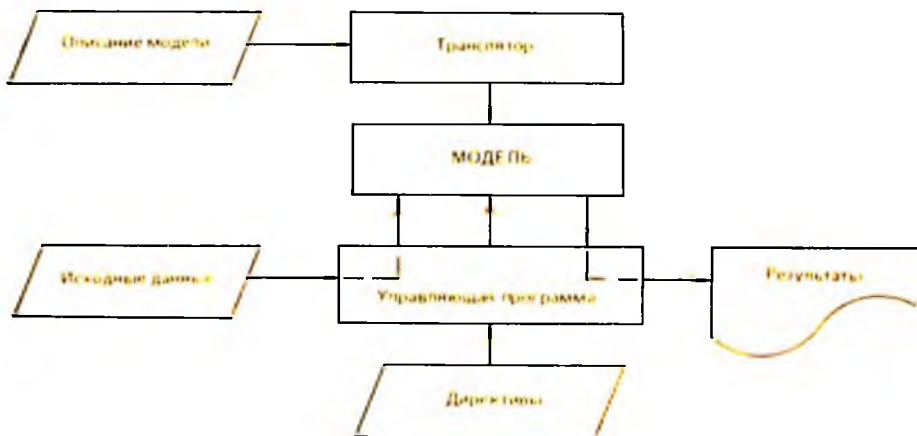


Рис. 5.5. Структура системы моделирования

оптимальности выбранной структуры, наблюдение за потоками информации в моделируемом объекте и т. п. В системе для управления процессом моделирования используют директивы. Директивы — это специальные операторы, воспринимаемые управляющей программой, позволяющие определить последовательность и содержание действий системы моделирования. Обычно используются следующие директивы:

трансляции;

ввода исходной информации в модель объекта (запись программы в ЗУ модели объекта, установка начальных состояний регистров, триггеров, переключателей);

определения точек объекта, в которых должна контролироваться (выводиться) информация в процессе моделирования;

задания режимов работы;

продолжительности моделирования;

определения действий системы моделирования при возникновении каких-либо непредвиденных ситуаций.

Результаты моделирования для каждого такта работы модели обычно выводятся на печатающее устройство или дисплей и представляют собой содержимое регистров, ячеек ЗУ, триггеров управления и т. п. Результаты выводятся в виде двоичных кодов либо представляются в восьмеричной или шестнадцатеричной системах счисления.

Анализируя содержимое ячеек ЗУ, регистров, триггеров управления в различные такты работы и сопоставляя их с тем, что предполагалось получить, определяют правильность работы устройства, алгоритмов и всего объекта в целом.

ГЛАВА 6

ЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

6.1. ПОСТАНОВКА ЗАДАЧИ

Следующим после этапа проектирования функциональных схем ЭВА является этап логического проектирования. На этом этапе детально разрабатываются логические схемы устройств, т. е. схемы на уровне простейших логических элементов (НЕ, И, ИЛИ и т. п.). Исходной информацией для разработки служат функциональные схемы, алгоритмы их работы и система базовых логических элементов, из которых строится устройство.

В настоящее время для разработки логических схем частично используются типовые функциональные решения, частично — традиционный интуитивный способ. В некоторых случаях применяют системы автоматизированного синтеза логических схем. Во всех случаях для формального описания схем, для решения задач синтеза и минимизации числа элементов в схемах применяется аппарат булевой алгебры и теории автоматов.

Корректность разработанных схем оценивается либо макетированием схем и проверкой их работы в различных режимах, либо моделированием на ЭВМ, причем моделирование применяется все чаще, так как позволяет ускорить процесс проектирования и уменьшить затраты труда. С помощью моделирования не только оценивается правильность функционирования разработанных схем, но и анализируются переходные процессы в схемах, надежность работы схемы, вероятность сбоев, генерируются тесты для проверки устройств и т. д.

Существует достаточно большое число методов моделирования логических схем. Они различаются степенью соответствия используемых моделей сигналов и элементов реальным объектам, т. е. степенью адекватности. Естественно стремление к увеличению адекватности моделирования, однако это приводит, как правило, к увеличению времени моделирования и требуемой памяти. Выбор конкретного метода моделирования определяется требованиями решаемой задачи и имеющейся технической базой.

Кроме основной задачи — проверки правильности логического функционирования устройства, модель на уровне логических элементов может использоваться для исследования временных характеристик и процессов в устройстве (рисков сбоев, критических состояний), а также для генерации тестов.

В САПР РЭА и ЭВА программы логического моделирования используются достаточно широко, однако каждая из программ имеет узко специальное назначение и применение ее в других системах затруднительно. В связи с этим отсутствуют какие-либо широко распространенные языки логического моделирования и соответствующие им системы программирования. Как правило, для каждой САПР программы логического моделирования разрабатываются заново, так же как и язык описания логических схем.

6.2. АВТОМАТИЗИРОВАННЫЙ СИНТЕЗ ЛОГИЧЕСКИХ СХЕМ

Автоматизированный синтез логических схем с использованием ЭВМ является достаточно сложной задачей. Сложность ее объясняется большим разнообразием как проектируемых схем, так и используемых для их построения логических элементов. Имеются сведения о нескольких системах автоматизированного синтеза логических схем.

Так, в [22] описана система ПРОЕКТ, предназначенная для синтеза различных цифровых устройств на логическом уровне. Система разработана для проверки и отработки принципов автоматизированного проектирования. Она позволяет использовать различные алгоритмы проектирования и расширять состав средств проектирования. Разработчик управляет системой с помощью директив в диалоговом режиме, загружая в память требуемые для работы программы. Предусмотрено прерывание процесса проектирования и проектирование устройства по частям. Опыт постро-

ния и эксплуатации системы был использован при разработке других систем.

Исходной информацией для проектирования является описание структуры, которое дается на специально разработанном языке Структура, и описание функциональных алгоритмов, которое производится на языке Алгоритм. Анализируя исходное описание, система выделяет три типа подустройств: операционные, управляющие и функциональные. Операционные устройства — это устройства, содержащие регистры, управляющие — схемы управления с памятью, а функциональные — комбинационные схемы. Далее проводится синтез соответствующих операционных, управляющих и функциональных блоков.

В системе ALERT [23] алгоритм функционирования проектируемого устройства и его структура представлены в виде программы на языке APL (см. § 5.1). Анализируя исходное описание, система выделяет комбинационные схемы и элементы памяти. После этого все устройства описываются системой булевых функций, которые после минимизации используются для построения схем из соответствующих базовых элементов.

В [24] рассмотрена еще одна система автоматизированного синтеза логических схем. В ней проектируемое устройство представляется в виде совокупности управляющих блоков, блоков с типовыми функциональными решениями и блоков, разработанных неавтоматизированными методами. Исходная структура устройства описывается на языке Структура, а алгоритм функционирования — на языке Микропрограмма, являющемся модификацией языка Алгоритм системы ПРОЕКТ. Синтез логических схем производится только для управляющих блоков. Логические схемы блоков других типов представляются непосредственно на языке Структура либо уже хранятся в архиве системы. В процессе синтеза логических схем управляющие блоки разделяются на две части: память и комбинационную схему, после чего задача сводится к задаче синтеза комбинационных схем.

Из рассмотренных примеров систем видно, что задача синтеза логических схем устройств требует для своего решения:

описание структуры проектируемого устройства на регистровом уровне;

описание алгоритма функционирования устройства на уровне микроопераций;

задания базовой системы логических и запоминающих элементов, которые могут быть использованы в устройстве.

Само же решение задачи сводится к выделению в проектируемом устройстве элементов памяти и комбинационных схем, синтезу комбинационных схем в заданной системе логических элементов и выбору запоминающих элементов.

Системы автоматизированного синтеза логических схем представляют собой сложные и трудоемкие программные комплексы, которые используют мощные ЭВМ. Так, процесс преобразования регистрового описания устройства в логическую схему требует

5...8 просмотров (проходов) и постепенного преобразования исходного описания. Автоматизированный синтез логических схем может быть произведен для относительно простых фрагментов реальных блоков, так как для сложных блоков затраты машинного времени недопустимо велики. Схемы, полученные с помощью систем автоматизированного синтеза, имеют в два-три раза большее число элементов и меньшую скорость работы по сравнению со схемами, разработанными неавтоматизированно. Поэтому на современном этапе развития вычислительной техники и алгоритмов системы автоматизированного синтеза не нашли широкого применения.

Далее рассмотрим основные модели сигналов и элементов, используемые при логическом моделировании.

6.3. МОДЕЛИ СИГНАЛОВ И ЭЛЕМЕНТОВ

6.3.1. Модели сигналов. Наиболее просто представить сигналы в виде логических 0 и 1. При этом один из электрических уровней принимается за 0, другой за 1. Переход сигнала из одного состояния в другое считается мгновенным и не отражается в модели сигнала.

Достаточно часто используется представление сигналов в виде трех значений, обозначаемых 0, 1 и x , где 0 и 1 имеют обычный смысл, а x обозначает либо переход из одного состояния в другое, либо неопределенное состояние. При таком представлении сигналов производится троичное моделирование.

В системах более детального моделирования используется большее количество символов для обозначения различных состояний сигналов. Например, применяют пятизначное кодирование — 0, 1, x , h , l , где 0 и 1 используются для обозначения верхнего и нижнего уровня сигналов, x — для неопределенного значения сигнала, а h и l — соответствуют переходам из 0 в 1 и из 1 в 0. Предложены и другие — семи-, девяти- и более значные представления сигналов. Дополнительные символы вводятся для обозначения каких-либо физических процессов, на которые хотят обратить внимание, например динамический риск сбоя, флуктуации сигналов и т. п.

Для моделирования устройств, содержащих микропроцессорные элементы, в сигналах выделяют так называемое высокоимпедансное состояние, когда цепь отключена от источника информации.

6.3.2. Модели элементов. Под элементом будем понимать простейшую функционально законченную часть логической схемы устройства, например отдельные комбинационные схемы типа И, ИЛИ, И—НЕ, ИЛИ—НЕ и т. п. или их более сложные комбинации. Часто на выбор элементов моделирования влияют конструктивные объединения логических элементов. Так, в качестве отдельного элемента могут рассматриваться триггеры, дешифраторы

ры и другие элементы, оформленные конструктивно в одном корпусе.

При логическом моделировании используются функциональные модели элементов. Функциональная модель представляется в виде «черного ящика», для которого связь между входными и выходными сигналами задается в виде булевых уравнений, таблицы истинности или другими способами. В функциональных моделях структура моделируемого элемента не задается. Простейшими моделями комбинационных логических элементов — схем И, ИЛИ, НЕ и др. являются реализуемые ими соответствующие булевы функции. При многозначном моделировании модель элемента задается с помощью многозначной таблицы истинности. В моделях, учитывающих временные задержки, указываются значения задержек сигнала при переключении элементов.

На рис. 6.1 представлены таблицы истинности для схем И, ИЛИ, НЕ для троичного и пятизначного моделирования.

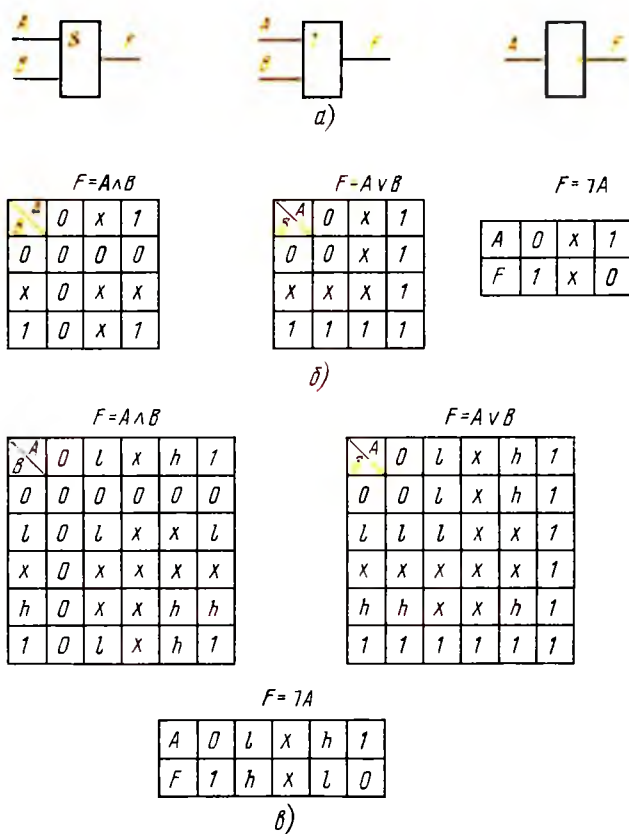


Рис. 6.1. Логические элементы И, ИЛИ, НЕ (а) и их таблицы истинности для трехзначного (б) и пятизначного (в) моделирования

При логическом моделировании могут решаться весьма разнообразные задачи: проверка логики работы схем, анализ переходных процессов, определение надежности работы схем в зависимости от разброса параметров комплектующих элементов, генерация тестов и т. п. В зависимости от поставленной задачи выбирается метод моделирования. Основными отличительными чертами методов являются: способ учета времени распространения сигнала в схеме, способ кодирования сигналов, способ построения модели в ЭВМ, очередность моделирования элементов.

В зависимости от способа учета времени распространения сигнала методы делятся на синхронный (без учета задержек в элементах схемы) и асинхронный (с учетом задержек); в зависимости от способа представления сигналов — на двоичный и многозначный (троичный, пятизначный и т. д.), по способу организации работы программы моделирования — на метод компиляции и метод интерпретации; по организации очередности моделирования — сквозной и событийный (рис. 6.2).

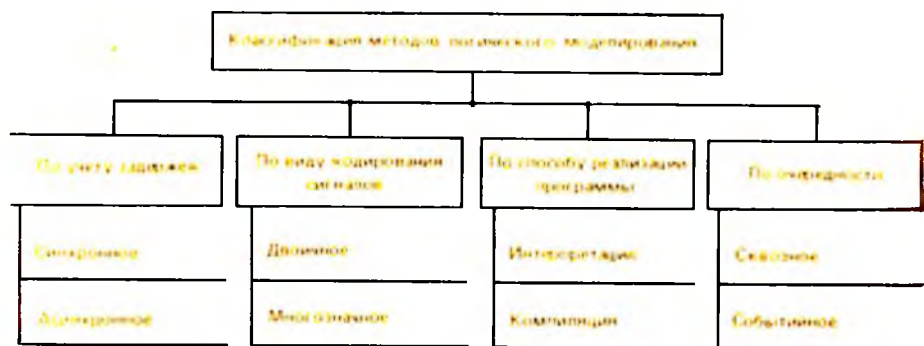


Рис. 6.2. Классификация методов логического моделирования

6.4.1. Синхронное логическое моделирование. При синхронном моделировании модели элементов представляются их логическими функциями без учета задержек сигналов, а сигналы — значениями 0 и 1. Синхронное моделирование применяется для оценки правильности логического функционирования дискретных устройств без учета переходных процессов. Моделирование работы схемы сводится к вычислению значений сигналов на выходах логических элементов схемы по заданным входным сигналам. Моделирование производится для каждого изменения сигналов на входах. Обычно эти изменения связаны с синхросигналами, поэтому значения сигналов на выходах элементов вычисляются для каждого синхросигнала. Предполагается, что в промежутке между синхросигналами входные сигналы не меняются, а переходной про-

цесс в схеме завершается в течение некоторого Δt , меньшего периода повторения тактирующих сигналов.

Наиболее удобно использование синхронного моделирования для анализа работы комбинационных схем в установившемся режиме. В этом случае результат моделирования точно соответствует реальной схеме.

Каждый из элементов схемы описывается логическим уравнением $Y=f(x_1, x_2, \dots, x_n)$, где Y — выходной, $x_1, x_2, \dots, x_n$ — входные сигналы элемента. В результате вся схема описывается системой таких уравнений. Синхронное моделирование сводится к их решению последовательно одно за другим. Под решением логического уравнения понимается вычисление логического значения Y по известным логическим значениям $x_1, x_2, \dots, x_n$. Особенностью синхронного моделирования является решение логических уравнений в определенном порядке, соответствующем последовательности прохождения сигналов через элементы схемы. Для определения этого порядка схему предварительно нужно ранжировать (см. § 4.8), чтобы к моменту решения каждого уравнения $Y=f(x_1, x_2, \dots, x_n)$ значения всех $x_1, \dots, x_n$ были известны. При ранжировании принимается, что входные сигналы схемы к моменту начала ее моделирования известны и им присваивается ранг $r=0$.

Пример 6.1. Для схемы, представленной на рис. 6.3, порядок вычисления сигналов будет следующим:

$$\begin{aligned} C_5 = C_2 \& C_3 \quad \left. \begin{array}{l} C_5 = C_2 \& C_3 \\ C_6 = C_3 \& C_4 \end{array} \right\} r=1 \quad \begin{array}{l} C_7 = C_5 \mid C_6 \\ C_8 = C_1 \& C_7 \end{array} \quad \begin{array}{l} r=2, \\ r=3, \end{array}$$

где C_1, C_2, C_3, C_4 — входные сигналы схемы; C_5, C_6, C_7, C_8 — выходные сигналы элементов; r — ранг элементов схемы.

Для выполнения ранжирования схем разработаны формальные алгоритмы, задача обычно решается автоматически в самой системе моделирования.

Несколько сложнее моделируются логические схемы с обратными связями (рис. 6.4). Для рассмотрения принципов моделиро-



Рис. 6.3. Ранжирование логических схем

вания таких схем удобно в цепь обратной связи ввести элемент задержки (в схеме на рис. 6.4 — элемент D). В реальных схемах задержка D как физический элемент отсутствует, а задержка происходит из-за конечного времени срабатывания комбинационных схем.

При моделировании логических схем с обратными связями схема ранжируется и моделируется как обычная комбинационная схема в предположении, что в месте включения элемента задержки цепь обратной связи временно размыкается и цепям обратной связи, подключенным к входам элементов, присваивается ранг 0, такой же как и входным цепям. При этом считается, что по цепям обратной связи на входы элементов по-

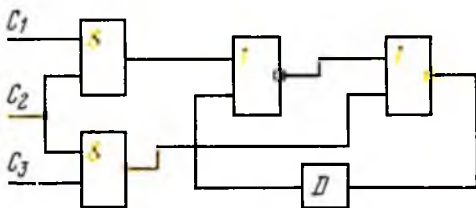


Рис. 6.4. Логическая схема с обратными связями

даются сигналы, соответствующие ранее заданным состояниям 0 или 1. Схема моделируется и новые вычисленные значения сигналов обратной связи подаются на входы элементов через некоторое время Δt , равное времени задержки цепи обратной связи.

Для синхронных схем, у которых период синхроимпульсов обычно равен времени задержки D в цепях обратной связи, на этом моделирование заканчивается. Для асинхронных, т. е. нетактируемых, схем цикл моделирования повторяется с новыми значениями сигналов обратной связи до появления установившегося состояния или до обнаружения генерации — чередования 0 и 1 от цикла к циклу.

Результатом синхронного моделирования является временная диаграмма, представленная в виде последовательности 0 и 1. На диаграмме для каждого такта моделирования приводятся значения входных воздействий и значения сигналов на выходах элементов схемы. По ней анализируется правильность работы устройства. Обычно моделирование устройства производится для некоторой совокупности тестовых воздействий, для которых известна эталонная реакция схемы. Сопоставляя в конкретном случае результаты с требуемой эталонной реакцией, определяют правильность работы устройства. В случае обнаружения несоответствия более детальный анализ временной диаграммы позволяет локализовать ошибки в схеме.

Синхронное моделирование с двоичным представлением сигналов является простейшим способом моделирования. Его важное достоинство заключается в скорости, однако синхронное моделирование не позволяет анализировать переходные процессы в схемах и выявлять ошибки, которые могут возникнуть из-за задержки сигналов в элементах схемы.

6.4.2. Асинхронное моделирование. Анализ переходных процессов в логических схемах ведется асинхронным методом моде-

лирования, в котором учитывается время распространения сигналов в элементах и соединительных цепях схемы.

Срабатывание логического элемента происходит с некоторым запаздыванием по отношению к входным сигналам, которое учитывается задержкой в моделях элементов. Каждый элемент характеризуется некоторой средней задержкой, значение которой может меняться в зависимости от режима работы элемента, комбинации входных сигналов, температуры, отклонения в технологии изготовления элемента и т. д. В зависимости от требуемой степени адекватности моделирования учет задержек производится с той или иной степенью детализации. Задержки в линиях связи весьма малы и обычно не учитываются, однако при моделировании устройств с очень высоким быстродействием учитываются и они.

Временное рассогласование входных сигналов элемента может привести к появлению ложного сигнала на выходе логического элемента. Возможность появления ложных сигналов называется риском сбоя. Если сигналы на выходе схемы для двух смежных наборов входных воздействий A и B остаются одинаковыми, а во время переходного процесса возможно появление ложного сигнала противоположного значения, то такая ситуация называется *статическим риском сбоя* (рис. 6.5). Временное рассогласование сигналов, как видно из рисунка, приводит к появлению на выходе ложного сигнала.

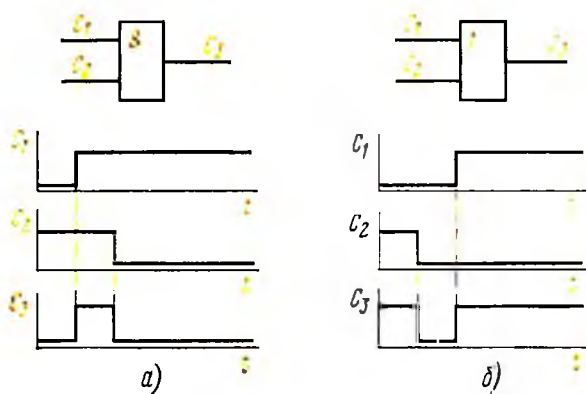


Рис. 6.5. Временные диаграммы работы элементов И (а) и ИЛИ (б) при статическом риске сбоя

Динамический риск сбоя предполагает возможность многократного изменения сигнала на выходе при переходе от входного набора A к набору B , когда выходной сигнал меняется на противоположный (рис. 6.6). Как видно из рисунка, при переходе от набора $A = C_1C_2C_3 = \{010\}$ к набору $B = C_1C_2C_3 = \{101\}$ возникает динамический риск сбоя. Динамический риск сбоя является следствием статического риска сбоя.

В логических схемах с памятью, т. е. в схемах с обратными связями, под воздействием входных сигналов могут изменить состояние сразу несколько элементов памяти. Окончательное состояние схемы в некоторых случаях зависит от очередности переключения элементов памяти. В этих случаях говорят, что в схеме существуют состязания сигналов обратных связей, или просто —

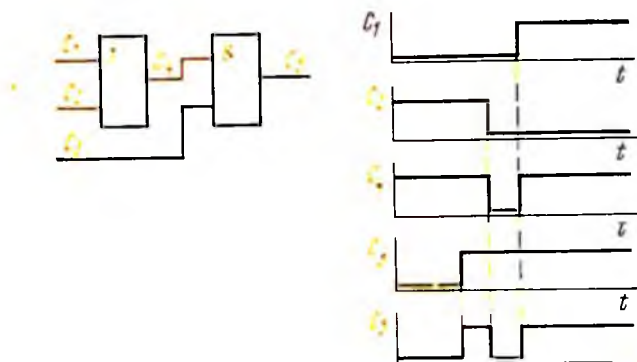


Рис. 6.6. Временная диаграмма работы элемента при динамическом риске сбоя

состязания сигналов. Если под воздействием входного сигнала схема из одного состояния может перейти в различные состояния в зависимости от задержек в элементах схемы, то в этом случае состязания называются *критическими*.

Асинхронное моделирование позволяет выявлять риски сбоев и критические состязания.

Модель логического элемента для асинхронного моделирования представляется в виде последовательного соединения безынерционного логического элемента, реализующего указанную функцию, и элемента задержки (рис. 6.7, а).

Асинхронное моделирование заключается в вычислении сигналов на выходах логических элементов схемы в соответствии с рассмотренной моделью. При асинхронном моделировании требуется определить не только состояние на выходе элемента (0 или 1), но и момент времени, когда произошли изменения. В простейшем случае для этого многократно просчитываются состояния элементов схемы через некоторый интервал времени Δt , который выби-

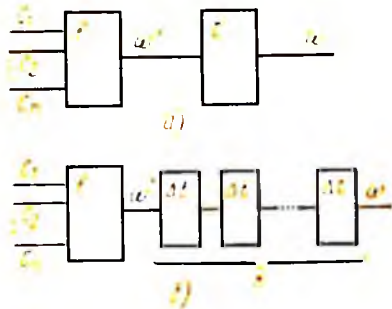


Рис. 6.7. Асинхронная модель логического элемента (а) и асинхронная модель с дискретными задержками (б)

рается как наибольший общий делитель времен задержек элементов, используемых в моделируемом устройстве. Так, если в устройстве используются элементы, имеющие задержки 15, 27, 21 и 30 нс, то $\Delta t = 3$ нс.

Таким образом, каждый элемент имеет задержку $\tau = K\Delta t$, где $K\Delta t = 1, 2, 3, \dots$, а модель элемента может быть представлена в виде цепочки (рис. 6.7,б), в которой первый компонент реализует логическую функцию, а остальные K элементов — элементарные линии задержки на Δt . За каждый цикл моделирования сигналы в схеме продвигаются на время, равное Δt . Если схема тактируема, то время ее такта T разбивается на последовательность микротактов длительностью Δt и моделирование сводится к многократному решению уравнений, описывающих схему. Окончание моделирования одного такта работы схемы происходит либо по истечении времени T , либо по окончании переходного процесса в схеме.

При асинхронном моделировании ранжирование схемы не производится.

По сравнению с синхронным асинхронное моделирование требует выполнения существенно большего количества вычислений. Объем программ также возрастает из-за необходимости моделирования элементов задержек.

6.4.3. Асинхронное событийное моделирование. Анализ работы дискретных устройств показывает, что одновременно находятся в активном состоянии лишь 1...2,5% всех элементов схемы. Отсюда следует, что существенное уменьшение времени моделирования может быть достигнуто, если каждый раз моделировать только те элементы, у которых изменились входные сигналы. В этом и заключается принцип событийного моделирования. Событие в системах событийного моделирования — это изменение состояния какого-либо элемента и связанных с ним цепей.

В системах асинхронного событийного моделирования время моделирования изменяется не постепенно с заранее заданным шагом Δt , а в моменты возникновения событий. Моменты же возникновения событий определяются задержками логических элементов, которые в общем случае различны.

В программах асинхронного событийного моделирования важную роль играют два массива — *массив состояния цепей* моделируемой схемы и *очередь будущих событий*. Массив состояния цепей хранит текущие состояния всех цепей моделируемой схемы в виде логических 0 и 1. В очередь будущих событий (ОБС) в процессе моделирования записываются события, которые должны произойти в моделируемой схеме. Каждый элемент ОБС содержит номер цепи, в которой должно произойти изменение состояния (т. е. произойдет событие), и момент времени возникновения события. В ОБС события записаны в порядке возрастания времени, и в вершине очереди находится событие, которое произойдет раньше всех.

Асинхронное событийное моделирование выполняется следующим образом. Перед началом моделирования устанавливается исходное состояние схемы путем записи значений в массив состояния цепей. Тестовые входные воздействия, подаваемые в схему, заносятся в ОБС в соответствии с временем их появления. Далее начинается собственно моделирование, которое состоит из следующих действий.

1. Из ОБС выбирается верхний элемент. Время, указанное в нем, заносится в счетчик модельного времени, а в массив состояния цепей по номеру, указанному в элементе, вместо старого производится запись нового состояния цепи, указанного в ОБС.

2. Находятся логические элементы, для которых данная цепь является входной, и вычисляются значения сигналов на выходах этих элементов (т. е. определяются новые состояния цепей) и их задержки.

3. Для каждой из цепей значение сигнала сравнивается со значением, хранящимся в массиве состояния цепей, и, если они не совпадают, то, следовательно, происходит изменение состояния цепи, и событие заносится в ОБС. Если значения совпадают, то запись в ОБС не производится.

Далее операции повторяются, начиная с п. 1.

Процесс моделирования заканчивается при исчерпании всех элементов ОБС либо заданного времени моделирования.

Результаты асинхронного моделирования представляются в виде временных диаграмм, по которым и анализируется работа устройств, в частности, выявляются критические состязания, статические и динамические риски сбоев. Наиболее полно поведение моделируемого устройства раскрывается при формировании задержек элементов с учетом их случайного характера.

Недостатком асинхронного моделирования является существенно большее по сравнению с синхронным время моделирования.

Значительно быстрее риски сбоев и критические состязания могут быть выявлены моделированием логических схем с использованием многозначного представления сигналов.

6.4.4. Методы многозначного моделирования. Из методов многозначного моделирования наиболее часто используется троичное моделирование. Представление сигналов при троичном моделировании дано в п. 6.3.1. Следует отметить, что символом x , кроме переходных ситуаций $0 \rightarrow 1$, $1 \rightarrow 0$, обозначается также и неопределенное состояние цепей. Например, символом x может быть обозначено неопределенное состояние цепей обратной связи триггера перед началом моделирования.

Достоинством троичного моделирования является простота реализации и высокое быстродействие, всего в 4...6 раз ниже, чем при двоичном синхронном моделировании.

При троичном моделировании задержки на элементах принимаются равными нулю. Модель логического элемента представ-

ляется таблицей истинностей, в которой каждый из входных сигналов может принимать три значения (см. п. 6.3.2).

Моделирование работы логических схем производится как на каждом из значений входных сигналов, так и на их промежуточном значении. Так, если входное воздействие представляет собой n -разрядный двоичный код $A = (a_1 a_2 \dots a_i \dots a_n)$ и этот код сменяется кодом $B = (b_1 b_2 \dots b_i \dots b_n)$, то промежуточное воздействие определяется как $A/B = (c_1 c_2 \dots c_i \dots c_n)$, где $c_i = x$, если $a_i \neq b_i$, и $c_i = a_i$, если $a_i = b_i$, при $i = 1, 2, 3, \dots, n$.

Входные наборы	C_1	C_2	C_3	C_4	C_5
A	0	1	0	1	1
A/B	x	x	x	x	x
B	1	0	1	1	0



Рис. 6.8. Пример троечного моделирования комбинационной схемы

Пример моделирования комбинационной схемы приведен на рис. 6.8. Троечное моделирование позволяет определить места в схеме, где возможны статические риски сбоев. В приведенном примере статический риск сбоя зафиксирован в цепи C_4 . Действительно, при переходе от набора A к набору B сигнал в этой цепи не должен меняться, однако во время смены входных сигналов зафиксировано неопределенное состояние $C_4 = x$, что показывает возможность появления ложного сигнала. Как видно из примера, динамический риск сбоя не выявляется (хотя в схеме он имеет место на цепи C_5). Значение x на выходе C_5 указывает только на смену значений с 1 на 0 при изменении входного набора A на B . Динамический риск сбоя может быть определен как следствие статического риска сбоя на одном из входов элемента.

Наличие критических состояний в последовательностных схемах также может быть выявлено путем троечного моделирования. При троечном моделировании последовательностных схем задаются начальные значения сигналов обратных связей (если они неизвестны, то принимается значение x), обратные связи разрываются и осуществляется ранжирование элементов. Далее вычисляются значения на выходах логических элементов схемы для заданного входного воздействия (переходного либо нового воздействия) и определяются новые значения сигналов обратных связей. Процесс моделирования для каждого входного воздействия (набора) осуществляется многократно до установления стабильного состояния или режима генерации. Алгоритм троечного моделирования представлен на рис. 6.9, а на рис. 6.10 дана схема, для которой в табл. 6.1 приведены результаты ее моделирования для двух последовательных входных наборов $A = \{1, 1\}$, $B = \{0, 1\}$. Исходные зна-

чения сигналов обратной связи схемы (в табл. 6.1 они не показаны) берутся неопределенными, т. е. $C_{12}=x$, $C_{13}=x$, $C_{14}=x$, $C_{15}=x$.

После подачи входного набора $\{1, 1\}$ состояние цепей обратной связи (или, что то же, сигналов в цепях обратной связи) устанавливается $C_{13}=0$, $C_{15}=0$. При подаче входного набора $\{0, 1\}$ пере-

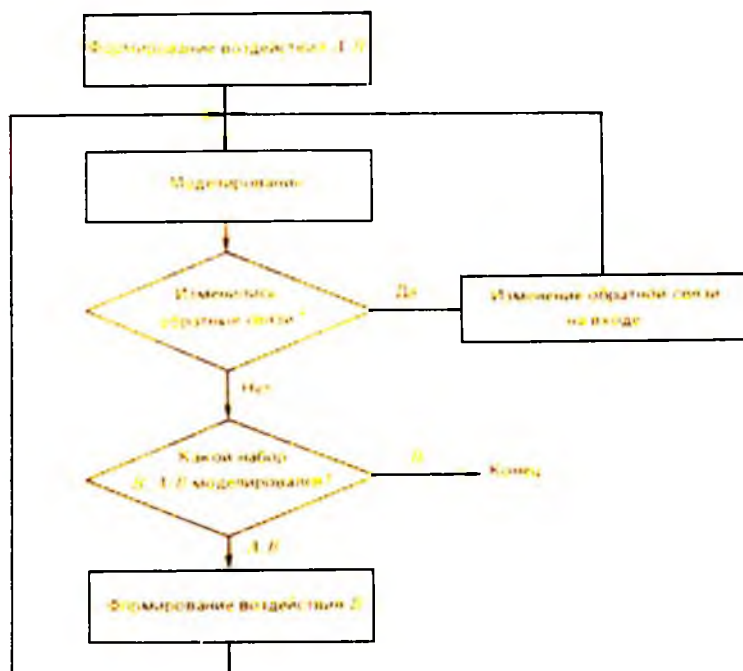


Рис. 6.9. Алгоритм тричного моделирования последовательных схем

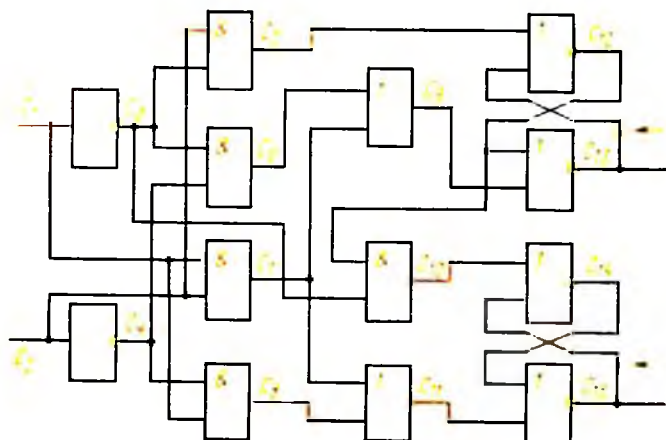


Рис. 6.10. Последовательная схема

ходный набор равен $A/B = \{x, 1\}$, а сигналы цепей обратной связи для переходного набора становятся равными $C_{13}=x$, $C_{15}=x$ и т. д. В результате моделирования схемы определяется, что сигналы цепи обратной связи C_{14} , C_{15} принимают устойчивое значение x .

Таблица 6.1

Входной набор	Номер итерации	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}
A	I	1	1	0	0	0	0	1	0	1	0	1	x	0	x	0
	II	1	1	0	0	0	0	1	0	1	0	1	1	0	1	0
	III	1	1	0	0	0	0	1	0	1	0	1	1	0	1	0
A/B	I	x	1	x	0	x	0	x	0	x	x	x	x	x	x	x
	II	x	1	x	0	x	0	x	0	x	x	x	x	x	x	x
B	I	0	1	1	0	1	0	0	0	0	x	0	0	1	x	x
	II	0	1	1	0	1	0	0	0	0	0	0	0	1	x	x
	III	0	1	1	0	1	0	0	0	0	0	0	0	1	x	x

Если в результате трюичного моделирования выявлено, что в схеме хотя бы один сигнал обратной связи принимает значение x , то в схеме имеются критические состязания. Таким образом, в рассмотренной выше схеме (рис. 6.10) существуют критические состязания сигналов.

Более детальные сведения о работе схемы могут быть получены при использовании пятизначного моделирования. Так, пятизначное моделирование позволяет обнаружить динамические риски сбоев в схемах.

Модели сигналов и элементов для пятизначного моделирования даны в п. 6.3.2. Существенным в пятизначном моделировании является обозначение не только факта изменения сигнала, как это делается при трюичном моделировании, но и характера изменения. Выделяется изменение сигнала $1 \rightarrow 0$ (обозначим это изменение символом l) и изменение $0 \rightarrow 1$ (обозначим h). Неопределенное состояние обозначается x .

Так же, как и в трюичном моделировании, в пятизначном моделировании схема просчитывается не только для двух последовательных входных наборов $A = (a_1 \dots a_i \dots a_n)$ и $B = (b_1 \dots b_i \dots b_n)$, но и для промежуточного набора A/B , который формируется следующим образом: $A/B = (c_1 \dots c_i \dots c_n)$, где $c_i = l$, если $a_i = 1$ и $b_i = 0$; $c_i = h$, если $a_i = 0$ и $b_i = 1$; $c_i = a_i$, если $a_i = b_i$ для $i = 1, 2, 3, \dots, n$.

На рис. 6.11 приведен пример пятизначного моделирования комбинационной схемы для наборов $A_1 = \{0, 1, 0\}$ и $B_1 = \{1, 0, 0\}$, а также для $A_2 = \{0, 0, 1\}$ и $B_2 = \{1, 0, 0\}$. Риски сбоев в моделируемых схемах обнаруживаются по появлению состояния x на переходных наборах, причем для статического риска сбоя состояние цепи на наборах A и B одинаково, а для динамического — различно. В дан-

ном примере при переходе от A_1 к B_1 статический риск сбоя имеется в цепи C_5 , а динамический — в цепях C_6, C_8 .

Алгоритм моделирования и анализа последовательностных схем при пятизначном моделировании аналогичен троичному.

Входные наборы	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
A_1	0	1	0	1	0	1	0	1
A_1/B_1	h	l	0	l	x	x	0	x
B_1	1	0	0	0	0	0	0	0
A_2	0	0	1	1	0	1	0	1
A_2/B_2	h	0	l	l	0	l	0	l
B_2	1	0	0	0	0	0	0	0

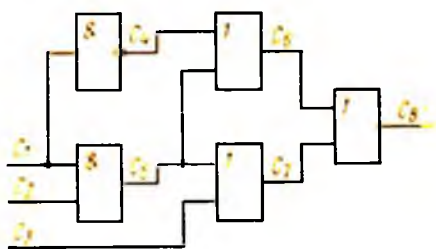


Рис. 6.11. Пример пятизначного моделирования

Принятая при многозначном моделировании идеализация предполагает произвольное соотношение задержек в элементах схемы. Вследствие этого при моделировании выявляются даже маловероятные случаи возникновения риска сбоя и критических ситуаций. Так, например, в схеме, изображенной на рис. 6.12, при смене входного набора с $\{1, 1\}$ на $\{1, 0\}$ троичное моделирование определяет новое состояние цепей обратной связи $\{x, x\}$, т. е. в схеме имеются критические состязания. На самом же деле состояние цепи C_4 остается равным 1, так как изменение состояния на элементе 4 произойдет раньше, чем придет сигнал с элемента 3, и состояние триггера не изменится.

Таким образом, многозначное моделирование позволяет определить все возможные места рисков сбоев, даже тех, которые в реальных схемах не могут произойти. Асинхронное же моделирование, напротив, может пропускать некоторые случаи рисков сбоев.

С целью повышения адекватности моделирования разработаны методы, совмещающие в себе элементы многозначного и асинхронного моделирования. Это — метод моделирования с нарастающей неопределенностью и метод Δ -троичного моделирования [24]. В них принята троичная система кодирования сигналов, однако возможные изменения задержек ограничены. Использование этих методов позволяет в значительной мере устранить недостатки, присущие как асинхронному методу, так и методу многозначного моделирования, но требует увеличения количества вычислительных опера-

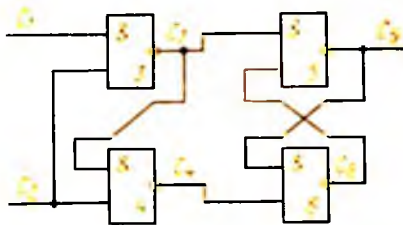


Рис. 6.12 К анализу критических состязаний

щий и времени моделирования. Это ограничивает применение методов случаями, когда необходимо весьма тщательное исследование логических схем.

6.5. СИНТЕЗ ТЕСТОВ С ПРИМЕНЕНИЕМ СИСТЕМ МОДЕЛИРОВАНИЯ

В процессе производства и эксплуатации ЭВА возникают задачи проверки правильности функционирования аппаратуры, отыскания неисправностей и их устранения.

Обнаружение и локализация неисправностей в ЭВА производится путем подачи на входы проверяемого устройства некоторой последовательности наборов (векторов) входных сигналов и анализа реакции устройства на эти сигналы. Каждому входному набору соответствует эталонный результат, полученный на исправной аппаратуре. Входной набор и соответствующий ему выходной набор называется *элементарной проверкой*. Совокупность входных наборов и соответствующих им эталонных выходных результатов называют *тестом* данного устройства. Тесты бывают *контролирующими* и *диагностическими*. При помощи контролирующих тестов определяется наличие или отсутствие неисправности в устройстве. При помощи диагностических тестов локализуется тип и место неисправности в устройстве. Обычно тесты представляют собой довольно длинные последовательности элементарных проверок. Разработка тестов является трудоемкой задачей, которая в большинстве случаев решается с применением ЭВМ. В САПР обычно имеется подсистема генерации тестов, основывающаяся на системе логического моделирования устройств.

С целью упрощения задачи создания тестов рассматривают ограниченный класс неисправностей, которые необходимо обнаружить. Предполагают, что в аппаратуре могут быть лишь одиночные неисправности типа обрывов и коротких замыканий выводов. Такие неисправности приводят к тому, что на входах или выходах элементов присутствует постоянный сигнал 0 либо 1. В некоторых случаях класс неисправностей расширяют рассмотрением инверсной неисправности, когда сигнал на выходе элемента имеет значение, инверсное по отношению к правильному значению.

Ограничив класс рассматриваемых неисправностей, можно с помощью системы логического моделирования получить диагностическую таблицу. В этой таблице каждая строка соответствует определенному входному набору (i), а каждый столбец — определенной неисправности (j) рассматриваемой схемы. В каждой клетке (a_{ij}) таблицы записывается 0 либо 1; 0 соответствует тому, что реакция схемы с j -й неисправностью при i -м воздействии совпадает с реакцией исправной схемы; 1 — обратной ситуации.

Если все столбцы диагностической таблицы попарно различимы, то в схеме может быть обнаружена любая неисправность из рассматриваемого класса.

Непосредственное использование диагностических таблиц для локализации неисправностей обычно невозможно, так как, во-пер-

вых, тест, состоящий из всех возможных наборов, слишком длинен, а, во-вторых, формирование диагностической таблицы затруднительно из-за большой трудоемкости. Для оптимизации процедуры разработки тестов используются различные методы: вероятностный метод, активации одномерного пути, *D*-алгоритм Рота и др.

Уменьшение длины тестов основано на том, что для обнаружения большинства неисправностей в схеме нет необходимости проводить все возможные элементарные проверки. Нахождение таких достаточно коротких тестов, позволяющих локализовать большинство неисправностей, производится вероятностным методом. Метод заключается в следующем: случайным образом формируется входной набор, и для него определяется строка диагностической матрицы; оцениваются диагностические возможности данного набора, т. е. определяется, позволяет ли он выявить дополнительные неисправности; принимается решение о включении набора в тест. С увеличением длины теста диагностические возможности каждого очередного набора уменьшаются и дальнейшее увеличение длины теста становится неэффективным.

Поиск входных наборов, локализующих неисправности, не найденные вероятностным методом, производится детерминированными методами. В детерминированных методах тест подбирают для каждой отдельной неисправности.

Из детерминированных методов наиболее широко используется так называемый *D*-алгоритм Рота. В этом методе для каждой неисправности определяют пути, т. е. цепочки элементов, по которым рассматриваемая неисправность может проявиться на выходе схемы. При этом отказываются от тех путей, которые невозможно активизировать из-за схождения разветвлений. Затем осуществляется обратный проход. На нем производится подбор входных сигналов, позволяющих рассматриваемой неисправности проявиться на выходе схемы, и исключаются пути, приводящие к противоречивым требованиям к входным переменным. Совокупность входных сигналов, подобранных для рассматриваемой неисправности, вместе с сигналами на выходе схемы, появляющимися при этой неисправности, и является тестом.

6.6. РЕАЛИЗАЦИЯ ПРОГРАММ ЛОГИЧЕСКОГО МОДЕЛИРОВАНИЯ

6.6.1. Компилирующие и интерпретирующие системы. В зависимости от способа реализации программы в ЭВМ системы моделирования делятся на компилирующие и интерпретирующие.

В системах компилирующего типа исходное описание моделируемой схемы, представленное на каком-либо входном языке, транслируется на машинный язык и оформляется в виде объектного модуля, который затем и выполняется при моделировании. Для перевода исходного описания схемы на машинный язык используется специальная программа — транслятор, являющаяся частью системы моделирования.

В системах интерпретирующего типа перевод входного описания на машинный язык не производится, а каждый оператор входного описания выполняется с помощью специальной подпрограммы. Таким образом, для каждого типа операторов входного языка применяется собственная подпрограмма.

Сравнивая системы интерпретирующего и компилирующего типов, необходимо отметить следующее. В системах интерпретирующего типа входное описание модели хранится в оперативной памяти ЭВМ во время моделирования и вследствие этого внесение изменений в модель в интерпретирующих системах осуществляется достаточно просто. В системах компилирующего типа внесение изменений в модель (т. е. в исходный текст описания) требует его повторной трансляции. Преимуществом систем компилирующего типа является большая скорость работы. Однако разработка транслятора для перевода описания схемы на машинный язык представляет более сложную и трудоемкую задачу, чем разработка программного обеспечения интерпретирующей системы. Таким образом, системы интерпретирующего типа проще и менее трудоемки для создания, однако, если требуется большая скорость работы системы моделирования, предпочтение следует отдать системам компилирующего типа.

6.6.2. Языки логического моделирования. В настоящее время отсутствует какой-либо широко распространенный специализированный язык для логического моделирования схем. Обычно для каждой вновь разрабатываемой системы моделирования предлагается свой язык.

В системах моделирования компилирующего типа, как правило, применяют языки, основывающиеся на каком-либо распространенном универсальном алгоритмическом языке. Так, система МОДИС-В [25] использует алголоподобный язык, а система ФОРЭС [26] — язык, в некотором отношении близкий к языку Фортран. В системах интерпретирующего типа обычно используются автоматные языки описания схем. Описание моделируемой схемы на этих языках состоит из отдельных предложений, каждое из которых служит для задания очередного логического элемента.

```
EXMP: PROC OPTIONS(MAIN);
      DCL C(8) BIT(1);
      ON ENDFILE(SYSIN) GO TO FIN;
      L: GET LIST(C(1),C(2),C(3));
          CALL MODEL(C);
          PUT SKIP EDIT(C)(X(20)B V(1));
          GO TO L;
      FIN: END EXMP;
MODEL: PROC(C);
      DCL C(8) BIT(1);
      C(4)=~A;
      C(5)=A&B;
      C(6)=C(4)|C(5);
      C(7)=C(3)|C(5);
      C(8)=C(6)|C(7);
      RETURN;
      END MODEL;
```

Рис. 6.13. Программа синхронного моделирования на языке ПЛ/1

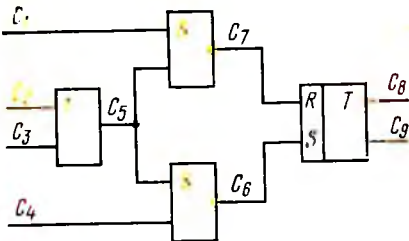
В некоторых случаях в качестве языка входного описания и моделирования может использоваться универсальный алгоритмический язык (Алгол, Фортран, ПЛ/1 и т. п.). На рис. 6.13

приведена программа на языке ПЛ/1 синхронного моделирования схемы рис. 6.11. Программа состоит из основной процедуры, выполняющей операции ввода-вывода информации и вызова подпрограммы-модели, и собственно подпрограммы-модели.

Входные воздействия для моделирования считываются с перфокарт. Результаты для каждого из входных воздействий печатаются в виде 0 и 1, возникающих на выходах элементов схемы. Подпрограмма содержит логические уравнения, описывающие модели элементов схемы и связи между ними.

Использование языка автоматного типа для описания логической схемы показано на рис. 6.14. Как видно из примера, каждому элементу моделируемой схемы соответствует строка описания. В ней указывается тип элемента (И, ИЛИ, НЕ, 2И, ...), номера входных цепей и после пробела — номера выходных цепей. Такое описание предполагает наличие в моделирующей системе библиотек моделей логических элементов с соответствующими именами — И, ИЛИ, НЕ, 2И и т. п.

Как уже указывалось, процедура моделирования состоит в подаче на вход модели некоторых воздействий, вычислении реакции модели и анализе этих реакции. Количество воздействий, подаваемых на модель, может быть достаточно большим, поэтому для описания входных воздействий разрабатываются и используются так называемые языки входных воздействий. В них обычно последовательность входных воздействий задается явным образом в виде сигналов на входах схемы для каждого входного набора. В некоторых случаях воздействия представляются в виде номеров входных контактов, на которых для данного набора произошла смена сигналов.



ВХОДЫ	1, 2, 3, 4
2ИЛИ	2, 3 5
2И-НЕ	1, 5 7
2И-НЕ	5, 4 6
T-RS	7, 6 8, 9

Рис. 6.14 Моделируемая схема и пример ее описания

ГЛАВА 7

СХЕМОТЕХНИЧЕСКОЕ МОДЕЛИРОВАНИЕ

7.1. ПОСТАНОВКА ЗАДАЧИ

В этой главе схемотехническое моделирование (СхМ) понимается в узком смысле, как моделирование электрических процессов в электронных устройствах, традиционно изображаемых в виде принципиальных электрических схем, т. е. соединений условных обозначений диодов, транзисторов, резисторов и т. д.

Схемотехническое моделирование электрических процессов учитывает в отличие от идеализированного моделирования информационных процессов реальные физические ограничения в электрических процессах — так называемые законы сохранения. Такими ограничениями являются первый и второй законы Кирхгофа. Они вытекают из законов сохранения заряда и работы и называются обычно *законами электрического равновесия*. Необходимость выполнения этих законов в каждой расчетной точке требует решения соответствующих уравнений электрического равновесия, что составляет наиболее существенное отличие СхМ от функционального, логического и других видов информационного моделирования, не требующего решения уравнений равновесия.

В связи с этим в математическую модель электронного устройства в СхМ входят не только модели отдельных элементов и уравнения их связи, как в информационном моделировании, но и уравнения электрического равновесия, составляемые на основе законов Кирхгофа и называемые обычно *топологическими уравнениями*. Уравнения отдельных элементов схемы называются *компонентными*. Таким образом, математическая модель для СхМ в общем случае состоит из двух подсистем уравнений — компонентной и топологической.

Более высокая степень строгости описания электронных схем при СхМ позволяет получить более точные сведения о процессах в схемах по сравнению с моделированием на информационном уровне, например функциональным. Платой за это служит увеличение времени моделирования из-за необходимости решения уравнений равновесия.

Цель СхМ состоит обычно в определении формы и параметров сигналов тока и напряжения, возникающих в разных точках схемы. Для этого приходится решать ряд таких типовых задач СхМ, как расчет статического режима, переходных процессов, частотных характеристик. На основе решения этих задач можно далее вычислить параметры сигналов (фронт, длительность, задержку и др.), рассчитать спектр выходного сигнала, чувствительность схемы к изменению параметров ее элементов, решить задачи статистического анализа схемы и оптимизации ее параметров.

7.2. МОДЕЛИРОВАНИЕ СТАТИЧЕСКИХ РЕЖИМОВ

Расчет статического режима может иметь самостоятельное значение, например, при составлении карты режимов схемы по постоянному току, указывающей номинальные значения токов, напряжений и рассеиваемых мощностей в разных точках и элементах схемы. Однако чаще всего моделирование статического режима является основной составной частью других, более сложных задач. Например, расчет переходных процессов в схеме можно представить как последовательность расчетов квазистатических режимов в отдельные моменты времени, в каждый из которых нужно выполнить законы равновесия, т. е. найти квазистатический режим.

7.2.1. Прямой метод. Для расчета статического режима используют несколько форм исходных математических моделей. Наиболее распространена модель в виде системы конечных (не дифференциальных) линейных или нелинейных уравнений

$$F(x) = 0. \quad (7.1)$$

Составление модели схемы в виде (7.1) и ее последующее решение каким-либо численным методом получило название прямого метода расчета статического режима.

Для решения системы (7.1) можно использовать метод простых итераций

$$x^{(k+1)} = x^{(k)} - \lambda F(x^{(k)}), \quad (7.2)$$

где k — номер итерации; λ — множитель, регулирующий сходимость.

В общем случае алгоритм (7.2) не гарантирует сходимости, однако при расчете большинства апериодических негенераторных схем при достаточно малых λ алгоритм будет сходиться (так же как будет устойчив явный метод Эйлера, совпадающий по форме с (7.2)).

Главный недостаток этого алгоритма — медленная сходимость, поэтому вместо (7.2) для решения (7.1) обычно используют различные модификации метода Ньютона

$$F'(x^{(k)}) \Delta x^{(k)} = -F(x^{(k)}), \quad (7.3)$$

где $\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$ — вектор поправок; $F(x^{(k)})$ — вектор невязок; $F'(x^{(k)})$ — матрица Якоби.

Решая (7.3) при заданном начальном приближении $x^{(0)}$ относительно $\Delta x^{(0)}$, находим затем $x^{(1)} = \Delta x^{(0)} + x^{(0)}$, далее, вновь решая (7.3) относительно $\Delta x^{(1)}$, определим $x^{(2)}$ и т. д. до сходимости.

В отличие от линейно сходящегося метода простых итераций алгоритм Ньютона сходится квадратично, и если $F(x)$ — выпуклая функция, то решения с точностью 0,1...1% обычно получаются за 3...5 ньютоновских итераций, тогда как методом (7.2) нужно выполнить несколько десятков или даже сотен простых итераций. Таким образом, хотя вычислительные затраты времени и памяти на выполнение одной ньютоновской итерации больше, чем простой, однако высокая скорость сходимости метода Ньютона в итоге приводит к более быстрому решению системы (7.1). Недостатками метода Ньютона являются малый размер области сходимости (обычно меньше, чем для метода простых итераций) и необходимость задания точки начального приближения достаточно близко к точке решения — в противном случае скорость сходимости заметно снижается.

Теоретические условия сходимости метода Ньютона — существование и ограниченность обратной матрицы Якоби в точке начального приближения, ограниченность и непрерывность первых и вторых производных функций $F(x)$ — носят характер теоремы существования и не могут быть использованы на практике для определения положения и размера области сходимости.

В связи с этим начальные условия $X^{(0)}$ в общем случае приходится задавать эмпирически, при этом сходимость уже не гарантируется. Для повышения надежности сходимости можно использовать модификацию метода Ньютона — метод Ньютона—Бroyдена:

$$F'(x^{(h)}) \Delta x^{(h)} = -r_h F(x^{(h)}), \quad (7.4)$$

где r_h — параметр, выбираемый на каждой итерации из условия минимума нормы $(k+1)$ -й невязки $F(x^{(k+1)}) = F(x^{(k)}) - r_h (F'(x^{(k)})^{-1} \times F(x^{(k)})) \rightarrow \min$.

Минимизация нормы невязки $F(x^{(k+1)})$ на каждой итерации расширяет область сходимости этого метода по сравнению с обычным методом Ньютона.

7.2.2. Метод движущейся области сходимости. Для повышения надежности сходимости методов решения системы (7.1) часто применяется специальный прием — метод *движущейся области сходимости* (ДОС). В этом методе система $F(x) = 0$ заменяется эквивалентной системой $F(x, r) = 0$, где параметр $r \in (r_{\min}, r_{\max})$ выбирается так, что решение $x^{(0)}$ системы $F(x, r_{\min}) = 0$ известно, а $F(x, r_{\max}) = F(x)$. Однократное решение системы (7.1) заменяется серией решений с изменяющимся от решения к решению параметром r .

Для удобства описания этого метода введем оператор $\text{Alg}_x F(x)$, обозначающий решение системы $F(x) = 0$ любым численным методом, и оператор $\text{alg } F(x)$, обозначающий результат одной итерации. Тогда метод ДОС можно записать в следующем виде:

$$x_{p+1} = \text{Alg } F(x_p, r_{p+1}), \quad p = 0, 1, \dots, n-1; \quad (7.5a)$$

$$r_{p+1} = f(r_p), \quad r_0 = r_{\min}, \quad r_n = r_{\max}. \quad (7.5b)$$

где p — номер промежуточного решения (продолжения или шага).

В этом методе, если $x_p \in \delta x_p$, где δx_p — область сходимости около точки x_p , то возмущение параметра $\Delta r_p = r_{p+1} - r_p$ передвигает область сходимости и выполняется условие $x_{p+1} \in \delta x_{p+1}$ (рис. 7.1). Условием надежности метода является $x_p \in \delta x_{p+1}$, так как в этом случае при выборе в (7.5a) x_p в качестве начального

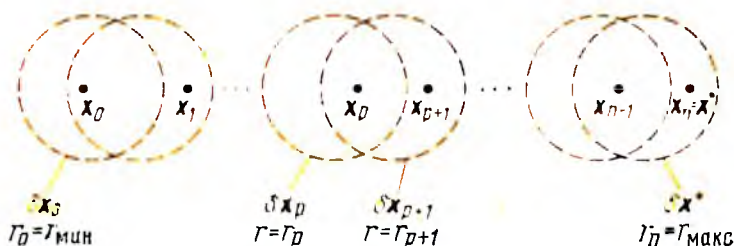


Рис. 7.1. Перемещение области сходимости и точки решения в методе движущейся области сходимости

приближения $(p+1)$ -го решения, это решение не расходится. Условие $x_p \in \delta x_{p+1}$ легко выполнить выбором r_{p+1} .

Конкретные способы реализации метода ДОС отличаются типом и способом изменения параметра r и функции $r_{p+1} = f(r_p)$. Обычно в качестве r выбирают напряжение источника питания E , причем $E_{p+1} = E_p + \Delta E_p$, где ΔE_p подбирается при каждом p из условия сходимости.

Иногда вместо полного цикла итерации до сходимости при каждом p , кроме последнего, выполняется только одна итерация, а при последнем p , когда $r = r_{\max}$ — полный цикл итераций. С помощью введенных выше операторов Alg и alg этот способ ДОС можно записать так:

$$x_{p+1} = \underset{x}{\text{alg}} F(x, r_{p+1}), p = 0, 1, \dots, n-2; \quad (7.6a)$$

$$r_{p+1} = f(r_p), r_0 = r_{\min}, r_n = r_{\max}; \quad (7.6б)$$

$$x^* = \underset{x}{\text{Alg}} F(x_{n-1}, r_{\max}), \quad (7.6в)$$

где x^* — решение системы $F(x) = 0$.

Этот способ более быстрый чем (7.5), но менее надежный, так как Δr_p выбирается произвольно и к моменту последнего решения (7.6в) может оказаться, что $x_{n-1} \notin \delta x_n$. Сравнительная графическая иллюстрация методов (7.5) и (7.6) показана на рис. 7.2, причем параметром служит напряжение источника питания E .

7.2.3. Метод установления. В этом методе статический режим рассматривается как состояние схемы, к которому она стремится асимптотически при затухании переходных процессов, возникающих в схеме при подаче на нее напряжения источника питания. Поэтому часто этот метод называют расчетом *статики через динамику*. Очевидно, математическая реализация этого метода требует составления и решения не системы (7.1), а системы дифференциальных уравнений, описывающих переходные процессы в схеме при включении напряжения питания.

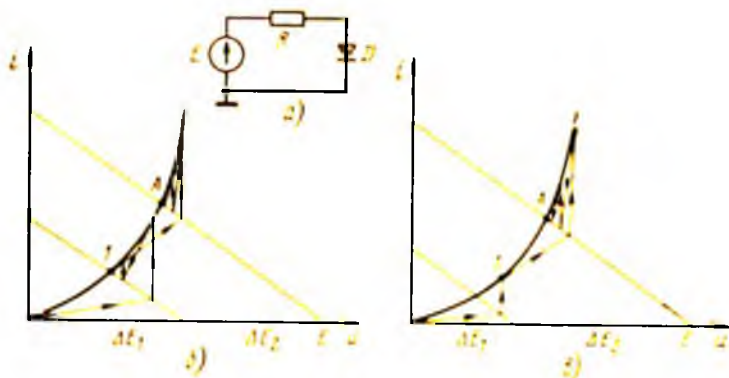


Рис. 7.2. Схема с диодом (а) и графическая иллюстрация ее расчета методами ДОС (б) и с одной итерацией на шаге (в)

Пример 7.1. Пусть нужно рассчитать статический режим простейшего резистивного делителя с емкостной нагрузкой (рис. 7.3). Это можно сделать либо прямым методом, решив уравнение

$$(E-u)/R_2 - u/R_1 = 0,$$

либо методом установления, решив дифференциальное уравнение

$$Cdu/dt = (E-u)/R_2 - u/R_1.$$

Очевидно, решение этого уравнения

$$u = \frac{ER_1}{R_1 + R_2} \left[1 - \exp \left(- \frac{t(R_1 + R_2)}{R_1 R_2 C} \right) \right]$$

асимптотически стремится к значению $u = ER_1/(R_1 + R_2)$, совпадающему с решением прямым методом.

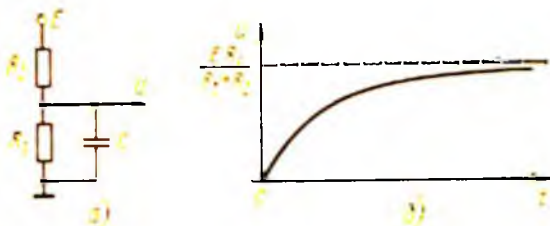


Рис. 7.3. Резистивный делитель (а) и эпюра установления статического режима (б)

Разумеется, в отличие от этого простейшего иллюстративного примера расчет сложных нелинейных схем методом установления ведется на основе численных методов решения дифференциальных уравнений и ничем не отличается от расчета переходных процессов в схеме за исключением того, что входным сигналом служит напряжение источника питания E , обычно увеличиваемое (во избежание расходимости вычислений) не скачком, а по линейному закону $E(t) = at$ при $E(t) \leq E$, а при $t > E/a$ равное номинальному значению: $E(t) = E$. Поскольку методом установления в основном определяется не форма переходного процесса, а его асимптотическое значение, то численное решение дифференциальных уравнений можно вести с максимально большим шагом, при котором численный метод решения еще сохраняет устойчивость.

Основные преимущества метода установления перед другими методами заключаются в следующем.

Во-первых, алгоритм составления математической модели и расчета статического режима мало отличается (за исключением выбора шага и критерия окончания расчета) от алгоритма составления модели и расчета переходного процесса. Благодаря этому сокращается объем и время разработки программы расчета статики и динамики.

Во-вторых, метод установления для большинства схем допускает достаточно произвольный выбор начальных условий $x^{(0)}$, по-

сколько в конечном счете переходный процесс независимо от $x^{(0)}$ стремится к одним и тем же асимптотическим значениям. Исключения составляют лишь многоустойчивые схемы, статическое состояние которых зависит от выбора $x^{(0)}$.

В-третьих, как правило, расчет статического режима выполняется не только для расчета статических параметров схемы, но весьма часто с целью определения начальных условий для расчета переходных процессов. Метод установления позволяет без дополнительных преобразований схемы определить начальные значения ее потенциалов с учетом всех реактивных элементов. Это важно отметить, поскольку, например, статические режимы схемы без индуктивностей и емкостей и с ними могут быть различны.

Недостатком метода установления следует считать довольно значительное время расчета статического режима и сложность определения момента окончания расчета.

7.2.4. Метод оптимизации. Этот метод основан на эквивалентности результатов решения системы (7.1) прямыми методами и минимизации какого-либо из функционалов

$$\Phi_1(x) = \sum_i F_i^2(x), \quad \Phi_2(x) = \sum_i |F_i(x)|, \quad \Phi_3(x) = \max_i |F_i(x)| \quad (7.7)$$

хорошо отработанными методами оптимизации. При этом количество решений системы (7.1) равно количеству минимумов в (7.7). Отсюда следует, что локальные методы оптимизации можно использовать для расчета статического режима схем с одним состоянием равновесия (ключи, усилители и т. д.). Если же схема имеет несколько состояний равновесия (триггеры), то локальные методы могут и не дать нужного решения. В целом, методы оптимизации не нашли широкого применения для решения системы (7.1) при расчете схем, так как в большинстве своем они уступают методу Ньютона по скорости сходимости, не обеспечивая в то же время высокой надежности сходимости. Повышение скорости сходимости методов оптимизации до уровня метода Ньютона требует вычисления матрицы Гессе $\Phi''(x)$, что существенно усложняет программы.

7.3. УРАВНЕНИЯ СТАТИЧЕСКОГО РЕЖИМА В БАЗИСЕ УЗЛОВЫХ ПОТЕНЦИАЛОВ

7.3.1. Основная математическая модель. В большинстве современных программ схмотехнического моделирования используется базис узловых потенциалов φ . Обычно теоретическая модель-алгоритм схемы не совпадает с видом исходной модели-уравнения.

В базисе узловых потенциалов исходная модель-уравнение, соответствующая (7.1), имеет вид

$$I(\varphi) = 0, \quad (7.8)$$

где $I(\varphi)$ — вектор узловых токов, в то время как теоретическая модель-алгоритм, используемая для расчета схемы, обычно фор-

мируется в виде, соответствующем решению (7.8) методом Ньютона,

$$Y(\varphi^{(k)}) \Delta \varphi^{(k)} = -I(\varphi^{(k)}), \quad (7.9)$$

где $Y = [\partial I / \partial \varphi]$ — матрица узловых проводимостей; k — индекс ньютоновских итераций; $\Delta \varphi^{(k)} = \varphi^{(k+1)} - \varphi^{(k)}$ — вектор поправок. Пусть схема содержит $n+1$ узлов, причем узел — это точка соединения двух и более ветвей.

Для удобства описания процесса формирования математической модели схемы (ММС), перепишем (7.9) в матричном виде:

$$\begin{bmatrix} \frac{\partial (\Sigma i)_1}{\partial \varphi_1} & \dots & \frac{\partial (\Sigma i)_1}{\partial \varphi_p} & \dots & \frac{\partial (\Sigma i)_1}{\partial \varphi_n} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial (\Sigma i)_p}{\partial \varphi_1} & \dots & \frac{\partial (\Sigma i)_p}{\partial \varphi_p} & \dots & \frac{\partial (\Sigma i)_p}{\partial \varphi_n} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial (\Sigma i)_n}{\partial \varphi_1} & \dots & \frac{\partial (\Sigma i)_n}{\partial \varphi_p} & \dots & \frac{\partial (\Sigma i)_n}{\partial \varphi_n} \end{bmatrix} \begin{bmatrix} \Delta \varphi_1^{(k)} \\ \dots \\ \Delta \varphi_p^{(k)} \\ \dots \\ \Delta \varphi_n^{(k)} \end{bmatrix} = - \begin{bmatrix} (\Sigma i)_1 \\ \dots \\ (\Sigma i)_p \\ \dots \\ (\Sigma i)_n \end{bmatrix}, \quad (7.10)$$

где $i = i(\varphi_1^{(k)}, \dots, \varphi_r^{(k)}, \dots, \varphi_n^{(k)})$ — ток ветви или полюсный ток многополюсника; $(\Sigma i)_j$ ($j = 1, \dots, p, \dots, n$) — узловой ток I_j j -го узла, т. е. алгебраическая сумма токов ветвей или полюсов элементов схемы, соединенных в j -м узле (инцидентных j -му узлу); $\partial (\Sigma i)_j / \partial \varphi_m$ — собственная (при $j = m$) или взаимная (при $j \neq m$) узловая проводимость Y_{jm} между узлами j и m .

7.3.2. Алгоритмы формирования модели схемы. Во всех программах СхМ модель схемы формируется не в виде уравнений исходной модели (7.8), а в виде векторов и матриц, используемых в теоретическом методе решения этих уравнений. Для метода Ньютона (7.9) формирование математической модели заключается в построении вектора узловых токов $I(\varphi^{(k)})$ и матрицы узловых проводимостей $Y(\varphi^{(k)})$, причем и то, и другое формируется при каждом k , т. е. на каждой итерации, как числовой вектор и числовая матрица, соответствующие значениям $\varphi^{(k)}$.

Методика формирования $I(\varphi^{(k)})$ и $Y(\varphi^{(k)})$ заключается в последовательном рассмотрении каждого элемента схемы и определении его вклада в $I(\varphi^{(k)})$ и $Y(\varphi^{(k)})$. Рассмотрим эту методику.

Формирование вектора узловых токов состоит в образовании для каждого узла суммы полюсных токов элементов, соединенных с этим узлом. При этом условимся считать, что ток, втекающий в узел, входит в узловой ток этого узла со знаком «—», а вытекающий — со знаком «+». Например, ток двухполюсника $i = f(\varphi_p - \varphi_j)$, включенного между узлами p и j , направленный от узла p к узлу j , будет участвовать в формировании p -го компонента I_p вектора узловых токов I в качестве слагаемого со знаком «+» и j -го компонента I_j — со знаком «—».

Рассмотрим методику формирования матрицы узловых проводимостей Y . Оценим сначала вклад каждого двухполюсника в эту матрицу. Если двухполюсник описывается уравнением $i=f(u)=f(\varphi_p-\varphi_j)$ и его ток направлен от узла p к узлу j , то его проводимость будет участвовать в качестве слагаемого при формировании четырех элементов матрицы Y : Y_{pp} , Y_{jj} , Y_{pj} , Y_{jp} .

Действительно, ток двухполюсника будет учтен в двух компонентах вектора узловых токов $I(\varphi)$: в компоненте $I_p(\varphi)$ как слагаемое $f(\varphi_p-\varphi_j)$ и в компоненте $I_j(\varphi)$ как слагаемое $-f(\varphi_p-\varphi_j)$. Тогда дифференцирование I_p по φ_p и φ_j даст проводимости $\partial f(\varphi_p-\varphi_j)/\partial \varphi_p$ в составе элемента $Y_{pp}=\partial I_p/\partial \varphi_p$ и $\partial f(\varphi_p-\varphi_j)/\partial \varphi_j$ в составе элемента $Y_{pj}=\partial I_p/\partial \varphi_j$, а дифференцирование I_j по φ_p и φ_j даст проводимости $\partial(-f(\varphi_p-\varphi_j))/\partial \varphi_p$ в составе элемента $Y_{jp}=\partial I_j/\partial \varphi_p$ и $\partial(-f(\varphi_p-\varphi_j))/\partial \varphi_j$ в составе $Y_{jj}=\partial I_j/\partial \varphi_j$.

Таким образом, при формировании матрицы Y схемы проводимость $y=di/du$ каждого двухполюсника $i=f(u)$, включенного между узлами p и j , должна записываться в качестве слагаемых на четырех позициях матрицы Y : со знаком «+» на диагональных позициях pp , jj в составе собственных узловых проводимостей Y_{pp} , Y_{jj} и со знаком «-» на позициях pj , jp , расположенных симметрично относительно диагонали, в составе взаимных узловых проводимостей Y_{pj} и Y_{jp} .

Заметим, что если один из узлов двухполюсника — опорный («земляной»), то проводимость y двухполюсника учитывается лишь в одном элементе матрицы Y — в собственной проводимости Y_{rr} , где r — незаземленный узел двухполюсника.

Пример 7.2. Сформируем, основываясь на приведенной выше методике, вектор узловых токов и матрицу узловых проводимостей для схемы на рис. 7.4,а. Используя для тока диода уравнение $I=I_T(e^{u/(m\varphi_T)}-1)$, получаем вектор узловых токов

$$I(\varphi) = \begin{bmatrix} I_1(\varphi_1, \varphi_2, \varphi_3) \\ I_2(\varphi_1, \varphi_2, \varphi_3) \\ I_3(\varphi_1, \varphi_2, \varphi_3) \end{bmatrix} = \begin{bmatrix} -\frac{E-\varphi_1}{R_1} + I_{T1}(e^{\frac{\varphi_1-\varphi_2}{m\varphi_T}} - 1) - I_{T2}(e^{\frac{\varphi_1-\varphi_3}{m\varphi_T}} - 1) \\ -I_{T1}(e^{\frac{\varphi_1-\varphi_2}{m\varphi_T}} - 1) + \frac{\varphi_2}{R_2} \\ -\frac{\varphi_2}{R_3} + I_{T2}(e^{\frac{\varphi_2-\varphi_3}{m\varphi_T}} - 1) \end{bmatrix}$$

m — эмпирический коэффициент; φ_T — температурный потенциал; I_T — тепловой ток.

Матрица узловых проводимостей этой схемы имеет вид

$$Y(\varphi) = \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & \frac{1}{R_1} + y_{д1} + y_{д2} & -y_{д1} & -y_{д2} \\ 2 & -y_{д1} & y_{д1} + \frac{1}{R_2} & 0 \\ 3 & -y_{д2} & 0 & \frac{1}{R_3} + y_{д2} \end{array}$$

где

$$y_{д1} = \frac{I_{T1}}{m \varphi_T} \exp \left(\frac{\varphi_1 - \varphi_2}{m \varphi_T} \right),$$

$$y_{д2} = \frac{I_{T2}}{m \varphi_T} \exp \left(\frac{\varphi_3 - \varphi_1}{m \varphi_T} \right).$$

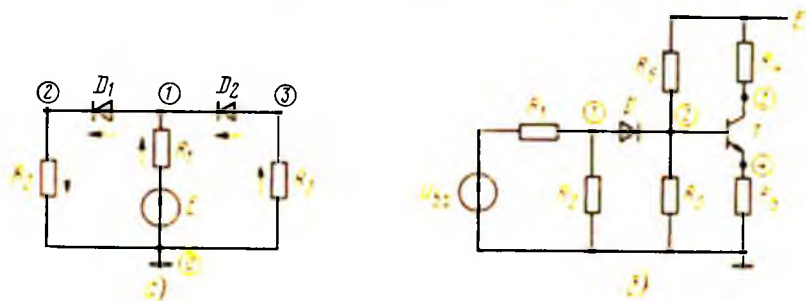


Рис. 7.4. К расчету схем в базисе узловых потенциалов:
а — диодно-резистивная схема; б — схема с транзистором

7.3.3. Учет многополюсников при формировании ММС. Вектор узловых токов для схемы с многополюсниками формируется так же, как для схемы с двухполюсниками, поэтому рассмотрим формирование матрицы узловых проводимостей $Y(\varphi)$. 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 $i_k = i_k(\varphi_1, \dots, \varphi_n)$. Пусть $p, r, \dots, j$ — номера полюсов многополюсника в схеме. Тогда матрица полюсных проводимостей многополюсника будет иметь вид

$$Y_{p-j} = \begin{array}{c|ccc} & p & r & j \\ \hline p & y_{pp} & y_{pr} & y_{pj} \\ r & -y_{rp} & y_{rr} & -y_{rj} \\ j & -y_{jp} & -y_{jr} & y_{jj} \end{array}$$

здесь $y_{pp}, y_{rr}, \dots, y_{jj}$ — собственные проводимости полюсов $p, r, \dots, j$, равные суммам проводимостей n -полюсника, инцидентных полюсам $p, r, \dots, j$ соответственно; $y_{pr}, y_{pj}, \dots$ и т. д. — взаимные проводимости между полюсами n -полюсника.

При формировании матрицы $Y(\varphi)$ для всей схемы используется так называемый *метод позиционного суммирования*. Этот метод заключается в том, что элемент Y_{ik} матрицы $Y(\varphi)$ получается между узлами i и k , в том числе той взаимной проводимости y_{ik} , которая входит в состав матрицы Y_{pj} n -полюсника, если i и k являются его полюсами. Аналогично, в состав элемента Y_{ii} матрицы $Y(\varphi)$ входит как слагаемое собственная проводимость y_{ii} n -полюсника, если i -й узел схемы одновременно является полюсом n -полюсника.

Пример 7.3. Любой биполярный транзистор можно рассматривать как трехполюсник с матрицей проводимостей

$$Y_{\text{тр}} = \begin{matrix} & \begin{matrix} \text{э} & \text{б} & \text{к} \end{matrix} \\ \begin{matrix} \text{э} \\ \text{б} \\ \text{к} \end{matrix} & \begin{vmatrix} y_{\text{ээ}} & -y_{\text{эб}} & -y_{\text{эк}} \\ -y_{\text{эб}} & y_{\text{бб}} & -y_{\text{кб}} \\ -y_{\text{эк}} & -y_{\text{кб}} & y_{\text{кк}} \end{vmatrix} \end{matrix}$$

Формулы для элементов этой матрицы $y_{\text{ээ}}$, $y_{\text{эб}}$ и т. д. различны и зависят от выбранной схемы замещения транзистора (см. гл. 3). Рассматривая транзистор в схеме на рис. 7.4,б как n -полюсник ($n=3$), можно сформировать матрицу узловых проводимостей $Y(\varphi)$ всей схемы в следующем виде:

$$Y(\varphi) = \begin{matrix} & \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{vmatrix} \frac{1}{R_1} + \frac{1}{R_2} + y_d & -y_d & 0 & 0 \\ -y_d & y_d + \frac{1}{R_6} + \frac{1}{R_3} + y_{\text{бб}} & -y_{\text{кб}} & -y_{\text{эб}} \\ 0 & -y_{\text{кб}} & \frac{1}{R_4} + y_{\text{кк}} & -y_{\text{эк}} \\ 0 & -y_{\text{эб}} & y_{\text{эк}} & y_{\text{ээ}} + \frac{1}{R_5} \end{vmatrix} \end{matrix}$$

Подводя итог рассмотренным методам формирования ММС в базе узловых потенциалов, можно сформулировать следующий алгоритм формирования ММС.

1. Для каждого элемента схемы вычисляется ток i_s (для двухполюсника) или вектор токов I_s (для многополюсника).

2. Для каждого элемента схемы вычисляется проводимость y_s (для двухполюсника) или матрица полюсных проводимостей Y_s (для многополюсника). Заметим, что двухполюсник, включенный между узлами m и n , можно описать как одной проводимостью ветви y , так и матрицей узловых проводимостей

$$Y_s = \begin{matrix} & \begin{matrix} m & n \end{matrix} \\ \begin{matrix} m \\ n \end{matrix} & \begin{vmatrix} y & -y \\ -y & y \end{vmatrix} \end{matrix} \quad (7.11)$$

3. Выделяются и обнуляются массивы Y и I для записи в них матрицы узловых проводимостей схемы Y и вектора узловых токов I , являющихся основными компонентами модели схемы.

4. Рассматривается очередной элемент схемы, начиная с первого. Его ток i_s или компоненты вектора I_s заносятся на соответствующие позиции массива I , а проводимость y_s или компоненты матрицы Y_s заносятся на соответствующие позиции матрицы Y , алгебраически суммируясь с содержимым этих позиций.

5. Если список элементов не исчерпан, то повторяется п. 4. Если исчерпан, то формирование модели схемы на этом заканчивается.

7.3.4. Представление элементов схемы в базисе узловых потенциалов. Для включения элемента в модель схемы в базисе узловых потенциалов необходимо, чтобы его уравнение имело вид

$$i = f(u). \quad (7.12)$$

Наиболее простым элементом является постоянный резистор R . Его уравнение $i = u/R$, поэтому вклад в вектор узловых токов I будет u/R , а в матрицу $Y - y = 1/R$. Аналогично вклад нелинейного двухполюсника с уравнением $i = f(u)$ в вектор I равен $f(u)$, а в матрицу $Y - y = di/du$.

Однако многие элементы схем описываются уравнениями отличного от $i = f(u)$ вида. Такие элементы считаются неудобными для составления модели схемы в базисе узловых потенциалов. К ним относятся идеальные источники тока и напряжения I и E , а также управляемые источники вида $i = f(i)$, $u = f(i)$, $u = f(u)$. Для неудобных элементов существует ряд приемов, позволяющих представить их уравнение в виде $i = f(u)$. Наиболее простым и универсальным является включение в ветвь с неудобным элементом дополнительных элементов — последовательных малых сопротивлений или параллельных малых проводимостей. Рассмотрим этот прием подробнее.

1. Идеальный источник тока I . Его неудобство состоит в том, что при соединении источников I в одном узле его узловая проводимость будет равна нулю, что приводит к вырождению матрицы узловых проводимостей в алгоритме (7.9). Включение параллельно источнику малой проводимости y (рис. 7.5,а) позволяет описать его уравнением требуемого вида (7.12):

$$i = I - yu. \quad (7.13)$$

2. Идеальный источник напряжения E . Включение последовательного малого сопротивления r (рис. 7.5,б) позволяет записать уравнение вида (7.12)

$$i = E/r - u/r. \quad (7.14)$$

Использование рассмотренных преобразований позволяет приводить к виду (7.12) уравнения любых управляемых источников. Обозначим переменные u , i управляющей ветви индексом 1, а управляемой ветви — индексом 2 и рассмотрим преобразования уравнений управляемых источников к виду (7.12).

3. Управляемый источник $i_2 = f(I_1)$. Представляя управляющую ветвь в виде (7.13) (рис. 7.5,б), получаем уравнение вида

$$i_2 = f(I_1 - y_1 u_1). \quad (7.15)$$

Если управляющая ветвь — не идеальный источник I_1 , а просто нелинейная ветвь $i_1 = f_1(u_1)$, то уравнение управляемого источника приобретает вид

$$i_2 = f(f_1(u_1)) \quad (7.16)$$

и никаких преобразований не требуется.

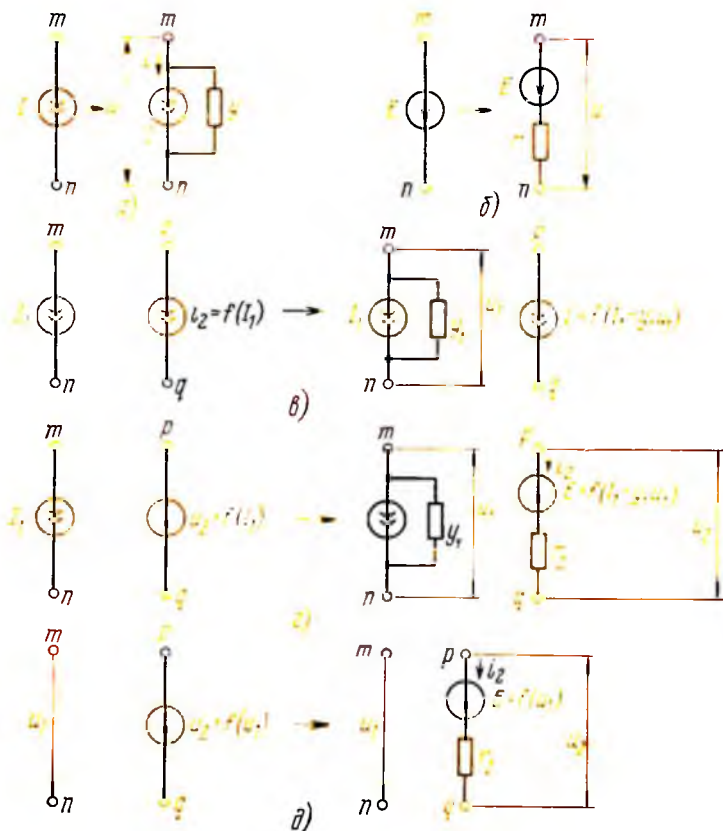


Рис. 7.5. Преобразования неудобных элементов для расчета схем в базисе узловых потенциалов

4. Управляемый источник $u_2 = f(I_1)$. Представляя управляющую ветвь в виде (7.13) (рис. 7.5,б), а управляемую — в виде (7.14), получаем

$$i_2 = [f(I_1 - y_1 u_1)]/r_2 - u_2/r_2. \quad (7.17)$$

5. Управляемый источник $u_2=f(u_1)$. Преобразуя управляемую ветвь к виду (7.14) (рис. 7.5,д), получаем

$$i_2=f(u_1)/r_2-u_2/r_2. \quad (7.18)$$

Выражения (7.15)—(7.18) позволяют легко найти матрицы узловых проводимостей преобразованных источников.

Как известно из теории цепей, для управляемого источника вида $i_2=f(u_1)$ матрица узловых проводимостей имеет вид

$$Y_a = p \begin{vmatrix} m & n \\ \bar{S} & -\bar{S} \\ q & -S & S \end{vmatrix} \quad (7.19)$$

где $S=di_2/du_1$ — крутизна источника; m и n — номера узлов управляющей ветви; p и q — номера узлов управляемой ветви, причем m и p — начальные, а n и q — конечные узлы. Как видим, на пересечении однонаправленных номеров узлов (оба начальные или оба конечные) крутизна записывается со знаком «+», а на пересечении разнонаправленных — со знаком «-». Это же правило действует и при составлении матриц Y_a других управляемых источников.

Рассмотрим каждый управляемый источник как четырехполюсник; учитывая (7.11) и (7.19), общую форму этих матриц можно представить в виде

$$Y_a = \begin{array}{c} \begin{matrix} m & n & p & q \end{matrix} \\ \begin{matrix} m \\ n \\ p \\ q \end{matrix} \begin{vmatrix} \bar{y}_1 & -\bar{y}_1 & & \\ -\bar{y}_1 & \bar{y}_1 & & \\ \bar{S} & -\bar{S} & \bar{y}_2 & -\bar{y}_2 \\ -\bar{S} & \bar{S} & -\bar{y}_2 & y_2 \end{vmatrix} \end{array} \quad (7.20)$$

Конкретный вид матрицы и содержание элементов $\bar{y}_1$, y_2 , S определяется типом источника и способом его преобразования к виду $i=f(u)$. Например, для источника $u_2=f(u_1)$ (рис. 7.5,д) значение $\bar{y}_1$ определяется проводимостью элементов, включенных между узлами m и n , $\bar{y}_2=1/r_2$, а $S=(df/du_1) \frac{1}{r_2}$. Для источника $i_2=f(I_1)$ (рис. 7.5,е) $\bar{y}_1=y_1$, $\bar{y}_2=0$, а $\bar{S}=di_2/du_1=df(I_1-y_1u_1)/du_1$.

7.4. МОДЕЛИРОВАНИЕ ПЕРЕХОДНЫХ ПРОЦЕССОВ

7.4.1. Явная форма модели. Наиболее распространены две формы представления исходной модели схемы для расчета переходных процессов — явная и неявная. Явная форма в общем случае состоит из двух подсистем

$$\frac{d\mathbf{u}_i}{dt} = F_{1i}(\mathbf{x}, \mathbf{v}, \mathbf{w}), \quad i = 1, \dots, n; \quad (7.21a)$$

$$F_{2j}(\mathbf{x}, \mathbf{v}, \mathbf{w}) = 0, \quad j = 1, \dots, p; \quad (7.21b)$$

здесь $\mathbf{x}$ — вектор переменных u_C, i_L реактивных элементов, называемых обычно переменными состояния; $\mathbf{v}$ — вектор постоянных и времязависимых источников E, I ; $\mathbf{w}$ — вектор переменных u, i линейных и нелинейных резистивных элементов.

Подсистема (7.21a) — это система обыкновенных дифференциальных уравнений (ОДУ) первого порядка, разрешенных относительно производных. Такая система называется *нормальной системой* ОДУ.

Подсистема (7.21b) — это система конечных, в общем случае нелинейных уравнений.

Пример 7.4. Модель схемы на рис. 7.6,а в явной форме имеет вид

$$\frac{du_C}{dt} = \frac{1}{C} \left(\frac{E - u_C - u_D}{R_1} + \frac{E - u_C - u_D}{R_2} \right), \quad (7.22a)$$

$$\left. \begin{aligned} I_0 (e^{u_D / U_0} - 1) - \frac{u_D}{R_1} - \frac{u_D}{R_2} &= 0 \\ E - u_C - u_D - U_{R_1} &= 0 \\ E - u_C - u_D - U_{R_2} &= 0 \end{aligned} \right\} \quad (7.22b)$$

Первое уравнение — дифференциальное, а три последних образуют нелинейную систему конечных уравнений.

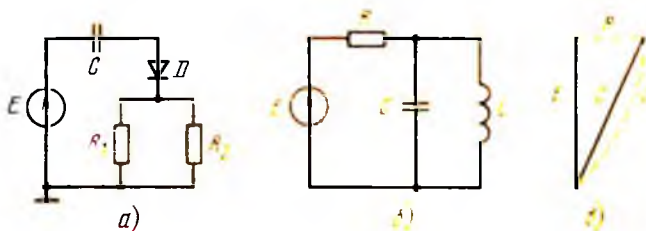


Рис. 7.6. Нелинейная RC-схема (а), линейная RLC-схема (б) и ее нормальное дерево (в)

Представление модели схемы в явной форме налагает ограничения на выбор независимых переменных — ими могут быть только напряжения на емкостях u_C (или заряды емкостей $Q = Cu_C$) и токи в индуктивностях i_L (или потокосцепления индуктивностей $\psi = Li_L$).

Пример 7.5. Если в схеме на рис. 7.6,б выбирать в качестве независимых переменных узловые потенциалы, получим уравнение

$$\frac{E - \varphi_1(t)}{R} - C \frac{d\varphi_1(t)}{dt} - \frac{1}{L} \int \varphi_1(t) dt = 0,$$

а если контурные токи, то уравнения

$$E - j_1(t)R - \frac{1}{C} \int (j_1(t) - j_2(t)) dt = 0,$$

$$\frac{1}{C} \int (j_1(t) - j_2(t)) dt + L \frac{dj_2(t)}{dt} = 0.$$

В обоих случаях имеем интегродифференциальные уравнения, а не ОДУ в нормальной форме. Если же выбрать в качестве независимых переменных ток i_L и напряжение u_C , то получим ОДУ в нормальной форме

$$\frac{du_C(t)}{dt} = \frac{1}{C} \left(\frac{E - u_C(t)}{R} - i_L(t) \right),$$

$$\frac{di_L(t)}{dt} = \frac{u_C(t)}{L}.$$

Заметим, что в частных случаях для получения нормальной системы ОДУ можно использовать не только переменные состояния, но и другие переменные. Например, когда в схеме нет индуктивностей, можно использовать узловые потенциалы.

Рассмотрим методику получения нормальной формы ОДУ. В общем случае она состоит из двух этапов. Пусть имеется система топологических и компонентных уравнений, причем компонентными уравнениями реактивных элементов служат дифференциальные зависимости

$$i_C(t) = C \frac{du_C(t)}{dt}, \quad u_L(t) = L \frac{di_L(t)}{dt}. \quad (7.23)$$

Тогда первый этап методики заключается в том, что на основе законов Кирхгофа переменные i_C , u_L выражаются через токи и напряжения остальных ветвей;

$$i_C = f_1(\text{остальные переменные}); \quad (7.24a)$$

$i_R, i_E, i_L, \dots$

$$u_L = f_2(\text{остальные переменные}), \quad (7.24b)$$

$u_R, u_I, u_C, \dots$

а на втором этапе i_C и u_L заменяются производными в соответствии с (7.23), а остальные переменные с помощью комбинирования законов Кирхгофа и компонентных уравнений выражаются через u_C и i_L . В результате вместо системы (7.24) получим нормальную систему

$$\frac{du_C}{dt} = f_1(u_C(t), i_L(t), E, I); \quad (7.25a)$$

$$\frac{di_L}{dt} = f_2(u_C(t), i_L(t), E, I). \quad (7.25b)$$

На первом этапе выполняются следующие действия:

составляется топологический граф схемы, каждой ветви которого задается произвольное направление;

строится так называемое *нормальное дерево* графа; для этого в состав дерева последовательно включаются все независимые и зависимые источники напряжения E , $u(i)$, затем емкости C , сопротивлений R и, если необходимо, индуктивности L ;

для построенного нормального дерева составляется топологическая система всех контурных уравнений и уравнений сечений;

из полученной системы выделяется подсистема вида (7.24), включающая контурные уравнения, образованные индуктивными хордами, и уравнения сечений, образованные емкостными ветвями.

Напомним, что *ветвями* называются ребра графа, вошедшие в это дерево, а *хордами* — ребра, не вошедшие в дерево.

На втором этапе с помощью оставшихся топологических уравнений и компонентных уравнений из системы вида (7.24) окончательно формируется нормальная система ОДУ.

Проиллюстрируем эту методику простым примером составления уравнений для схемы на рис. 7.6,б. Нормальное дерево графа этой схемы изображено на рис. 7.6,в, система топологических уравнений имеет вид

$$\begin{aligned} 1) \quad i_E - i_R &= 0, & 2) \quad i_C + i_L - i_R &= 0, \\ 3) \quad -E + u_R + u_C &= 0, & 4) \quad u_L - u_C &= 0, \end{aligned}$$

подсистема вида (7.24) состоит из уравнений 2) и 4):

$$i_C = i_R - i_L, \quad u_L = u_C.$$

Используя компонентные уравнения $u_R = Ri_R$, $i_C = Cdu_C/dt$, $u_L = Ldi_L/dt$ и одно из оставшихся топологических уравнений — 3), получаем нормальную систему ОДУ, приведенную в примере 7.5.

В более сложных случаях, когда в схеме имеются контуры из емкостей или звезды из индуктивностей, процедура формирования нормальной системы ОДУ усложняется. Причина усложнения заключается в следующем.

Из теории цепей известно, что переходные процессы в любой электрической схеме, содержащей реактивные элементы, описываются нормальной системой ОДУ, содержащей n линейно-независимых уравнений, где n — количество реактивных элементов с независимыми начальными запасами энергии — емкостей с зарядами Q и индуктивностей с потокоцеплениями ψ . Если же в схеме есть контур из емкостей, то напряжение, а значит и заряд, а одной из них уже не является независимым, поскольку определяется в данный момент времени напряжениями на других емкостях этого контура по второму закону Кирхгофа. Следовательно, одна из емкостей контура не может быть использована для получения независимого дифференциального уравнения.

Формально этот факт проявляется в описанной выше методике тем, что при построении нормального дерева одна из емкостей контура войдет в число хорд и не даст соответствующего уравнения в составе системы (7.24). Чтобы выразить теперь ток этой емкости через напряжения других емкостей, приходится решать

дополнительные системы уравнений. То же можно сказать о случае звезды из индуктивностей.

Емкостные контуры и индуктивные звезды принято называть *топологическими вырождениями*. Таким образом, порядок n нормальной системы ОДУ (7.21a) для схемы, содержащей n_C емкостей, n_L индуктивностей и n_T топологических вырождений, будет не $n = n_C + n_L$, а $n = n_C + n_L - n_T$.

Пример 7.6. Схема на рис. 7.7 содержит шесть реактивных элементов но описывается нормальной системой из четырех дифференциальных уравнений так как содержит один емкостный контур C_1, C_2, C_3 и одну звезду индуктивностей L_1, L_2, L_3 .

Чтобы сохранить простоту формирования нормальной системы ОДУ для схемы с топологическими вырождениями, можно эти вырождения устранить включением в емкостный контур небольшого сопротивления R или параллельно одной из индуктивностей звезды — малой проводимости G . Тогда в емкостном контуре все напряжения u_C можно считать независимыми, так же как токи i в звезде индуктивностей.

Погрешность расчета от этого приема при малых R и G будет невелика, однако следует иметь в виду, что введение малых R

G приводит к появлению малых постоянных времен схемы, а это, как будет видно из дальнейшего, может увеличить время численного расчета. Методика составления нормальной системы ОДУ также усложняется если в схеме есть контуры



Рис. 7.7. Схема с топологическими вырождениями

или звезды из нелинейных резистивных элементов, так как в этом случае выражение их токов и напряжения через u_C, i требует решения дополнительных систем конечных уравнений (7.21б), общем случае нелинейных (см. пример 7.4).

Рассмотренный способ формирования модели схемы получил в литературе название *метода переменных состояния*.

7.4.2. Организация расчета модели схемы в явной форме. Перейдем к изучению организации вычислений и численных методов для решения системы (7.21).

Прежде всего отметим, что расчет переходных процессов с помощью модели схемы в явной форме (7.21) требует решения двух принципиально различных по характеру систем уравнений: системы конечных уравнений (7.21б) для расчета квазистатического режима и системы дифференциальных уравнений (7.21a) для расчета переменных состояния u_C, i_L . Расчет квазистатического режима может быть осуществлен любым численным методом решения конечных уравнений — Ньютона, Бroyдена, простых итераций и т. д.

Расчет переменных состояния по (7.21а) может быть выполнен любым численным методом решения ОДУ.

Организация вычислений для системы (7.21) заключается в поочередном решении систем (7.21а) и (7.21б). Пусть из расчета статического режима известны начальные значения u_{c0} и i_{L0} переменных состояния и w_0 резистивных переменных. Тогда:

1) подставляя u_{c0} , i_{L0} , w_0 в систему ОДУ (7.21а) и решая ее любым численным методом, определяем значения u_{c1} , i_{L1} ;

2) считая u_{c1} , i_{L1} постоянными источниками напряжения и тока, рассчитываем квазистатический режим схемы, т. е. решаем систему конечных уравнений (7.21б) и определяем вектор $w_1 = w(t_1)$ токов и напряжений нелинейных и резистивных элементов. После этого опять выполняем п. 1.

Пример 7.7. Рассмотрим систему (7.22). Пусть известны u_{c0} , u_{D0} . Тогда из решения (7.22а) найдем u_{c1} . Подставляя u_{c1} в (7.22б) и решая эту систему, находим u_{D1} , u_{R1} , u_{R2} . Таким образом, имеем значения u_{c1} , u_{D1} , после чего можем решать снова уравнение (7.22а).

Система (7.21б) в общем случае состоит как из линейных, так и из нелинейных уравнений. Однако если все нелинейные элементы описываются уравнениями вида $w_n = f(u_c, i_L)$, то система (7.21б) может быть только линейной. Действительно, задав или вычислив значения u_c , i_L , можно по указанным уравнениям сразу найти составляющие вектора w , относящиеся к нелинейным элементам, без применения какого-либо численного метода. Наиболее широко распространен случай $i_n = f(u_c)$, так как большинство схем замещения нелинейных компонентов содержит емкости на входе и выходе. Естественно, если в системе (7.21б) после расчета нелинейных компонентов остаются какие-либо уравнения, то они могут быть только линейными.

7.4.3. Основные численные методы решения системы ОДУ. Методы решения системы (7.21а) рассмотрим на примере одностепенного уравнения

$$\frac{dx(t)}{dt} = f(x(t), t), \quad (7.26)$$

поскольку решение системы ОДУ (7.21а) аналогично. Задача расчета переходного процесса в схеме по уравнению (7.26) математически формулируется как задача Коши, заключающаяся в том, что ищется решение $x(t)$ уравнения (7.26), удовлетворяющее заданному начальному условию $x(t_0) = x_0$ на интервале $t_0 \leq t \leq t_k$, причём конец интервала t_k заранее может быть и неизвестен.

При решении численными методами искомая функция $x(t)$ определяется в отдельных точках интервала $t_1, t_2, \dots, t_n$, называемых узлами, в виде таблицы значений $x_1, x_2, \dots, x_n$, приближенно равных значениям $x(t_1), x(t_2), \dots, x(t_n)$ точного решения $x(t)$. Расстояние между узлами $\Delta t = t_{n+1} - t_n$ называется шагом интегрирования h и может либо быть задано перед началом вычислений (интегрирование с постоянным шагом), либо определяться р

цессе вычислений (интегрирование с автоматическим выбором шага). Значения искомой функции в двух соседних узлах связаны соотношением

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(x(t), t) dt. \quad (7.27)$$

Простейшими методами решения уравнения (7.26) являются: явный метод Эйлера $x_{n+1} = x_n + hf(x_n, t_n)$; (7.28)

метод трапеций $x_{n+1} = x_n + \frac{h}{2} [f(x_{n+1}, t_{n+1}) + f(x_n, t_n)]$; (7.29)

неявный метод Эйлера $x_{n+1} = x_n + hf(x_{n+1}, t_{n+1})$. (7.30)

Все три метода являются частными случаями обобщенного двухточечного метода

$$x_{n+1} = x_n + h[a f(x_n, t_n) + (1-a) f(x_{n+1}, t_{n+1})], \quad (7.31)$$

из которого они получаются при $a=1, 1/2, 0$ соответственно.

В более общем виде линейный m -шаговый алгоритм решения уравнения (7.26) можно представить как

$$x_{n+1} = \sum_{p=n-m}^n \alpha_p x_p + \sum_{p=n-m}^n \beta_p f(x_p, t_p) + \gamma f(x_{n+1}, t_{n+1}). \quad (7.32)$$

При $\gamma=0$ получаем явные методы, так как для определения x_{n+1} достаточно знать значения x_{n-j} , вычисленные в предыдущие моменты времени. При $\gamma \neq 0$ получаем неявные методы, так как, чтобы найти x_{n+1} , уравнение (7.32) на каждом шаге нужно решить относительно этой переменной, причем вычисление функций $f(x, t)$ требует полного расчета всей схемы. Коэффициенты $\alpha_p, \beta_p, \gamma$ в (7.32) определяются методом неопределенных коэффициентов, при этом нужно учитывать, что $f(x, t)$ есть производные от x .

Главное отличие явных методов от неявных состоит в том, что явные методы всегда имеют ограничение на шаг интегрирования ОДУ $h \leq h_{\max}$ и превышение $h_{\max}$ приводит к потере устойчивости решения, т. е. неограниченному росту ошибки по мере продолжения решения. Неявные методы таких ограничений или вообще не имеют, или эти ограничения значительно слабее.

Пример 7.8. Переходный процесс в простейшей RC-цепи, в которую подан сигнал $E(t)$, описывается дифференциальным уравнением

$$\frac{du_C(t)}{dt} = \frac{E - u_C(t)}{RC}$$

Положим для простоты $E=R=C=1$, тогда явный метод Эйлера для этого уравнения будет иметь вид

$$u_{C, n+1} = u_{C, n} + h(1 - u_{C, n}),$$

а неявный метод — вид

$$u_{C, n+1} = u_{C, n} + h(1 - u_{C, n+1});$$

отсюда

$$u_{C, n+1} = (u_{C, n} + h)/(1 + h).$$

Значения $u_{c, n+1}$, вычисленные обоими методами для разных значений шага h , приведены в табл. 7.1.

Таблица 7.1

Метод	Значение шага	Номер шага				
		0	1	2	3	4
Явный метод Эйлера	0,5	0	0,5	0,75	0,875	0,9375
	1	0	1	1	1	1
	2	0	2	0	2	0
	3	0	3	-3	9	-15
Неявный метод Эйлера	1	0	0,5	0,75	0,875	0,9375
	9	0	0,9	0,99	0,999	0,9999
	99	0	0,99	0,999	0,9999	0,99999

Видно, что явный метод Эйлера уже при $h > 2$ теряет устойчивость, тогда как неявный метод устойчив даже при $h = 99$. Можно показать теоретически, что явный метод Эйлера устойчив при $h < T_{\text{мин}}$, где $T_{\text{мин}}$ — минимальная постоянная времени схемы, а неявный метод Эйлера сохраняет устойчивость при любом шаге h .

7.4.4. Неявная форма математической модели. Вторая форма представления динамической модели схемы — неявная. В общем случае она имеет вид

$$F_i \left(\frac{dx(t)}{dt}, \int x(t) dt, x(t) \right) = 0, \quad i = 1, \dots, p. \quad (7.33)$$

Далее вместо производных и интегралов подставляются их конечно-разностные аппроксимации, соответствующие тем или иным формулам численного дифференцирования и интегрирования:

$$dx/dt \approx j_{\Delta}(x_{n+1}, x_n, \dots, x_{n-k}); \quad (7.34a)$$

$$\int_a^b x(t) dt \approx f_{\Delta}(x_{n+1}, x_n, \dots, x_{n-k}). \quad (7.34б)$$

Таким образом, от исходной системы (7.33) переходим к системе конечно-разностных алгебраических уравнений

$$F_i(x_{n+1}, x_n, \dots, x_{n-k}) = 0, \quad i = 1, \dots, p. \quad (7.35)$$

Замена производных и интегралов по (7.34) аппроксимирующими конечно-разностными формулами называется *дискретизацией компонентных уравнений* емкости и индуктивности, а подстановка этих формул в (7.33) и получение системы (7.35), не содержащей производных, называется *алгебраизацией*. Таким образом, *дискретизация* и *алгебраизация* составляют существо построения динамической модели в неявной форме.

Составленная модель (7.35) далее решается относительно x_{n+1} каким-либо численным методом решения конечных уравнений, обычно методом Ньютона

$$F'(x_{n+1}^{(k)}, x_n, \dots, x_{n-k}) \Delta x_{n+1}^{(k)} = -F(x_{n+1}^{(k)}, x_n, \dots, x_{n-k}), \quad (7.36)$$

где $\Delta x_{n+1}^{(k)} = x_{n+1}^{(k+1)} - x_{n+1}^{(k)}$ — вектор поправок; $F'(\cdot)$ — матрица Якоби; k — индекс ньютоновских итераций.

После вычисления с заданной точностью значений x_{n+1} полагаем $x_n = x_{n+1}$, $x_{n-1} = x_n$ и т. д., т. е. сдвигаем рассчитываемые точки на один шаг по оси автоматного времени (см. гл. 1) и снова решаем (7.36) относительно нового значения x_{n+1} . Этот процесс повторяется до тех пор, пока не будет пройден заданный интервал времени t .

7.4.5. Особенности неявной формы модели. Расчет модели, заданной в неявной форме, имеет ряд особенностей.

Первая особенность состоит в том, что отсутствует численный метод решения ОДУ в виде одной из формул (7.28)—(7.32). Метод решения интегродифференциального уравнения (7.33) определяется формулами дискретизации (7.34). Так, использование

$$dx/dt = x' \approx (x_{n+1} - x_n) / \Delta t \quad (7.37)$$

соответствует решению ОДУ методом Эйлера, а использование формул

$$x'_{n+1} = (x_{n+1} - x_n) / (\Delta t/2) - x'_n \quad (7.38)$$

— методом трапеций.

Более того, формулы дискретизации (7.34) можно не связывать с каким-либо конкретным численным методом решения ОДУ, а рассматривать только как соотношения, связывающие предыдущие известные значения x_{n-k} , x_{n-k+1} , ..., x_n с неизвестным последующим значением x_{n+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.

Вторая особенность решения (7.33) заключается в тесной связи способа дискретизации (7.34) с типом метода решения (7.33) — явным или неявным. Дискретизация производных по так называемым формулам дифференцирования назад (только через предыдущие значения)

$$x'_{n+1} = \sum_{i=1}^{n+1} a_i x_i + \sum_{i=1}^n b_i x'_i, \quad (7.39)$$

а также дискретизация интегралов по формулам численного интегрирования замкнутого типа (с включением концов интервала интегрирования)

$$\int_{t_1}^{t_{n+1}} x(t) dt = \sum_{i=1}^{n+1} A_i x_i \quad (7.40)$$

соответствует неявным методам решения ОДУ, а дискретизация производных и интегралов по формулам дифференцирования вперед

$$x'_n = \sum_{i=1}^{n+1} a_i x_i + \sum_{i=1}^{n-1} b_i x'_i \quad (7.41)$$

и по формулам численного интегрирования открытого или полуоткрытого типа

$$\int_{t_1}^{t_{n+1}} x(t) dt = \sum_{i=1}^n A_i x_i \quad (7.42)$$

соответствует явным методам решения ОДУ. В первом случае для вычисления x_{n+1} нужно решать (7.35) относительно x_{n+1} , например с помощью (7.36), а во втором — структура (7.35) получится такой, что из нее можно выразить x_{n+1} явно, в виде формулы, без решения уравнения (7.35).

Коэффициенты a_i , b_i , A_i , B_i формул (7.39) — (7.42) определяют методом неопределенных коэффициентов.

Третья особенность заключается в том, что система (7.33) смешанная. Она может содержать уравнения без производных или без интегралов, или без тех и других, т. е. конечные уравнения. Это зависит от типа компонентов, входящих в тот контур или соединенных в том узле, для которого составляется уравнение Кирхгофа. Таким образом, в отличие от (7.21), где выделены две подсистемы уравнений — дифференциальная и конечная, здесь такого разделения нет.

Наконец, *четвертая* особенность состоит в том, что система (7.33) не накладывает ограничений на тип переменных. В отличие от явной формы (7.21), где независимыми переменными могут быть только переменные состояния u_C , i_L , при составлении системы (7.33) в качестве независимых переменных можно выбирать узловые потенциалы, контурные токи, токи или напряжения ветвей и т. д.

Пример 7.9. Для схемы на рис. 7.6,б компонентные уравнения C и L дискретизируем по (7.39), (7.40):

$$i_{C, n+1} = C \frac{du_C(t)}{dt} \approx C \frac{u_{C, n+1} - u_{C, n}}{\Delta t}; \quad (7.43a)$$

$$i_{L, n+1} = i_{L, n} + \frac{1}{L} \int_{t_n}^{t_{n+1}} u_L(t) dt \approx i_{L, n} + \frac{\Delta t}{L} u_{L, n+1}. \quad (7.43б)$$

Заменяя напряжения через разности потенциалов ($u_C = \varphi_1$, $u_L = \varphi_1$), получаем уравнение в базисе потенциалов в форме (7.35)

$$i_{R, n+1}(\varphi_1) - i_{L, n+1}(\varphi_1) - i_{C, n+1}(\varphi_1) = 0,$$

откуда

$$\frac{E - \varphi_{1, n+1}}{R} - \left(\frac{\Delta t}{L} \varphi_{1, n+1} + i_{L, n} \right) - \frac{C}{\Delta t} (\varphi_{1, n+1} - \varphi_{1, n}) = 0.$$

Поскольку в схеме нет нелинейных элементов, то уравнение получилось линейным и его можно решать не только методом Ньютона, но и аналитически, однако в обоих случаях вследствие принятых формул дискретизации решение этого уравнения соответствует неявному методу.

Если хотим «погрузить» в уравнения схемы явный метод, то дискретизацию нужно выполнять по (7.41) и (7.42):

$$i_{C,n} = C \left. \frac{du_C}{dt} \right|_{t=t_n} \approx C \frac{u_{C,n+1} - u_{C,n}}{\Delta t}; \quad (7.44a)$$

$$i_{L,n} = i_{L,n-1} + \frac{1}{L} \int_{t_{n-1}}^{t_n} u_L(t) dt \approx i_{L,n-1} + \frac{\Delta t}{L} u_{L,n-1}. \quad (7.44b)$$

Тогда уравнение узловых потенциалов примет вид

$$i_{R,n} - i_{L,n} - i_{C,n} = 0,$$

откуда

$$\frac{E - \Phi_{1,n}}{R} - \left(i_{L,n-1} + \frac{\Delta t}{L} \Phi_{1,n-1} \right) - C \frac{\Phi_{1,n+1} - \Phi_{1,n}}{\Delta t} = 0.$$

Заметим, что значения $i_{L,n}$ в неявном методе и значения $i_{L,n-1}$, $\Phi_{1,n-1}$, $\Phi_{1,n}$ в явном методе известны, так как вычисляются на предыдущих шагах.

Как видно из приведенного примера, при реализации неявных методов уравнения равновесия схемы составляются для следующего $(n+1)$ -го момента времени. Это требует вычисления резистивных переменных i_R , u_R тоже для $(n+1)$ -го момента времени, тогда как при реализации явных методов уравнения равновесия составляются для данного n -го момента времени и соответственно для этого же момента должны быть вычислены резистивные переменные. Данное положение является определяющим при организации вычислений для расчета переходных процессов явным или неявным численным методом решения ОДУ. Если реализуется явный метод, то независимо от формы модели схемы — явной или неявной — расчет должен начинаться с вычисления резистивных переменных, в том числе нелинейных, после чего вычисляются реактивные переменные, связанные с L и C для требуемых моментов времени.

Если реализуется неявный метод, то резистивные и реактивные переменные вычисляются на каждом шаге одновременно. Из предыдущего должно быть ясно, что явная форма модели схемы (7.21), где имеются отдельные подсистемы для реактивных и резистивных переменных, лучше приспособлена для реализации явных методов решения ОДУ, тогда как неявная форма (7.33), где такого разделения нет, — для неявных методов. Вместе с тем в каждой из этих форм можно реализовать как явные, так и неявные методы.

7.4.6. Расчет неявной формы модели схемы в базисе узловых потенциалов. Поскольку в базисе узловых потенциалов уравнения компонентов должны иметь вид $i = f(u)$, то для реактивных ком-

понентов — емкости и индуктивности — используются компонентные уравнения вида

$$i_C = C \frac{du_C}{dt}, \quad i_L = \frac{1}{L} \int u_L dt. \quad (7.45)$$

Далее в соответствии с общей методикой формулы (7.45) дискретизируются. Простейшая дискретизация имеет вид (7.43). Ей можно поставить в соответствие схемы замещения емкости и индуктивности на рис. 7.8, где компоненты $\Delta t/C$, $L/\Delta t$ играют роль фиктивных резисторов, поскольку имеют соответствующую резисторам размерность. В более общем случае дискретизированные формулы (7.45) можно представить в следующем виде:

$$i_{C, n+1} = \frac{C}{\Delta t} u_{C, n+1} + f(u_{C, n}, u_{C, n-1}, \dots, u_{C, n-k}) = y_C u_{C, n+1} + I_C; \quad (7.46a)$$

$$i_{L, n+1} = f(i_{L, n}, i_{L, n-1}, \dots, i_{L, n-k}) + \frac{\Delta t}{L} u_{L, n+1} - I_L = y_L u_{L, n+1} + I_L. \quad (7.46b)$$

и составить соответствующие схемы замещения.

Полученные дискретизации легко вписываются в общую методику составления и расчета модели схемы в базисе узловых потенциалов. В этом базисе теоретическая модель схемы, соответствующая алгоритму (7.36), имеет вид

$$Y(\Phi_{n+1}^{(k)}) \Delta \Phi_{n+1}^{(k)} = -I(\Phi_{n+1}^{(k)}, \Phi_n, \dots, \Phi_{n-k}). \quad (7.47)$$

При формировании вектора токов I в (7.47) каждое из уравнений (7.43) или (7.46) рассматривается как уравнение тока соответствующей ветви. При этом напряжения $u_{C, n+1}$, $u_{L, n+1}$ заменяются через разности потенциалов, а значения $u_{C, n}$, $u_{C, n-1}$, $u_{C, n-k}$, $i_{L, n}$, $i_{L, n-1}$, $i_{L, n-k}$ предполагаются известными из предыдущих расчетов или начальных условий.

При формировании матрицы узловых проводимостей Y вклад каждой емкостной ветви равен проводимости $y_C = C/\Delta t$, а вклад каждой индуктивной ветви — проводимости $y_L = \Delta t/L$ с соответствующими знаками.

Отметим, что при расчете переходных процессов с автоматическим выбором шага решения Δt проводимости реактивных ветвей нужно пересчитывать при каждом изменении шага Δt .

Таким образом, в базисе узловых потенциалов формирование модели схемы (7.47) для расчета переходных процессов не отличается от формирования модели (7.9) для расчета статического ре-

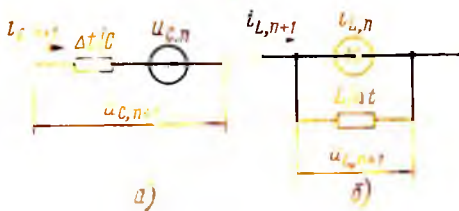


Рис. 7.8. Дискретные схемы замещения для емкости (а) и индуктивности (б)

жима. Это важное достоинство базиса узловых потенциалов значительно упрощает программы схемотехнического моделирования.

Пример 7.10. Используя дискретизацию уравнений L и C по (7.43) и применяя описанную методику формирования динамической модели для схемы на рис. 7.9, получим компоненты I и Y этой модели в следующем виде:

$$I(\varphi) = \begin{bmatrix} -\frac{E - \varphi_{1,n+1}}{R_1} + C_1 \frac{(\varphi_{1,n+1} - \varphi_{2,n+1}) - u_{C2,n}}{\Delta t} + \\ + C_2 \frac{(\varphi_{1,n+1} - \varphi_{2,n+1}) - u_{C2,n}}{\Delta t} + \frac{\varphi_{1,n+1} - \varphi_{2,n+1}}{R_2} \\ - \frac{\varphi_{1,n+1} - \varphi_{2,n+1}}{R_2} - C_2 \frac{(\varphi_{1,n+1} - \varphi_{2,n+1}) - u_{C2,n}}{\Delta t} + \\ + \frac{\Delta t}{L} \varphi_{2,n+1} + i_{L,n} \end{bmatrix}$$

$$Y = \begin{bmatrix} \frac{1}{R_1} + \frac{C_1}{\Delta t} + \frac{C_2}{\Delta t} + \frac{1}{R_2} & -\frac{C_2}{\Delta t} - \frac{1}{R_2} \\ -\frac{C_1}{\Delta t} - \frac{1}{R_2} & \frac{1}{R_1} + \frac{C_2}{\Delta t} + \frac{\Delta t}{L} \end{bmatrix}$$

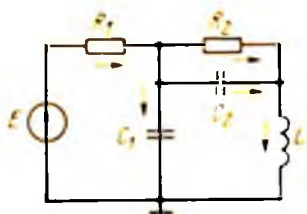


Рис. 7.9. RLC-схема

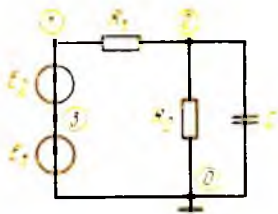


Рис. 7.10. RC-схема

7.5. МОДЕЛИРОВАНИЕ ЧАСТОТНЫХ ХАРАКТЕРИСТИК

Амплитудно-частотная характеристика (АЧХ) определяется как отношение

$$F(j\omega) = u_{\text{вых}}(j\omega) / u_{\text{вх}}(j\omega),$$

где $j(\omega)$ — комплексная частота.

Существуют различные способы расчета АЧХ. Аналитические способы основаны на вычислении АЧХ как отношения полиномов:

$$F(j\omega) = \frac{M(j\omega)}{N(j\omega)} = \frac{a_m(j\omega)^m + a_{m-1}(j\omega)^{m-1} + \dots + a_0}{b_n(j\omega)^n + b_{n-1}(j\omega)^{n-1} + \dots + b_0}$$

Моделирование АЧХ на ЭВМ основано на трех подходах: символьном, численно-символьном и численном. При символьном под-

ходе программы составляются так, чтобы вычислить коэффициенты a_i , b_i в виде формул. Этот подход применяется при расчете АЧХ небольших схем (10 ... 20 компонентов), так как с увеличением размера схем объем вычислений быстро возрастает. Аналогичный недостаток имеет численно-символьный подход, когда коэффициенты a_i , b_i вычисляются в виде чисел, но АЧХ, как и ранее, представляется через отношение полиномов. В современных программах расчета АЧХ средних и больших схем наибольшее распространение получил численный подход, когда АЧХ вычисляется как численное значение $F(j\omega)$ при разных значениях ω , т. е. поточечно. Рассмотрим этот подход применительно к базису узловых потенциалов.

В качестве входного сигнала используется источник напряжения $E_{вх}(j\omega) = 1 \cdot e^{j\omega}$ с единичной комплексной амплитудой, нулевой начальной фазой и малым внутренним сопротивлением. В базисе узловых потенциалов этот источник напряжения преобразуется в единичный источник тока $I_{вх}(j\omega) = 1 \cdot e^{j\omega}$ с большой параллельно включенной проводимостью. Таким образом, при указанном входном сигнале будет верно соотношение

$$F(j\omega) = u_{вых}(j\omega)/E_{вх}(j\omega) = u_{вых}(j\omega),$$

где $F(j\omega)$ — комплексная АЧХ. Следовательно, рассчитывая $u_{вых}(j\omega)$ в любой ветви схемы, мы тем самым рассчитываем АЧХ схемы для этой ветви. Если один из полюсов ветви с входным сигналом — земляной, то $E_{вх} = \varphi_{вх}$; тогда вместо $u_{вых}(j\omega)$ можно рассчитать $\varphi_{вых}(j\omega)$.

Метод узловых потенциалов позволяет легко формировать узловые уравнения не только для временной, но и для частотной области. В этом случае вся предыдущая методика формирования узловых уравнений сохраняется, изменяются лишь компонентные уравнения реактивных ветвей, принимающие следующий вид:

$$i_C = j\omega C (\varphi_{нач} - \varphi_{кон}), \quad i_L = \frac{1}{\omega L} (\varphi_{нач} - \varphi_{кон}), \quad (7.48)$$

где $\varphi_{нач}$, $\varphi_{кон}$ — потенциалы на концах реактивных ветвей; j — мнимая единица; ω — частота.

Соответственно проводимости реактивных ветвей равны

$$y_C = j\omega C, \quad y_L = -j/(\omega L). \quad (7.49)$$

Уравнения (7.48) используются при формировании вектора узловых токов, а (7.49) — матрицы узловых проводимостей. При этом в схеме замещения постоянные источники напряжения E закорачиваются, а постоянные источники тока I размыкаются. В результате получим узловое уравнение линейной схемы в частотной области

$$Y(j\omega)\varphi(j\omega) = -I(j\omega). \quad (7.50)$$

Здесь в векторе $I(j\omega)$ составляющая, соответствующая входной ветви, равна 1, а остальные составляющие — нулевые, так как при

нулевых начальных приближениях в линейных схемах все токи ветвей равны нулю.

В отличие от уравнения (7.47) для временной области, которое в каждый момент времени нужно решать несколько раз до сходимости, уравнение (7.50) на каждой частоте ω нужно решать лишь один раз, поскольку схема линейная. Решение уравнения (7.50) можно выполнить, используя стандартную программу либо обращения комплексной матрицы, либо решения системы линейных уравнений с комплексными коэффициентами. Подставляя в (7.50) разные значения частоты ω_i и вычисляя вектор $\Phi(j\omega_i)$ на каждой частоте, получаем комплексную частотную характеристику рассчитываемого устройства.

Если интерес представляет частотная характеристика в каком-нибудь одном узле схемы k , то на каждой частоте ω_i нужно выбирать из вектора $\Phi(j\omega_i)$ комплексное значение потенциала $\Phi_k(j\omega_i) = A_k(\omega_i) + jB_k(\omega_i)$ и вычислять точку амплитудно-частотной и фазо-частотной характеристики в узле k :

$$\Phi_{k \text{ ачх}}(\omega_i) = \sqrt{A_k^2(\omega_i) + B_k^2(\omega_i)}, \quad \Phi_{k \text{ фчх}}(\omega_i) = \operatorname{arctg}[B_k'(\omega_i), A_k(\omega_i)].$$

7.6. ВХОДНЫЕ ЯЗЫКИ СХЕМОТЕХНИЧЕСКОГО МОДЕЛИРОВАНИЯ

Входной язык программ СхМ включает следующие основные части:

язык описания схемы;

язык описания задания;

язык управления работой программы (режимами работы, печатью результатов и т. д.).

7.6.1. Язык описания схемы. Описание принципиальной схемы сводится к описанию ее элементов (компонентная информация) и связей между ними (топологическая информация). Обычно описание схемы состоит из построчного описания типа, параметров и связей каждого элемента. Каждая строка описания содержит сведения о типе элемента, о значениях параметров его модели, используемой в программе, о номерах полюсов, которыми элемент включен в программу. В разных языках последовательность указания этих сведений различна. Приведем несколько примеров описания элементов.

$R5(1,5)=10 \text{ Ом}$ — резистор $R5$, имеющий номинал 10 Ом, включен между первым и пятым узлом;

$E1(2, 3)=\text{ТРАП}$ ($A=5$, $t_0=1$, $t_1=2$, $t_2=4$, $t_3=5$) — источник напряжения трапецевидной формы с параметрами — амплитудой $A=5 \text{ В}$, фронтом между точками $t_0=1 \text{ мкс}$, $t_1=2 \text{ мкс}$ срезом между точками $t_2=4 \text{ мкс}$ и $t_3=5 \text{ мкс}$, включенный между узлами 2 и 3;

$D2(5, 1)=\text{ЭМ}$, ($I_0=1 \text{ мкА}$; $RБ=10 \text{ Ом}$, $RУ=10 \text{ МОм}$) — диод $D2$, включенный между узлами 1 и 5, рассчитываемый в программе по модели ЭМ, т. е. Эберса — Молла, имеющий параметры $I_0=1 \text{ мкА}$, $R_{базы}=10 \text{ Ом}$, $R_{утечки}=10^7 \text{ Ом}$.

Параметры можно перечислять в любом порядке.

В некоторых языках для сокращения количества вводимых символов обозначения параметров опускаются. При этом последовательность перечисления параметров уже не может быть произвольной и жестко фиксируется. Например, предыдущее описание диода будет выглядеть так: $D2, 5, 1, 1 \text{ мкА}, 10 \text{ Ом}, 10 \text{ МОм}$.

Заметим, что номера полюсов указываются тоже в определенном порядке, например, для источника напряжения сначала положительный полюс, затем отрицательный, для диода сначала номер вывода анода, затем катода, для транзистора номера выводов коллектора, базы, эмиттера и т. д. Разумеется, в разных языках существуют различные правила описания элементов, которые могут отличаться от приведенных. Так, в некоторых языках необходимо заранее приводить все параметры к какой-либо одной системе размерностей и тогда в описаниях размерность параметров указывать не нужно.

7.6.2. Способы сокращения описания. При описании больших схем объем описания становится весьма большим и его запись в ЭВМ требует много времени; к тому же при этом возрастает вероятность появления ошибок. В то же время во многих схемах используются полностью или частично однотипные элементы, одинаковые дискретные транзисторы, диоды или элементы интегральных схем, изготавливаемые по одной и той же базовой технологии. В связи с этим ряд строк описания будет содержать одинаковые числа и появляется возможность сократить объем описания исключением повторяющихся данных.

Для сокращения описания обычно используют различные виды так называемых *ссылок*. Рассмотрим некоторые из них.

Если элемент, имеющий в описании порядковый номер $N2$, аналогичен элементу с номером $N1$, отличаясь от него лишь номерами узлов (полюсов) включения в схему, то элемент с номером $N2$ может быть описан ссылкой

CC1, $N1$, номера полюсов,
где CC1 — признак ссылки (CC) и ее типа (1).

Если элемент с номером $N2$ отличается от элемента $N1$ не только номерами полюсов, но и частью параметров, например первыми w параметрами, то вместо строки $N2$ может быть записана ссылка CC2:

CC2, $N1$, номера полюсов, w , список отличающихся параметров.

Отметим еще один вид ссылки — ссылку на подсхему. Пусть в схеме можно выделить одинаковые подсхемы, например в двоичном счетчике — триггерные ячейки. Тогда целесообразно описать лишь одну подсхему $P1$, заменив описания остальных ссылками на описание $P1$.

В некоторых языках предусмотрена возможность иерархической системы ссылок на вложенные друг в друга подсхемы. В этом случае группа из N подсхем типа $P1$, описанная с использованием ссылок, может рассматриваться как новая типовая подсхема ти-

па P2 следующего уровня вложения, на которую можно снова ссылаться при описании сложных подсхем, идентичных P2.

7.6.3. Язык описания задания. Эта часть языка предназначена для описания следующих видов заданий:

- на расчет статического режима;
- на расчет переходных процессов;
- на расчет АЧХ;
- на расчет выходных параметров;
- на анализ выходных параметров;
- на оптимизацию выходных параметров.

Следует отметить, что если состав и структура описания схемы в разных языках достаточно похожи, то описания задания существенно отличаются. Тем не менее можно указать характерные особенности структур описания задания. Как правило, эти структуры содержат указания, какие подпрограммы (алгоритмы) должны реализовать тот или иной вид расчета или анализа и каковы должны быть параметры, используемые в этих подпрограммах.

В задании на расчет статического режима указываются способ расчета (прямой, на основе движущейся области сходимости или *через* динамику), тип алгоритма и требуемая точность расчета.

В задании на расчет переходных процессов указываются тип алгоритма (Эйлера, трапеций, метода формул дифференцирования назад (ФДН) и т. д.), допустимая локальная погрешность на одном шаге, минимальный и максимальный допустимые шаги, начальный шаг, коэффициент автоматического изменения шага, предельное время расчета и т. д.

В задании на расчет АЧХ указываются частотный диапазон, шаг изменения частоты, коэффициент изменения шага и т. д.

Структура и содержание заданий на расчет, анализ, оптимизацию выходных параметров схем подробно описана далее в § 12.5 и 14.7.

Важным принципом построения описания задания является его *иерархичность*, т. е. включение в описание задания более высокого уровня описаний заданий более низкого уровня. Так, описание задания на расчет переходного процесса включает описание задания на расчет статического режима. Оба эти описания в свою очередь входят в описание задания на расчет выходных параметров, а оно входит в описания заданий на анализ или оптимизацию.

7.6.4. Язык управления работой программы. В этом языке указываются тип задания (статика, динамика, расчет выходных параметров, статистический, температурный и другие виды анализа, оптимизация) и последовательность смены заданий (например, после оптимизации нужно выполнить статистический анализ в окрестности оптимальной точки). Кроме того, здесь приводятся указания по выводу результатов (в виде графика или таблицы, параметры графика или таблицы, т. е. его размеры на бумаге, шаг печати и др.). Обычно элементами этого языка служат простые директивы типа: «статика», «динамика», «оптимизация», перечисляющие виды работ, или более подробные, например: «динамика, пе-

чать U5, таблица, шаг 3», что означает необходимость распечатки в процессе расчета динамики потенциала в узле 5 в виде таблицы его значений через каждые три расчетные точки.

Как видно из изложенного, входной язык для СхМ существенно сложнее языков для функционально-логического моделирования.

Пример 7.11. Пусть для простейшей RC-цепи (рис. 7.10) необходимо рассчитать статический режим и переходные процессы и распечатать значения переходного процесса в точке 2.

Приведем описание этой задачи на некотором условном языке СхМ. Условность языка в том, что вместо мнемонических обозначений приведены словесные.

Описание схемы

E1(3,0)=постоянный (номинал=2 В, внутреннее сопротивление=1 Ом),
E2(1,3)=скачок (амплитуда=6 В, момент появления=0 нс),
R1(1,2)=1 кОм,
R2(2,0)=1 кОм,
C1(2,0)=100 пФ.

Описание задания

Статика: метод Ньютона; точность=0,01; начальная точка=0.

Динамика: метод Эйлера явный, начальный шаг=10 нс, максимальный шаг=50 нс, локальная точность шага=0,01, время расчета=250 нс, коэффициент изменения шага=2.

Управление работой программы

Статика; печать потенциала в узле 2; динамика; печать потенциала в узле 2 через 5 точек; остановка расчета.

Разумеется, кодировка и ввод приведенного словесного описания потребует много времени, поэтому в конкретных языках вместо слов используют различные условные изображения, символы, часть информации вводится по умолчанию, т. е. независимо от пользователя, если она им не описана. Тем не менее приведенный пример дает достаточное представление о языке схмотехнического проектирования.

7.7. ПРОГРАММЫ СХМОТЕХНИЧЕСКОГО МОДЕЛИРОВАНИЯ

7.7.1. Структура программ. Программы СхМ относятся к одному из наиболее сложных типов программ, каждая из которых фактически является программным комплексом, состоящим из десятков и даже сотен подпрограмм, группируемых в отдельные программные модули. В наиболее общем виде структура программы СхМ приведена на рис. 7.11.

Исходное описание схемы и задачи на входном языке, удобном для пользователя, с помощью программы-транслятора трансфор-

мируется в массивы специальной формы, называемые внутренними форматами. Структура этих форматов выбирается из соображений удобства разработки программы, обеспечения ее максимального быстродействия, минимизации затрат памяти и т. д. Обычно эта структура отличается от структуры данных в исходном описании.

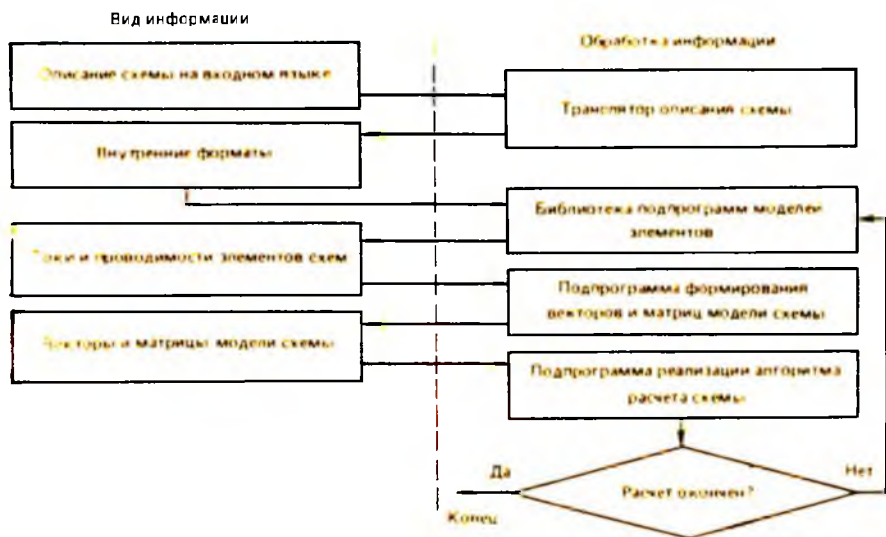


Рис. 7.11. Структура моделирующей части программы АСхП

Далее на основе внутренних форматов формируются основные компоненты модели схемы — векторы и матрицы, входящие в состав алгоритма расчета модели схемы. Например, в программах, работающих в базисе узловых потенциалов, формируются вектор узловых токов I и матрица узловых проводимостей Y , входящие в алгоритмы (7.9) и (7.47); а в программах для метода переменных состояния формируются правые части $f(x, t)$ дифференциальных уравнений в нормальной форме $dx/dt = f(x, t)$.

При этом формировании используются не только внутренние форматы, содержащие численные значения параметров элементов, но и библиотека моделей элементов, содержащая подпрограммы вычисления токов (или напряжений) каждого элемента при указанных во внутренних форматах численных значениях параметров моделей элементов. Эти параметры играют роль фактических параметров подпрограмм расчета моделей.

Далее сформированные векторы и матрицы используются в подпрограмме, реализующей алгоритм расчета модели схемы, например алгоритм Ньютона при расчете статического режима. Как правило, эта подпрограмма выполняет только один элементарный этап расчета, например одну итерацию Ньютона или один шаг численного расчета переходного процесса. После каждой итерации или

шага проверяется условие окончания расчета — сходимость итераций или достижение конца заданного интервала времени — и в случае невыполнения этого условия выполняется следующая итерация или шаг, причем для нелинейных схем необходимо заново формировать численные значения векторов и матриц модели схемы. В некоторых алгоритмах расчета переходных процессов, например основанных на неявных методах решения ОДУ, внутри каждого шага по времени выполняется итерационный процесс расчета квазистатического режима [см. алгоритм (7.47)].

В этом случае подпрограмма реализации алгоритма строится по принципу «цикл в цикле», где внутренним циклом служит, например, цикл итераций по методу Ньютона, а внешним — цикл шагов по методу Эйлера.

Представленная на рис. 7.11 структура относится только к той части программы АСХП, которая рассчитывает статические режимы, переходные процессы или частотные характеристики. Структура же всего комплекса программ схемотехнического проектирования с учетом блоков анализа и оптимизации оказывается значительно сложнее (см. п. 14.6.3).

Обычно в качестве основного языка программирования комплексов программ АСХП используется язык Фортран, хотя в ряде случаев отдельные наиболее часто работающие подпрограммы записываются в целях ускорения их работы на языках типа Ассемблера. Разработка таких комплексов занимает 10...15 человеко-лет, а количество операторов Фортран в них достигает десятков тысяч.

7.7.2. Особенности машинных алгоритмов. Создание программ СхМ связано с необходимостью учитывать различные особенности теоретических алгоритмов при их численной реализации в виде машинных алгоритмов. Так, при реализации теоретических алгоритмов расчета статического режима основными особенностями являются: способ задания начального приближения $x^{(0)}$ и способ контроля сходимости.

1. *Выбор начального приближения* определяется только требованием обеспечения сходимости. Значение $x^{(0)}$ либо задается пользователем на основе знания или приблизительного решения системы (3.23), либо выбирается в программе автоматически, как в методе движущейся области сходимости.

2. *Контроль сходимости* при решении (7.1) методом Ньютона (7.3) выполняется одним из двух способов — по значению нормы поправки $\|x^{(k+1)} - x^{(k)}\| = \|\Delta x^{(k)}\|$ или нормы невязки $\|F(x^{(k)})\|$, k — номер итерации. Если $\|\Delta x^{(k)}\| = 0$, то $\|F(x^{(k)})\| = 0$, однако если $\|F(x^{(k)})\| < \varepsilon$, то может быть, что $\|\Delta x^{(k)}\| \gg \varepsilon$, и наоборот. В связи с этим целесообразно в качестве критерия сходимости требовать

$$\max(\|\Delta x^{(k)}\|, \|F(x^{(k)})\|) < \varepsilon. \quad (7.51)$$

При плохой обусловленности матрицы $F'(x)$ вблизи точки решения критерий сходимости при малых ε может не выполняться из-за «флуктуации» ошибок округления, хотя расходимости метода Ньютона не будет. В этом случае для остановки вычислений

можно рекомендовать применять правило Гарвика, состоящее в том, что выбирается не слишком малое ϵ , при котором еще срывается критерий сходимости (7.51) в условиях «флюктуации» норм невязки или поправки, после чего итерации продолжают до тех пор, пока уменьшается хотя бы какая-либо из норм.

Машинные алгоритмы решения ОДУ отличаются большим разнообразием, так как они содержат значительное количество параметров и частных процедур, выбор и реализация которых не зависит от конкретного численного метода и может быть достаточно произвольной. К основным особенностям машинных алгоритмов решения ОДУ относятся:

- способы прогноза в неявных методах;
- способ контроля ошибки решения ϵ ;
- способ контроля потери устойчивости в явных методах;
- способ определения очередного шага;
- ограничение на максимальный шаг $h_{\max}$;
- способ начала решения в многошаговых методах.

Рассмотрим наиболее распространенные реализации этих способов.

1. *Прогноз в неявных методах* играет двоякую роль. Во-первых, необходимость в соответствии с (7.47) решать на каждом шаге дискретизированную систему ОДУ относительно x_{n+1} порождает около каждой точки x_{n+1} некоторую область сходимости δx_{n+1} численного метода решения. В связи с этим прогноз должен обеспечить попадание начального приближения $x^{(0)}_{n+1}$ в δx_{n+1} , причем возможно ближе к точному решению x_{n+1} , так как при этом уменьшается количество итераций при переходе от $x^{(0)}_{n+1}$ к x_{n+1} . Кроме того, более точный прогноз позволяет увеличить шаг h , при котором $x^{(0)}_{n+1} \in \delta x_{n+1}$.

Во-вторых, при попадании $x^{(0)}_{n+1}$ в δx_{n+1} и сходимости к точке x_{n+1} разность $x^{(0)}_{n+1} - x_{n+1}$ в неявных методах используется для оценки локальной ошибки ϵ численного решения ОДУ, которая получается как попутный результат реализации неявного метода. Таким образом, прогноз $x^{(0)}_{n+1}$ участвует в контроле точности решения.

Обычно прогноз осуществляется на основе предшествующих точке x_{n+1} значений переменной x или оценок ее производных. Для одно- и двухшаговых методов решения ОДУ прогноз чаще всего выполняют по формулам

$$x^{(0)}_{n+1} = x_n \text{ (постоянный прогноз);} \quad (7.52)$$

$$x^{(0)}_{n+1} = x_n + (x_n - x_{n-1})/h \text{ (линейный прогноз).} \quad (7.53)$$

Использование для повышения точности прогноза точек x_{n-2} , x_{n-3} и т. д. в этих методах нецелесообразно, так как требует дополнительной памяти. В многошаговых методах эти точки необходимы для реализации метода, поэтому их используют также и для прогноза, который обычно строится в виде интерполяционного полинома $P(t_n, \dots, t_{n-p})$. Следует, однако, заметить, что увели-

чение степени прогнозирующего полинома не всегда ведет к повышению точности прогноза.

2. *Контроль локальной ошибки решения* ε_{n+1} в точке t_{n+1} может быть осуществлен следующими способами.

Методом двойного счета, т. е. сравнением результатов x_{n+1}^h и $x_{n+1}^{h/2}$, полученных после одного полного шага h и двух половинных $h/2$; если $\varepsilon = |x_{n+1}^h - x_{n+1}^{h/2}| < \varepsilon_{\text{доп}}$, то требования точности удовлетворены. Этот способ, требующий увеличения времени счета вдвое, применяется лишь в явных методах, когда нет других возможностей для контроля ошибки.

В неявных методах для контроля ошибки используется процесс решения ОДУ относительно x_{n+1} :

по разности $\varepsilon_{n+1} = x_{n+1}^{(0)} - x_{n+1}$ между спрогнозированным и окончательным решением;

по числу итераций, необходимых для получения $x_{n+1}^{(k)}$, такого, что $\delta = x_{n+1}^{(k)} - x_{n+1}^{(k-1)} < \delta_{\text{доп}}$.

Если в первом и третьем способах оценки ε и δ лишь косвенно характеризуют локальную погрешность шага, то во втором способе величина ε_{n+1} характеризует эту погрешность непосредственно. Так, при решении ОДУ неявным методом Эйлера с линейным прогнозом (7.53) оценка локальной погрешности в точке t_{n+1}

$$\varepsilon_{n+1} = |(x_{n+1}^{(0)} - x_{n+1})/2|, \quad (7.54)$$

а при решении ОДУ методом ФДН с прогнозом на основе полинома p -го порядка

$$\varepsilon_{n+1} = \frac{h_{n+1} |x_{n+1}^{(0)} - x_{n+1}|}{t_{n+1} - t_{n-p}}. \quad (7.55)$$

Следует отметить, что оценка ошибки (7.55) при $p > 3$ мало надежна из-за большой дисперсии, характерной для интерполяционных полиномов высокой степени.

3. *Контроль потери устойчивости* решения ОДУ осуществляется методами, аналогичными или близкими к методам контроля ошибки. Очевидно, поскольку рост локальной ошибки ε предшествует потере устойчивости, то контроль ошибки предотвращает потерю устойчивости.

В неявных устойчивых методах под потерей устойчивости понимается расходимость итераций при расчете x_{n+1} . Формальным признаком потери устойчивости служит несходимость итерационного процесса за заданное число итераций с заданной ошибкой ε .

В явных методах потеря устойчивости означает превышение допустимого шага. Формальным признаком потери устойчивости служит обычно появление периодических осцилляций («гребенки») значений x_n , x_{n+1} и т. д.

В обоих случаях для сохранения устойчивости нужно уменьшить шаг h . В неявных методах при этом вновь начинает соблюдаться условие $x_{n+1}^{(0)} \in \delta x_{n+1}$, а в явных — условие $h < h_{\text{гр}}$.

4. *Выбор шага* осуществляется на основе результатов контроля точности или устойчивости.

В неявных методах чаще всего используется следующий алгоритм изменения шага. Назначаются оценки $\varepsilon_{\min}$ и $\varepsilon_{\max}$ для локальной ошибки ε . Тогда

$$\begin{aligned} h_{\text{нов}} &= h_{\text{ст}} \text{ при } \varepsilon_{\min} < \varepsilon < \varepsilon_{\max}, \\ h_{\text{нов}} &> h_{\text{ст}} \text{ при } \varepsilon < \varepsilon_{\min}, \\ h_{\text{нов}} &< h_{\text{ст}} \text{ при } \varepsilon > \varepsilon_{\max}. \end{aligned} \quad (7.56)$$

Конкретное соотношение между $h_{\text{нов}}$ и $h_{\text{ст}}$ зависит от способа контроля локальной ошибки ε . Если контроль выполняется прямым способом, с оценкой величины ошибки ε_{n+1} , например, по (7.56), то новый шаг вычисляется по формуле

$$h_{\text{нов}} = \frac{\varepsilon_{\text{доп}}}{\varepsilon_{n+1}} h_{\text{ст}}, \quad (7.57)$$

где $\varepsilon_{\text{доп}} = \varepsilon_{\min}$, если $\varepsilon_{n+1} < \varepsilon_{\min}$ и $\varepsilon_{\text{доп}} = \varepsilon_{\max}$, если $\varepsilon_{n+1} > \varepsilon_{\max}$.

Если контроль локальной ошибки выполняется косвенным способом, например по числу итераций k , требуемых для вычисления x_{n+1} с ошибкой не более $\delta_{\text{доп}}$, то вместо (7.56) можно использовать правило

$$\begin{aligned} h_{\text{нов}} &= h_{\text{ст}} \text{ при } k_{\min} \leq k \leq k_{\max}, \\ h_{\text{нов}} &= c h_{\text{ст}} \text{ при } k < k_{\min}, \\ h_{\text{нов}} &= h_{\text{ст}}/c \text{ при } k > k_{\max}. \end{aligned}$$

Здесь c — коэффициент изменения шага, субъективно задаваемый пользователем, обычно $c=2$. Значения $k_{\min}$, $k_{\max}$ зависят от алгоритма расчета x_{n+1} ; обычно для алгоритма Ньютона выбирают $k_{\min}=2 \dots 3$, $k_{\max}=5 \dots 6$.

В явных методах алгоритм определения шага формулируется следующим образом:

$$\begin{aligned} h_{\text{нов}} &= h_{\max} \text{ при } \Delta x \leq \Delta x_{\text{доп}}, \\ h_{\text{нов}} &= \frac{\Delta x_{\text{доп}}}{\Delta x} h_{\text{ст}} \text{ при } \Delta x > \Delta x_{\text{доп}}, \end{aligned}$$

где $\Delta x_{\text{доп}}$ — задаваемое предельное приращение x на один шаг, $\Delta x = x_n - x_{n-1}$, $h_{\text{ст}} = t_n - t_{n-1}$, $h_{\text{нов}} = t_{n+1} - t_n$, $h_{\max}$ — предельно допустимый шаг, задаваемый пользователем.

5. *Ограничение на максимальный шаг* $h_{\max}$ задается пользователем исходя из особенностей решения конкретной задачи так, чтобы не пропустить быстрых изменений решения $x(t)$, например фронтов. Иногда в неявных методах вводят ограничение на минимальный шаг, чтобы в случае невозможности обеспечить заданную локальную точность (на одном шаге) ограничить процесс уменьшения шага и при $h < h_{\min}$ выдать соответствующую информацию пользователю.

6. *Способ начала решения* в многшаговых методах отличается от способа продолжения решения, так как для него отсутствует

предыстория. Поэтому первые p шагов в многошаговом методе p -го порядка выполняются по тому же алгоритму, но либо с фиксированным первым порядком, т. е. неявным методом Эйлера, либо с фиксированным возрастанием порядка на единицу от шага к шагу, т. е. по x_0 определяется x_1 , по x_0 и x_1 определяется x_2 и т. д. Иногда для большей точности первые p шагов выполняются методом Рунге — Кутты четвертого порядка.

7.7.3. Основные характеристики программ. К основным характеристикам программ АСхП относятся:

выполняемые функции (расчет статических режимов, переходных процессов и др.);

требуемая емкость памяти, Кбайт;

типы используемых моделей элементов;

максимальный размер рассчитываемых схем.

Однако такие общепринятые вычислительные характеристики как точность и быстродействие для программ АСхП обычно не определены, поскольку они зависят от типа и объема схем, типа ЭВМ и других факторов, не позволяющих однозначно указать численные значения этих характеристик.

В настоящее время в эксплуатации находятся десятки программ АСхП. Широко известны программы АРОПС, ПАУМ-2, КАПР-Э, СПАРС, САМРИС-2, СПРОС, ЭЛАИС, ПА-6, КОМПРОМИС и др. Большинство этих программ реализовано на ЭВМ серии ЕС на языке Фортран. Требуемая емкость памяти — от 128 Кбайт (ПАУМ-2, ЭЛАИС) до 200 Кбайт и более (СПАРС, СПРОС). Программы обладают широкими функциональными возможностями, позволяя рассчитывать статические режимы, переходные процессы, частотные характеристики, чувствительность для схем, содержащих биполярные и МДП-транзисторы, диоды, различные типы управляемых и постоянных источников тока и напряжения, R , L , C -компоненты и т. д.

Как правило, библиотеки моделей этих программ содержат для одного типа транзисторов (биполярного или МДП) различные модели, отличающиеся точностью и скоростью расчета, и пользователь может выбрать любую модель при описании схемы в зависимости от ее объема, требований к точности и времени расчета и т. д. Кроме того, библиотеки моделей являются, как правило, открытыми и, если нужной модели в библиотеке нет, пользователь может сам составить модель и включить ее в библиотеку.

Помимо моделирования и анализа ряд программ (АРОПС, СПРОС, СПАРС, КАПР-Э и др.) может работать в режиме оптимизации.

Предельный размер схем, моделируемых программами АСхП, составляет 150 ... 300 узлов или 250 ... 500 элементов, что примерно соответствует схемам среднего уровня интеграции. Для расчета больших и сверхбольших интегральных схем необходимо применять специальные методы макро моделирования, рассматриваемые далее.

7.7.4. Примеры применения программ. Согласно экспертным оценкам современные программы АСхП позволяют моделировать переходные процессы в 100-узловой нелинейной схеме (около 150 транзисторов) на ЭВМ ЕС-1033 примерно за 30 мин. Эта оценка может служить ориентиром для определения времени и возможности решения различных схемотехнических задач.

Например, статистический расчет методом Монте-Карло требует проведения нескольких сотен статистических испытаний, что не реально для 100-узловой нелинейной схемы. Аналогично, эти схемы не удастся пока оптимизировать без применения специальных методов, так как для оптимизации необходимо выполнить как минимум 50...100 расчетов схемы, а в некоторых случаях — значительно больше. В связи с этим задачи статистического анализа и оптимизации могут быть решены только для нелинейных схем, содержащих в среднем не более 10...15 транзисторов.

Тем не менее даже ограниченная информация о функционировании схемы, получаемая решением простейших задач — моделирования статических режимов и переходных процессов, расчета чувствительности и варьирования параметров элементов — оказывается весьма ценной для разработчика схем, поскольку получение ее иными способами, например макетированием, затруднительно. Поэтому программы АСхП все шире применяются в практике разработки схем, особенно интегрального типа.

В заключение приведем несколько примеров применения программ АРОПС.

1. Расчет статики и динамики для двух входных импульсов в универсальном цифровом счетчике-преобразователе (50 биполярных транзисторов) потребовал 9 мин на ЭВМ ЕС-1033. На основании многовариантного расчета (150 вариантов) этой схемы, что потребовало около 20 ч машинного времени, удалось в два раза повысить максимальную частоту ее переключения. При этом в схему по результатам моделирования на ЭВМ было внесено 21 изменение структуры и параметров.

Нижеследующие расчеты были выполнены на ЭВМ с быстродействием 500 тыс. опер./с.

2. Расчет статики и динамики цифровой схемы, содержащей около 100 МДП-транзисторов, занял 20 мин. Многовариантный расчет этой схемы позволил спроектировать логическую часть ячейки однородной среды для ЭВМ типа ПС-300.

3. Оптимизация быстродействия ячейки ассоциативного параллельного процессора (18 МДП-транзисторов) путем варьирования трех параметров каналов транзисторов при ограничении на площадь кристалла, занимаемого схемой, была выполнена за 30 мин.

4. «Подгонка» формы импульса под заданную в одновибраторе с пленочной $R-C-NR$ -структурой в качестве времязадающей цепи путем оптимизации параметров этой структуры потребовала 20 мин.

Приведенные примеры и практика применения других программ подтверждает их высокую эффективность при решении за-

дач моделирования статике и переходных процессов в нелинейных схемах. Задачи оптимизации решаются пока для схем небольших размеров.

Для линейных схем эффективность программ АСхП еще выше, чем нелинейных, так как время их однократного моделирования значительно меньше. Для 50-узловой линейной схемы время расчета одной точки частотной характеристики на ЭВМ ЕС-1033 составляет в среднем около 1 с, а всей АЧХ — 15 ... 20 с. Это позволяет не только быстро моделировать линейные (50- ... 100-узловые) схемы, но и выполнять их статистический анализ и оптимизацию за приемлемое время.

ГЛАВА 8

СХЕМОТЕХНИЧЕСКОЕ МОДЕЛИРОВАНИЕ БОЛЬШИХ СХЕМ

8.1. ПОСТАНОВКА ЗАДАЧИ

Появление больших и сверхбольших интегральных схем (БИС и СБИС), содержащих $10^3 \dots 10^6$ элементов, привело к серьезным проблемам в области их автоматизированного проектирования.

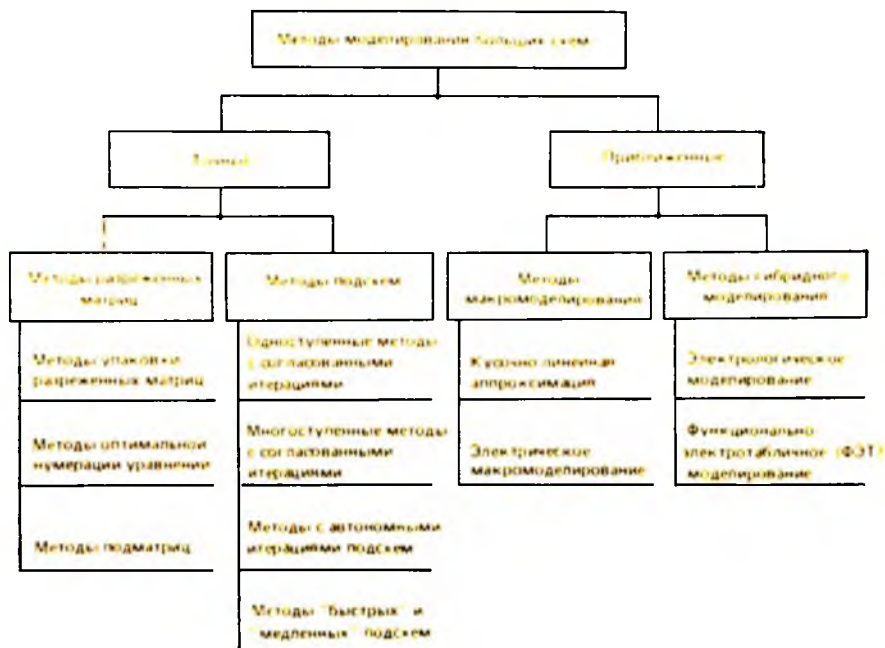


Рис. 8.1. Классификация методов моделирования больших схем

Одной из этих проблем является невозможность расчета математических моделей БИС, состоящих из тысяч и более уравнений, на современных даже наиболее быстродействующих ЭВМ за приемлемое время, составляющее обычно не более нескольких часов. Для этого необходимо использовать специальные методы расчета больших схем [1, 31, 34—37]. Эти методы можно разделить на четыре группы (рис. 8.1).

1. Точные методы, основанные на учете различных особенностей структуры исходной полной модели БИС, прежде всего большого числа нулевых коэффициентов в уравнениях этой модели. Эта группа методов получила название *методов разреженных матриц*.

2. Точные методы, основанные на применении эффективных способов организации вычислений при расчете моделей больших схем путем их декомпозиции, т. е. разделения на части. Эта группа методов получила название *методов подсхем*.

3. Приближенные методы, основанные на замене точных моделей элементов или фрагментов БИС приближенными, более простыми моделями. Эта группа методов получила название *макромоделирования*.

4. Приближенные методы, основанные на совместном использовании методов функционального и схемотехнического моделирования. Методы этой группы часто называют *гибридным моделированием*. Оно занимает промежуточное положение между схемотехническим и функциональным.

8.2. МЕТОДЫ РАЗРЕЖЕННЫХ МАТРИЦ

Данная группа методов предназначена для сокращения затрат памяти и времени при решении систем линейных уравнений вида

$$Ax=B \quad (8.1)$$

Расчет линейных схем сводится к однократному решению этой системы, а расчет нелинейных схем наиболее распространенным методом Ньютона — к ее многократному решению. Поскольку сам метод Ньютона лежит в основе решения дифференциальных уравнений неявными методами численного интегрирования, то ясно, что даже небольшая экономия времени решения (8.1) многократно растёт при расчете переходных процессов и частотных характеристик.

8.2.1. Методы упаковки разреженных матриц. В основе методов разреженных матриц лежит стремление организовать вычисления только с ненулевыми элементами (НЭ) матрицы A в (8.1). Обычно при описании схем системами (8.1) матрицы A оказываются сильно разреженными, т. е. содержащими большое число нулевых элементов.

Например, при использовании базиса узловых потенциалов в роли (8.1) выступает уравнение $Y\Delta\varphi^{(k)}=I(\varphi^{(k)})$ для нелинейных схем или $Y\varphi=I$ для линейных. Таким образом, роль матрицы A играет матрица узловых проводимостей Y .

Так как в каждом узле схемы соединяются обычно от двух до четырех ветвей, то в каждой строке матрицы Y независимо от размера схемы будет от трех до пяти ненулевых элементов, а остальные элементы будут нулевыми.

Степень разреженности матрицы оценивается обычно *коэффициентом заполнения* $k_3 = N_{нэ}/N$, где $N_{нэ}$ — число ненулевых элементов; N — общее число элементов. Для 1000-узловой схемы матрица Y имеет $k_3 = 0,003 \dots 0,005$.

Очевидно, имеет смысл работать только с НЭ. Одним из основных элементов организации такой работы является списочная запись разреженных матриц, получившая название методов *упаковки* матриц. Рассмотрим основные методы упаковки.

В основе всех методов упаковки лежит представление двумерного массива элементов матрицы в виде совокупности одномерных массивов — списков, содержащих сведения о значениях НЭ и об их расположении (координатах) в двумерном массиве. При организации списков используют понятия *позиционных указателей*, или просто указателей — номеров строк или столбцов НЭ и *разделителей* — чисел, позволяющих разделить списки указателей на группы, относящиеся к отдельным строкам и столбцам.

Методы упаковки легче всего пояснить на примерах. Пусть матрица A имеет следующую структуру:

$$\begin{array}{cccc} a_{11} & 0 & 0 & a_{14} \\ 0 & a_{22} & 0 & a_{24} \\ a_{31} & 0 & a_{33} & 0 \\ 0 & 0 & a_{43} & a_{44} \end{array} \quad (8.2)$$

Пример 8.1. Способ полных списков. Матрица A хранится в виде трех списков — двух списков указателей (номеров строк и столбцов) и одного списка значений НЭ. Для матрицы (8.2) эти списки имеют вид (при обходе по столбцам):

список номеров строк: 1, 3, 2, 3, 4, 1, 2, 4;

список номеров столбцов: 1, 1, 2, 3, 3, 4, 4, 4;

список значений НЭ: a_{11} , a_{31} , a_{22} , a_{33} , a_{43} , a_{14} , a_{24} , a_{44} .

Здесь каждому НЭ в (8.2) соответствуют три числа в списках, откуда следует, что экономия памяти при замене двумерного массива полными списками возможна только при коэффициенте заполнения $k_3 < 0,33$.

Пример 8.2. Способ укороченных списков отличается от предыдущего тем, что один из списков указателей заменяется списком разделителей, элементы которого указывают количество НЭ в каждой строке или столбце. Так, список разделителей столбцов для (8.2) имеет вид: 2, 1, 2, 3 и содержит только четыре элемента вместо восьми в списке указателей. Очевидно, при записи (8.2) по строкам нужно использовать список указателей номеров столбцов (1, 4, 2, 4, 1, 3, 3, 4), список разделителей строк (2, 2, 2, 2) и список значений НЭ.

Приведенные способы упаковки являются простейшими. В более сложных способах выделяют в отдельный список диагональные элементы в связи с их более частым использованием. Это позволяет экономить время на их поиск.

8.2.2. Методы оптимальной нумерации уравнений (их называют также методами упорядочения). Для решения системы (8.1) обычно применяют метод Гаусса или его модификации, чаще всего метод *LU*-разложения. В основе этих методов лежит идея последовательного исключения неизвестных из уравнений, так что во втором уравнении исключается одна неизвестная, в третьем две и т. д., а в последнем n -м уравнении — $n-1$ неизвестная.

Этот вспомогательный процесс называется *прямым ходом* решения. Последняя оставшаяся в n -м уравнении неизвестная легко вычисляется и ее значение подставляется в $(n-1)$ -е уравнение, в результате чего в нем остается тоже только одна легко вычисляемая неизвестная и т. д. Этот процесс называется *обратным ходом* решения. Специфика решения методом Гаусса системы (8.1) с разреженной матрицей A состоит в том, что в процессе прямого хода происходит так называемое заполнение матрицы ненулевыми элементами, появляющимися вместо нулевых.

Пример 8.3. Рассмотрим первые два уравнения системы с пятью неизвестными

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + 0 \cdot x_5 = b_1,$$

$$a_{21}x_1 + 0 \cdot x_2 + 0 \cdot x_3 + 0 \cdot x_4 + a_{25}x_5 = b_2$$

Применяя метод Гаусса, первое уравнение разделим на a_{11} , а затем умножим на a_{21} и вычтем из второго, чтобы исключить из него переменную x_1 . В результате получим

$$x_1 + \frac{a_{12}}{a_{11}}x_2 + \frac{a_{13}}{a_{11}}x_3 + \frac{a_{14}}{a_{11}}x_4 + 0 \cdot x_5 = \frac{b_1}{a_{11}},$$

$$-\frac{a_{21}a_{12}}{a_{11}}x_2 - \frac{a_{21}a_{13}}{a_{11}}x_3 - \frac{a_{21}a_{14}}{a_{11}}x_4 + a_{25}x_5 = b_2 - \frac{b_1a_{21}}{a_{11}}.$$

Как видим, вместо нулевых коэффициентов при x_2, x_3, x_4 появились ненулевые, требующие памяти и времени для дальнейшей обработки. Аналогичное явление имеет место во всех последующих уравнениях, из которых исключается x_2 . Легко заметить, что ненулевые коэффициенты первого уравнения приводят к исчезновению нулей во всех последующих уравнениях.

Если же поменять порядок исключения, переставив уравнения местами, то получим

$$x_1 + 0 \cdot x_2 + 0 \cdot x_3 + 0 \cdot x_4 + \frac{a_{25}}{a_{21}}x_5 = \frac{b_2}{a_{21}},$$

$$a_{12}x_2 + a_{13}x_3 + a_{14}x_4 - \frac{a_{25}a_{11}}{a_{21}}x_5 = b_1 - \frac{b_2a_{11}}{a_{21}}.$$

Здесь появился только один новый ненулевой коэффициент при x_5 . Если бы коэффициенты a_{12}, a_{13}, a_{14} были нулевыми, то их значения не изменились бы.

Таким образом, число новых НЭ существенно зависит от порядка рассмотрения уравнений при исключении неизвестных в процессе прямого хода. Правило определения этого порядка называется *критерием упорядочения*. Приведем наиболее распространенные критерии.

Пусть i и j — номера рассматриваемых уравнений (строк матрицы A в (8.1)). Наиболее простым будет критерий:

Кр.1: $i < j$, если $СНЭ_i < СНЭ_j$,
где $СНЭ_i$, $СНЭ_j$ — количество старых НЭ (до решения) в строках с номерами i и j . Справедливость этого критерия вытекает из предыдущего примера. Однако здесь не учитывается, что появление новых НЭ (ННЭ) может сильно повлиять на разреженность. Поэтому используют и другие критерии, учитывающие это влияние, например:

Кр. 2: $i < j$, если $ННЭ_i < ННЭ_j$,
где $ННЭ_i$, $ННЭ_j$ — число ННЭ, появляющихся при рассмотрении i -го или j -го уравнения;

Кр. 3: $i < j$, если $СНЭ_i + ННЭ_i < СНЭ_j + ННЭ_j$;

Кр. 4: $i < j$, если $\max ННЭ_i < \max ННЭ_j$,
где $\max ННЭ_i$ — максимально возможное число ННЭ при рассмотрении i -го уравнения.

Чаще всего применяют критерии Кр2 и Кр4.

Критерий применяется на каждом шаге исключения неизвестных, его результатом является определение номера уравнения (строки матрицы A), обработка которого дает минимальное количество ННЭ. Применение критерия на каждом из n шагов дает последовательность номеров уравнений, следуя которой при прямом ходе получим число ННЭ, близкое к минимально возможному.

Исследования эффективности различных критериев показывают следующее:

1) существенное уменьшение числа ННЭ получается уже при применении самых простых критериев;

2) с учетом затрат памяти и времени на сами программы упорядочения и их работу процедуры упорядочения целесообразно применять при порядке системы $n > 15 \dots 20$;

3) при увеличении n ($n > 100$) различие в эффективности критериев сглаживается.

Последнее объясняется необходимостью многократного применения критерия в так называемых неопределенных ситуациях, когда критерий удовлетворяется одновременно для разных номеров уравнений и, следовательно, порядок исключения может выбирать-ся неоднозначно.

Применение методов оптимальной нумерации совместно с методами упаковки разреженных матриц позволяет существенно снизить вычислительные затраты на решение системы (8.1). Так, обычный метод Гаусса при решении системы из n уравнений требует n^2 ячеек памяти и $(n^3 + 3n^2 - n)/3$ длинных операции (умножения и деления), при использовании же методов оптимальной нумерации затраты памяти пропорциональны n , а количество длинных операций — пропорционально n^k , где $1 < k < 2$.

8.2.3. Методы подматриц. Эти методы основаны на использовании разреженности матрицы A в (8.1) путем такой нумерации уравнений и переменных, при которой нулевые и ненулевые элементы матрицы A группируются в отдельных подматрицах, причём

нумерация выполняется так, чтобы образовалось возможно большее число нулевых подматриц, а ненулевые подматрицы имели возможно больший коэффициент заполнения. Идею этих методов можно проиллюстрировать на примере решения уравнения узловых потенциалов:

$$Y\varphi = I. \quad (8.3)$$

Разобьем схему на p подсхем. Так как подсхемы связаны друг с другом только через граничные узлы, то на пересечении строк и столбцов матрицы Y , соответствующих внутренним узлам разных подсхем, будут находиться только нулевые элементы. Чтобы сгруппировать эти строки и столбцы в единый массив, пронумеруем подряд сначала все граничные узлы всей схемы, а затем по очереди внутренние узлы первой, второй и т. д. подсхем. Тогда уравнение узловых потенциалов (8.3) можно записать в следующем виде:

$$\begin{bmatrix} Y_{гр} & Y_{гв1} & Y_{гв2} & \dots & Y_{гвi} & Y_{гвp} \\ Y_{вг1} & Y_{вв1} & 0 & \dots & 0 & 0 \\ Y_{вг2} & 0 & Y_{вв2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_{вгi} & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_{вгp} & 0 & 0 & \dots & 0 & Y_{ввp} \end{bmatrix} \begin{bmatrix} \varphi_{гр} \\ \varphi_{в1} \\ \varphi_{в2} \\ \vdots \\ \varphi_{ви} \\ \vdots \\ \varphi_{вp} \end{bmatrix} = \begin{bmatrix} I_{гр} \\ I_{в1} \\ I_{в2} \\ \vdots \\ I_{ви} \\ \vdots \\ I_{вp} \end{bmatrix}. \quad (8.4)$$

Здесь $Y_{вви}$ ($i=1, \dots, p$) — подматрицы проводимостей между внутренними узлами i -й подсхемы; $Y_{гви}$ и $Y_{вги}$ — подматрицы проводимостей между внутренними и граничными узлами этой же i -й подсхемы; $\varphi_{ви}$, $\varphi_{гр}$ — векторы потенциалов внутренних узлов каждой подсхемы и граничных узлов всех подсхем; $I_{ви}$ и $I_{гр}$ — векторы узловых токов во внутренних и граничных узлах; матрицы вида $Y_{гги}$ входят в матрицу $Y_{гр}$.

Как видим, значительная часть матрицы Y содержит нули, поэтому обращение полной матрицы Y невыгодно из-за излишних затрат памяти и времени на операции с нулевыми элементами. Чтобы решить (8.4) не обращая полную матрицу, заметим, что

$$\varphi_{ви} = -Y_{ввi}^{-1} Y_{вги} \varphi_{гр} - Y_{ввi}^{-1} I_{ви}. \quad (8.5)$$

Тогда для определения $\varphi_{гр}$ получим уравнение

$$\left(Y_{гр} - \sum_{i=1}^p Y_{гви} Y_{ввi}^{-1} Y_{вги} \right) \varphi_{гр} + \left(I_{гр} - \sum_{i=1}^p Y_{гви} Y_{ввi}^{-1} I_{ви} \right) = 0.$$

После вычисления $\varphi_{гр}$ легко найти по (8.5) все $\varphi_{ви}$.

Структура матрицы Y в (8.4) называется блочно-диагональной с окаймлением. Ее можно получить для схем любого типа. Однако для схем специального вида, например, представляющих собой последовательность каскадов без обратных связей между ними, матрица Y может иметь иную структуру — блочно-диагональную (с тремя диагоналями из подматриц) без окаймления. Для этого нет необходимости выделять граничные узлы в отдельную группу,

можно пронумеровать узлы последовательно каскад за каскадом.

В целом, в зависимости от вида схемы (каскадная, с обратными связями или без них, цепочная и т. д.), можно применять ту или иную нумерацию ее узлов и получать различные структуры матриц коэффициентов. Опираясь в этих структурах при решении (8.1) лишь с подматрицами, содержащими ненулевые элементы, можно получить большой выигрыш во времени и памяти по сравнению с решением на основе полной матрицы.

Методы подматриц получили меньшее распространение, чем методы упорядочения, так как они требуют вполне определенной нумерации переменных, выполняемой вручную, что создает определенные неудобства для пользователя.

8.3. МЕТОДЫ ПОДСХЕМ

8.3.1. Линейные подсхемы. Основную идею методов подсхем, называемых также методами *диакоптики*, или расчета по частям, легче всего пояснить для случая линейной схемы. Разобьем ее для простоты рассуждений на две подсхемы, уравнения которых в общем случае можно записать в следующем виде:

$$\Phi_{п1}(x_r, x_{в1}) = 0, \quad \Phi_{п2}(x_r, x_{в2}) = 0, \quad (8.6)$$

где x_r — граничные переменные подсхем, например потенциалы внешних узлов или напряжения на внешних ветвях подсхемы, общих с другими подсхемами; $x_{в1}$, $x_{в2}$ — внутренние переменные подсхем.

Уравнения всей схемы относительно граничных переменных запишем в виде

$$\Phi_r(x_r, x_{в1}, x_{в2}) = 0. \quad (8.7)$$

Заметим, что размер этой системы равен количеству граничных переменных x_r . Решая уравнения подсхем (8.6) относительно $x_{в1}$ и $x_{в2}$, получаем

$$x_{в1} = F_{п1}(x_r), \quad x_{в2} = F_{п2}(x_r). \quad (8.8)$$

Это — уравнения связи внутренних и граничных переменных каждой подсхемы. Подставляя их в уравнения всей схемы (8.7), получаем уравнения схемы с исключенными внутренними переменными подсхем; эти уравнения можно назвать уравнением сокращенной схемы

$$\Phi_r(x_r, F_{п1}(x_r), F_{п2}(x_r)) = \Phi_r(x_r) = 0. \quad (8.9)$$

Размер этой системы определяется только числом граничных переменных, которое может быть много меньше общего числа переменных x_r , $x_{в1}$ и $x_{в2}$. Решая (8.9), находим x_r . Подставляя x_r в уравнения связи (8.8), определяем $x_{в1}$ и $x_{в2}$.

Таким образом, вместо решения системы уравнений полной схемы размера $[x_r] + [x_{в1}] + [x_{в2}]$, где $[x]$ — размер вектора x , здесь раздельно решаются три системы уравнений меньшего размера: $[x_{в1}]$ и $[x_{в2}]$ для (8.6) и $[x_r]$ для (8.9).

Пример 8.4. Проиллюстрируем рассмотренный метод применительно к расчету линейной схемы в базисе узловых потенциалов Φ . Разделим схему на две подсхемы, проводя границу между ними через граничные узлы. Обозначим Φ_{B1} , Φ_{B2} , $\Phi_{Г1}$, $\Phi_{Г2}$ — потенциалы внутренних и граничных узлов каждой подсхемы. Хотя для двух подсхем в силу общности всех граничных узлов $\Phi_{Г1} = \Phi_{Г2} = \Phi_{Г}$, будем для общности рассмотрения считать $\Phi_{Г1} \neq \Phi_{Г2}$. Тогда (8.6) для внутренних узлов подсхем будут иметь вид

$$\begin{aligned} \Phi_{B1}(\Phi_{B1}, \Phi_{Г1}) &= |Y_{BB1} \ Y_{BG1}| \begin{vmatrix} \Phi_{B1} \\ \Phi_{Г1} \end{vmatrix} + I_{B1} = 0, \\ \Phi_{B2}(\Phi_{B2}, \Phi_{Г2}) &= |Y_{BB2} \ Y_{BG2}| \begin{vmatrix} \Phi_{B2} \\ \Phi_{Г2} \end{vmatrix} + I_{B2} = 0. \end{aligned} \quad (8.10)$$

Здесь Y_{BB} , Y_{BG} — подматрицы узловых проводимостей между только внутренними (BB) и между внутренними и граничными узлами (BG), I_{B1} и I_{B2} — суммы источников тока во внутренних узлах.

Решая (8.10) относительно Φ_{B1} , Φ_{B2} , получаем уравнения связи Φ_{B1} и Φ_{B2} с $\Phi_{Г}$ вида (8.8):

$$\Phi_{B1} = Y^{-1}_{BB1} (-I_{B1} - Y_{BG1} \Phi_{Г1}), \quad \Phi_{B2} = Y^{-1}_{BB2} (-I_{B2} - Y_{BG2} \Phi_{Г2}). \quad (8.11)$$

Запишем уравнения вида (8.7) для граничных узлов всей схемы:

$$|Y_{ГГ1} \ Y_{ГB1}| \begin{vmatrix} \Phi_{Г1} \\ \Phi_{B1} \end{vmatrix} + I_{Г1} = Q_{Г1}; \quad (8.12a)$$

$$|Y_{ГГ2} \ Y_{ГB2}| \begin{vmatrix} \Phi_{Г2} \\ \Phi_{B2} \end{vmatrix} + I_{Г2} = Q_{Г2}. \quad (8.12b)$$

Здесь $Q_{Г1}$ и $Q_{Г2}$ — невязки узловых токов в граничных узлах, т. е. вклады каждой подсхемы в узловые точки граничных узлов. Очевидно, для двух подсхем $Q_{Г1} = -Q_{Г2}$, так как должно быть $Q_{Г1} + Q_{Г2} = 0$ по первому закону Кирхгофа, а $Y_{ГГ1} = Y_{ГГ2} = Y_{ГГ}$, так как граничные узлы принадлежат обоим подсхемам. Однако для большего числа подсхем, когда общими для двух подсхем будут не все граничные узлы $Q_{Гi} \neq Q_{Гj}$, $Y_{ГГi} \neq Y_{ГГj}$, где i, j — номера подсхем.

Подставляя Φ_{B1} и Φ_{B2} из (8.11) в (8.12), получаем

$$\begin{aligned} (Y_{ГГ1} - Y_{ГB1} Y^{-1}_{BB1} Y_{BG1}) \Phi_{Г1} + (-Y_{ГB1} Y^{-1}_{BB1} I_{B1} + I_{Г1}) &= Q_{Г1}, \\ (Y_{ГГ2} - Y_{ГB2} Y^{-1}_{BB2} Y_{BG2}) \Phi_{Г2} + (-Y_{ГB2} Y^{-1}_{BB2} I_{B2} + I_{Г2}) &= Q_{Г2}. \end{aligned} \quad (8.13)$$

Каждое из этих уравнений представляет собой уравнение сокращенной подсхемы, т. е. с исключенными внутренними переменными.

Рассматривая граничные узлы каждой подсхемы как ее полюса, можно уравнения сокращенных подсхем (8.13) считать аналогичными уравнениям многополюсников и записать в эквивалентном виде

$$Y_{п1} \Phi_{Г1} + I_{п1} = Q_{Г1}, \quad Y_{п2} \Phi_{Г2} + I_{п2} = Q_{Г2},$$

где $Y_{п}$, $I_{п}$ — эквивалентные параметры сокращенных подсхем. Тогда для получения уравнений всей схемы нужно использовать методику позиционного суммирования из гл. 7. Учитывая, что $Q_{Г1} + Q_{Г2} = 0$, получаем уравнение всей схемы относительно граничных узлов

$$Y_{Г} \Phi_{Г} + I_{Г} = 0, \quad (8.14)$$

где Y_r , I_r — матрица узловых проводимостей и вектор узловых токов относительно всех граничных узлов.

Из (8.14) легко найти φ_r , а затем из (8.11) — φ_{v1} , φ_{v2} .

Рассмотренный метод подсхем называется *одноступенным*. Если размер подсхем велик, то их можно рассматривать как отдельные схемы, которые делятся в свою очередь на более мелкие подсхемы следующей ступени и т. д. Методика обработки уравнений подсхем внутри каждой подсхемы одинакова. Такой метод называется *многоступенным* методом подсхем.

8.3.2. Расчет нелинейных подсхем на основе алгоритма Ньютона. Основное отличие расчета нелинейных подсхем от расчета линейных состоит в том, что однократное решение уравнений подсхемы доставляет не окончательный результат, а лишь очередное приближение. Поэтому при расчете нелинейных схем методом подсхем предыдущая методика повторяется многократно, так как является лишь одной итерацией общего процесса решения.

Конкретная реализация расчета нелинейных подсхем обычно выполняется в базисе узловых потенциалов на основе алгоритма Ньютона как одного из наиболее быстро сходящихся алгоритмов. В этом случае матрица Якоби соответствует матрице дифференциальных узловых проводимостей Y , а вектор невязок — вектору узловых токов I .

Методика расчета схемы полностью аналогична изложенной в примере 8.4, в уравнениях которого нужно вместо φ использовать вектор поправок $\Delta\varphi^{(k)} = \varphi^{(k+1)} - \varphi^{(k)}$, k — номер ньютоновской итерации. Однократная реализация этой методики при заданных начальных значениях $\varphi^{(0)}_r$, $\varphi^{(0)}_{v1}$, $\varphi^{(0)}_{v2}$ соответствует одной ньютоновской итерации, в результате которой вычисляются поправки $\Delta\varphi^{(1)}_r$ и потенциалы $\varphi^{(1)}_r = \Delta\varphi^{(1)}_r + \varphi^{(0)}_r$. Далее по уравнениям связи (8.11) вычисляются поправки $\Delta\varphi^{(1)}_{v1}$, $\Delta\varphi^{(1)}_{v2}$ и потенциалы $\varphi^{(1)}_{v1} = \Delta\varphi^{(1)}_{v1} + \varphi^{(0)}_{v1}$, $\varphi^{(1)}_{v2} = \Delta\varphi^{(1)}_{v2} + \varphi^{(0)}_{v2}$. На этом первая ньютоновская итерация заканчивается и далее процесс повторяется при вычисленных значениях $\varphi^{(1)}_r$, $\varphi^{(1)}_{v1}$, $\varphi^{(1)}_{v2}$. Для контроля сходимости можно использовать нормы поправок $\|\Delta\varphi^{(k)}_r\|$, $\|\Delta\varphi^{(k)}_v\|$.

В силу того, что в данном способе расчета все итерации — по подсхемам и по граничным переменным всей схемы — выполняются одинаково и взаимосвязанно, этот способ можно назвать *методом подсхем с согласованными итерациями*.

Очевидно, этот метод можно реализовать не только на основе алгоритма Ньютона, но и других алгоритмов, например простых итераций.

8.3.3. Методы с автономными итерациями подсхем. Предыдущий способ на основе алгоритма Ньютона характерен тем, что одна ньютоновская итерация для всей схемы требует выполнения строго одной ньютоновской итерации для каждой подсхемы, так как ньютоновские итерации по подсхемам подчинены общему ньютоновскому алгоритму расчета всей схемы и являются его следствием. Можно сказать, что алгоритм расчета здесь строится

по принципу «сверху вниз», от алгоритма расчета всей схемы к алгоритму расчета каждой подсхемы. Вследствие этого нельзя, например, для одних подсхем выполнять ньютоновские итерации, а для других — простые, или для одних подсхем выполнять одну ньютоновскую итерацию, а для других — серию итераций.

Подчинение итераций по подсхемам алгоритму расчета всей схемы является недостатком данного способа расчета, так как не позволяет учесть индивидуальные свойства подсхем, например различие в степени их нелинейности. От этого недостатка свободны методы с *автономными итерациями подсхем*, в которых итеративный расчет каждой подсхемы выполняется независимо от алгоритмов расчета других подсхем, с учетом ее индивидуальных свойств.

Имеется весьма большое число способов организации итераций для отдельных подсхем и схемы в целом. Вместо детального описания каждого из этих способов поясним их суть следующим обобщающим примером. Пусть нелинейная схема разбита на две подсхемы с внутренними переменными x_{B1} , x_{B2} и граничными переменными x_{r1} , x_{r2} , причем для общности рассмотрения, как и ранее, будем полагать, что $x_{r1} \neq x_{r2}$. Пусть такие уравнения нелинейных подсхем имеют вид (8.6), а уравнения схемы относительно граничных переменных — вид (8.7). Используем обозначение $x = \arg \Phi(x)$ для алгоритмического решения системы уравнений $\Phi(x) = 0$ любым численным методом (см. гл. 3). Тогда итеративный процесс расчета всей схемы по частям в общем случае можно представить тремя последовательно работающими алгоритмами alg_1 , alg_2 , alg_3 , в совокупности образующими алгоритм расчета всей схемы:

$$x_{B1}^{(p+1)} = \text{alg}_1 \Phi_{п1}(x_{B1}^{(p)}, x_{r1}^{(r)}), \quad p = 0, 1, \dots, n; \quad (8.15a)$$

$$x_{B2}^{(k+1)} = \text{alg}_2 \Phi_{п2}(x_{B2}^{(k)}, x_{r2}^{(r)}), \quad k = 0, 1, \dots, m; \quad (8.15b)$$

$$x_r^{(s+1)} = \text{alg}_3 \Phi_r(x_{B1}^{(n)}, x_{B2}^{(m)}, x_r^{(s)}); \quad s = 0, 1, \dots, r, \quad (8.16)$$

где $\Phi_{п1}$ и $\Phi_{п2}$ — системы уравнений для внутренних элементов подсхем (узлов, контуров и т. д.); Φ_r — система уравнений для граничных элементов всей схемы, получаемая объединением уравнений для граничных элементов подсхем.

Алгоритмы (8.15a, б) соответствуют расчету внутренних переменных подсхем при фиксированных значениях граничных переменных, играющих роль параметра, а алгоритм (8.16) соответствует расчету граничных переменных всей схемы при фиксированных значениях x_{B1} и x_{B2} , полученных в конце предыдущих итераций по подсхемам.

Алгоритмы (8.15a, б) соответствуют уравнениям связи (8.8) граничных и внутренних переменных, а алгоритм (8.16) соответствует решению уравнения (8.7) относительно x_r . Однако в отличие от линейного случая, где решение сводится к однократному вычислению x_r по (8.9) и затем x_{B1} , x_{B2} по (8.7), здесь имеет место итеративный процесс решения, в общем случае как внутри алгоритмов (8.15), (8.16), так и между ними. Итерации могут начинаться с лю-

бого алгоритма, т. е. либо относительно x_b при заданных x_r , либо наоборот, в зависимости от того, что известно более точно — x_b или x_r .

Обозначим для краткости через F_i ($i=1, 2, 3$) каждый из алгоритмов (8.15), (8.16), причем пусть обозначение F^1_i соответствует одной итерации алгоритма, а $F^{e_i}_i$ — серии итераций до сходимости с точностью ε . Тогда разнообразные алгоритмы F расчета нелинейных подсхем, отличающиеся различными конкретными способами реализации алгоритмов F_i и способами чередования итераций, можно представить различными комбинациями $F=F_1, F_2, F_3$ (последовательность записи алгоритмов F_i соответствует последовательности их применения):

$$F^1 = F^1_1, F^1_2, F^1_3; \quad (8.17a)$$

$$F^1 = F^{e_1}_1, F^{e_2}_2, F^{e_3}_3; \quad (8.17б)$$

$$F^1 = F^1_1, F^{e_2}_2, F^{e_3}_3; \quad (8.17в)$$

$$F^1 = F^1_1, F^1_2, F^{e_3}_3; \quad (8.17г)$$

здесь F^1 — одна итерация расчета всей схемы.

Предыдущий метод с согласованными итерациями подсхем укладывается в комбинацию (8.17а), причем все алгоритмы F_1, F_2, F_3 должны быть одинаковыми. В более общем случае эти алгоритмы могут быть различны. В способах (8.17б, в, г) одна схемная итерация F^1 может включать несколько подсхемных итераций для каждой подсхемы. Эффективность и целесообразность применения этих способов зависит от индивидуальных свойств подсхем, прежде всего — от степени их нелинейности.

Чтобы оценить область применения данных способов, обратимся к случаю, когда схема линейна. Ее расчет, как показано в п. 8.3.1, сводится к точному расчету линейных подсхем, а затем точному расчету уравнений граничных переменных всей схемы, так что повторных схемных итераций не требуется.

Пусть теперь нелинейная схема — квазилинейна, т. е. ее нелинейности можно с достаточной точностью заменить линейными зависимостями. Очевидно, в этом случае следует ожидать, что количество схемных итераций будет минимальным, если рассчитывать нелинейные подсхемы подобно линейным с предельной точностью, т. е. использовать алгоритмы $F^{e_1}_1, F^{e_2}_2$, а также $F^{e_3}_3$ с минимальными значениями ошибки ε .

Таким образом, для слабо нелинейных схем нужно использовать способ (8.17б) с многократными подсхемными и граничными итерациями. Отсюда следует также, что для существенно нелинейных схем, когда подсхемные итерации по алгоритмам F^e_1 и F^e_2 дают сходимость к точке, недостаточно близкой из-за сильной нелинейности к точке решения для всей схемы, целесообразно использовать способ (8.17а). Способы (8.17в, г) занимают промежуточное положение между рассмотренными, их можно применять, когда степень нелинейности подсхем сильно отличается, или когда часть подсхем — линейна. В этом случае для линейных подсхем

получают точное решение, для квазилинейных подсхем расчет ведется до сходимости, а для сильно нелинейных подсхем выполняется только одна итерация.

Приведенными рекомендациями целесообразно пользоваться при использовании медленно сходящихся алгоритмов типа простых итераций, так как применение согласованных итераций алгоритма Ньютона по схеме (8.17а) при хороших начальных приближениях обеспечивает сходимость за 3...5 схемных (а следовательно, и подсхемных) итераций. В этом случае трудно ожидать, что «автономизация» подсхемных итераций, например по способу $F^1 = F_1^{e_1}, F_2^{e_2}, F_3^{e_3}$, приведет к заметному уменьшению числа схемных итераций.

В целом идея автономных итераций подсхем допускает разнообразные реализации, в частности, релаксационные алгоритмы, когда в первую очередь итерациям подвергаются подсхемы, в которых ошибка текущего приближения максимальна.

8.3.4. Метод «быстрых и медленных» подсхем. Этот метод основан на той же идее разного числа итераций для разных подсхем, что и предыдущий, однако роль итераций здесь выполняют временные шаги численного решения дифференциальных уравнений (ДУ), описывающих переходные процессы в разных подсхемах. Очевидно, для ускорения расчета имеет смысл решать ДУ для подсхем с медленно изменяющимися переменными с большим шагом, внутри которого делать малые шаги решения ДУ для подсхем с «быстрыми» переменными, во время выполнения которых граничные переменные «медленных» подсхем считаются постоянными или хорошо прогнозируются линейными функциями. После выполнения серии малых шагов для «быстрых» подсхем снова рассчитываются внутренние и граничные переменные «медленных» подсхем и т. д.

Идею данного метода упрощенно можно представить в виде последовательности алгоритмов решения ДУ для схемы из двух подсхем — «быстрой» и «медленной»:

$$x_{b1, n+1} = x_{b1, n} + h_{b1} f_1(x_{b1, n}, x_{r, k}, t_n), \quad (t_{n+1})_{\max} = t_k, \quad (8.18a)$$

$$x_{b2, q+1} = x_{b2, q} + h_{b2} f_2(x_{b2, q}, x_{r, k}, t_q), \quad (t_{q+1})_{\max} = t_k; \quad (8.18б)$$

$$x^{(s+1)}_{r, k+1} = x_{r, k} + h_r f_3(x^{(s)}_{r, k+1}, x_{b1, n}, x_{b2, q}, t_k). \quad (8.18в)$$

Если для определенности взять базис узловых потенциалов, то первое уравнение соответствует внутренним узлам «быстрой» подсхемы, а второе — «медленной» подсхемы, поэтому $h_{b2} > h_{b1}$. Точка t_k является точкой синхронизации. Решение в ней третьего уравнения для «граничных» узлов целесообразно выполнять неявными методами для поддержания устойчивости решения ДУ для всей схемы в целом, тогда как в промежуточных точках $t_n, t_q < t_k$ уравнения ДУ для подсхем в зависимости от обстоятельств можно решать как явными, так и неявными методами.

Разновидностью данного метода можно считать *метод учета зон возбуждения*, называемый также *методом учета латентности*, в ко-

тором быстрые и медленные подсхемы не выделяются заранее, а формируются динамически в процессе расчета, причем в качестве подсхем могут выступать отдельные узлы, контуры и т. д.

Рассмотрим узловый базис. В этом базисе узловые уравнения в каждый текущий момент времени t_n составляются и решаются только для так называемых «возбужденных» узлов, в которых изменение потенциалов $\Delta\varphi_i$ или узловых токов ΔI_i на каждом временном шаге или изменение $\Delta\varphi_k$, ΔI_k на каждой итерации внутри шага h превосходит соответственно заданное значение ε_i или $\varepsilon_k < \varepsilon_i$.

Если на k -й итерации для p -го узла $\Delta I_{k,p} < \varepsilon_k$ и $\Delta\varphi_{k,p} < \varepsilon_k$, то на следующей $(k+1)$ -й итерации p -й узел можно рассматривать как постоянный источник узлового тока и не учитывать его вклад в матрицу Y . Если на t -м шаге для p -го узла $\Delta\varphi_{t,p} < \varepsilon_t$, то p -й узел можно отнести к «медленным» и решать дифференциальные уравнения для потенциала этого узла с большим шагом, т. е. пересчитывать вклад данного узла в вектор I и матрицу Y через несколько шагов для «быстрых» узлов.

Успех практической реализации этого метода зависит от правильного выбора критериев возбуждения ε_k , ε_t , а также соотношения шагов для «быстрых» и «медленных» узлов. Поскольку быстрые переходные процессы в больших схемах обычно протекают одновременно лишь в весьма ограниченном количестве узлов, метод позволяет существенно сократить время расчета.

8.4. МЕТОДЫ МАКРОМОДЕЛИРОВАНИЯ

Под макромоделированием понимается упрощение модели схемы путем упрощения моделей ее элементов, достигаемого в результате упрощения характеристик или структуры этих элементов. В отличие от предыдущих точных методов, где сокращение вычислительных затрат достигалось без потери точности, методы макромоделирования относятся к приближенным, когда значительное ускорение расчета достигается ценой незначительной потери точности. Рассмотрим основные методы макромоделирования.

8.4.1. Метод кусочно-линейной аппроксимации (КЛА). Метод КЛА широко использовался еще при аналитических расчетах схем по формулам, но применение ЭВМ на первых порах привело к вытеснению кусочно-линейных моделей более точными нелинейными. Однако эксперименты на ЭВМ показали, что при расчете больших схем общее время расчета в основном определяется временем, затрачиваемым на расчет токов элементов и их производных, необходимых для формирования модели схемы.

В связи с этим оказалось целесообразным вновь вернуться от нелинейных моделей элементов к кусочно-линейным. Применение этих моделей сокращает, во-первых, время расчета токов и проводимостей из-за простоты линейной модели, во-вторых, общее количество вычислений матрицы Якоби, так как на линейном участке модели ее производные постоянны. Необходимая точность до-

стигается увеличением числа участков КЛА, а сходимость вычислений для монотонных характеристик доказана. Применение кусочно-линейных моделей по сравнению с нелинейными позволяет сократить время расчета в 5...10 раз ценой ухудшения точности на 10...15%.

8.4.2. Электрическое макро моделирование. Этот способ макро моделирования предполагает замену точных, но сложных моделей отдельных фрагментов схем (клапанов, усилителей, триггеров и т. д.) гораздо более простыми макро моделями, в которых отражены только наиболее важные для функционирования этих фрагментов характеристики.

Можно выделить два основных типа макро моделей — *физические и информационные*. Физические макро модели — это упрощенные электрические схемы, рассчитываемые, как обычно, на основе уравнений равновесия — законов Кирхгофа, тогда как в информационных макро моделях уравнения равновесия не используются. Информационные макро модели — это модели функционального типа вход-выход, уравнения которых имеют вид $y=f(x)$. Подробное описание макро моделей дано в гл. 11, здесь же подчеркнем только, что электрическое макро моделирование основано на применении физических макро моделей, полностью совместимых с обычными моделями диодов, транзисторов и т. д. Их включение в программу не требует изменения алгоритмов формирования и решения уравнений схемы, так как они состоят из резисторов, конденсаторов, постоянных и управляемых источников тока и напряжения, т. е. тех же элементов, что и обычные модели элементов для схемотехнического моделирования.

Упрощение исходных моделей (*микро моделей*) с помощью физических макро моделей достигается тремя способами: 1) исключением из микро моделей несущественных элементов, отражающих, например, физические эффекты второго порядка точности, 2) заменой отдельных фрагментов микро модели, состоящих из большого числа элементов, фрагментами с меньшим числом элементов и более простой структурой, 3) заменой сложных характеристик элементов микро модели более простыми.

Все три способа основаны на идее аппроксимации электрических (входных, передаточных, выходных, вольтамперных и др.) характеристик микро модели или ее фрагментов электрическими характеристиками макро модели или ее фрагментов. Поэтому физические макро модели называют еще *аппроксимирующими*. Отличительным признаком макро моделей этого типа является возможность их моделирования как в целом, так и по фрагментам на основе только методов теории цепей, что исключает применение в них, например, логических элементов, реализующих булевы или другие логические функции. Более подробно физические макро модели описаны в гл. 11.

Электрическое макро моделирование — еще более мощное средство снижения вычислительных затрат, чем КЛА. КЛА не пони-

жает размера модели большой схемы, тогда как физическая макромодель обычно заменяет фрагменты схемы, существенно снижая размер ее модели.

8.5. ГИБРИДНОЕ МОДЕЛИРОВАНИЕ

В отличие от электрического макромоделирования гибридное моделирование предполагает одновременную реализацию в программе различных видов моделирования — логического, функционального и схмотехнического. Соответственно для моделирования элементов схем используются разные типы моделей — физические и информационные.

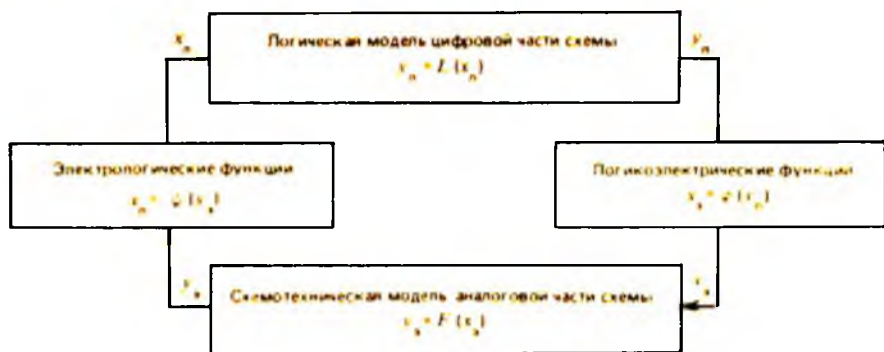
8.5.1. Электрологическое моделирование. Этот вид гибридного моделирования имеет место в том случае, если моделируемая схема состоит из двух частей — аналоговой и цифровой. Моделирование такой схемы целиком методами функционально-логического моделирования оказывается слишком грубым для аналоговой части, а методами схмотехнического моделирования — излишне точным и слишком времязатратным для логической части. В связи с этим каждая часть схемы моделируется методами, обеспечивающими примерно одинаковый порядок точности: цифровая часть — методами логического моделирования, а аналоговая часть — методами схмотехнического моделирования.

Наиболее интересными моментами электрологического моделирования являются переход от логических переменных к электрическим, и наоборот, а также согласование временных шкал (модельного времени) для программ логического и схмотехнического моделирования.

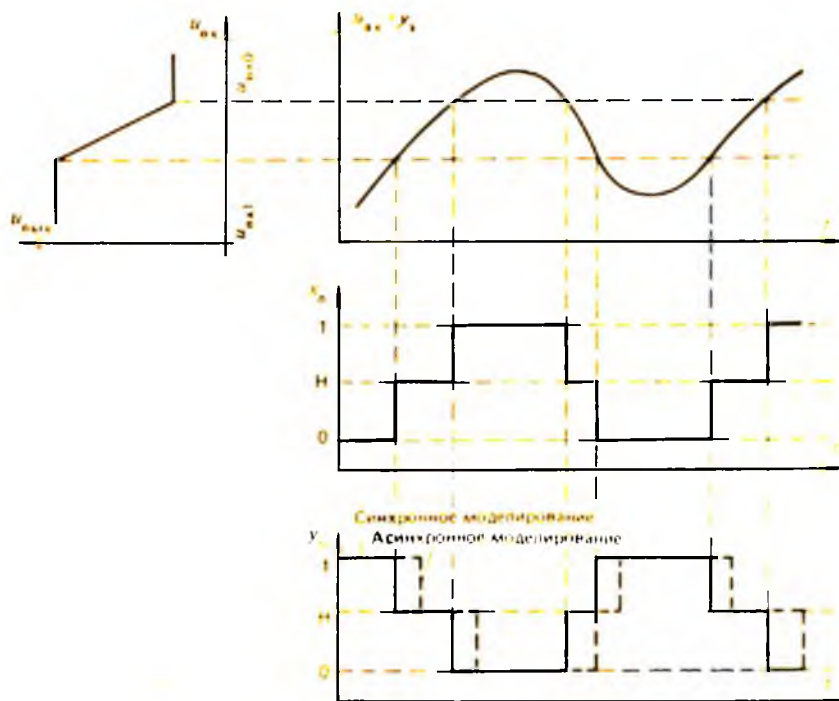
Общая схема согласования переменных разных типов показана на рис. 8.2,а, где $y_l = L(x_l)$ — логическая модель цифровой части схемы, а $y_a = F(x_a)$ — схмотехническая модель аналоговой части, связывающая электрические переменные — токи и напряжения — на выходе и входе логической модели. Для совмещения логического и схмотехнического моделирования в одном вычислительном процессе необходимо иметь соотношения, позволяющие в любой момент времени связать электрические переменные x_a , y_a с логическими переменными x_l , y_l . Для этого, учитывая однонаправленность логической модели, достаточно иметь соотношения двух типов.

Соотношения первого типа можно назвать по аналогии с вольтамперными характеристиками *электрологическими* функциями (аргумент — электрическая величина, функция — логическая). Обычно — это кусочные функции порогового типа $x_l = \varphi(u_{вх})$ или $x_l = \varphi(i_{вх})$. Они ставят в соответствие значениям электрических переменных на входе логической модели значения входных логических переменных x_l .

Соотношения второго типа можно назвать *логико-электрическими* функциями. Они имеют вид $i_{вых} = \varphi(y_l)$ или $u_{вых} = \varphi(y_l)$ или в более общем случае $x_a = \varphi(y_l)$ и ставят в соответствие значениям



а)



б)

Рис. 8.2. Согласование логических и схемотехнических переменных:
а — общий случай; б — на входе логического инвертора; x_n , y_n — логический сигнал соответственно на входе и выходе цифровой части схемы

выходных логических переменных y_n значения электрических переменных x_a .

Пример 8.5. Пусть на входе цифровой части схемы стоит логический инвертор, имеющий передаточную характеристику $u_{вых} = f(u_{вх})$, изображенную на

рис. 8.2,б. Тогда пороговая электрологическая функция может быть представлена в следующем виде:

$$\begin{aligned}x_{\text{л}} &= 0 \text{ при } u_{\text{вх}} < u_{\text{вх}1}, \\x_{\text{л}} &= \text{Н} \text{ при } u_{\text{вх}1} < u_{\text{вх}} < u_{\text{вх}2}; \\x_{\text{л}} &= 1 \text{ при } u_{\text{вх}} > u_{\text{вх}2}.\end{aligned}$$

Здесь $x_{\text{л}} = \text{Н}$ соответствует неопределенному логическому состоянию входа инвертора в трехзначной логике. При $x_{\text{л}} = \text{Н}$ логический сигнал $u_{\text{л}}$ на выходе инвертора также равен Н.

Логико-электрическая функция, если аналоговая часть схемы рассчитывается методом узловых потенциалов, имеет вид:

$$i_{\text{вых}} = \begin{cases} I_1 & \text{при } u_{\text{л}} = 0, \\ I_2 & \text{при } u_{\text{л}} = \text{Н}, \\ I_3 & \text{при } u_{\text{л}} = 1, \end{cases} \quad y_{\text{вых}} = \begin{cases} u_1 & \text{при } u_{\text{л}} = 0, \\ u_2 & \text{при } u_{\text{л}} = \text{Н}, \\ u_3 & \text{при } u_{\text{л}} = 1, \end{cases}$$

где $i_{\text{вых}}$, $y_{\text{вых}}$ — выходные ток и проводимость логической модели; $u_{\text{л}}$ — логическая переменная на выходе логической модели. Если вместо постоянных токов I_1 , I_2 , I_3 задать зависимости $i_1(u)$, $i_2(u)$, $i_3(u)$, то условия на $y_{\text{вых}}$ налагать не нужно.

Аналогичным образом можно согласовывать электрические переменные не только с логическими, но и с переменными любого другого типа, участвующими в расчете схемы, например с точностными, если часть схемы рассчитывается не с позиций логики ее работы, а с позиций ошибок в работе, или с тепловыми, если для части схемы проводится только расчет теплового режима. В этом случае под $x_{\text{л}}$, $u_{\text{л}}$ следует понимать значения погрешностей или температуры.

Таким образом, рассмотренный способ согласования электрических и неэлектрических переменных позволяет моделировать совместно в едином вычислительном процессе разные части схемы, представляемые разнотипными моделями — схемотехнической, логической, функциональной, точностной, тепловой и т. д.

Более того, оказывается возможным использовать разные типы переменных при моделировании не только разных частей схемы, но и одной и той же части или одного и того же элемента схемы. Модели элементов с разными типами переменных называются *гибридными* макромоделями. В зависимости от типа переменных на входе и выходе макромодели различают электрофункциональные и функционально-электрические гибридные модели. Подробно они описаны в гл. 11, здесь укажем только общий принцип их построения.

Обычно гибридная макромодель состоит из трех последовательно включенных частей (рис. 8.3). Первая часть моделирует на электрофункциональном уровне входные цепи, последняя — на функционально-электрическом уровне — выходные цепи, а средняя часть моделирует выполняемое элементом функциональное преобразование, например логическую функцию для комбинационных

логических схем или таблицу переходов для последовательностных схем.

В отличие от достаточно простых правил согласования разнотипных переменных вопрос о согласовании модельного времени логической или любой другой и схмотехнической модели более

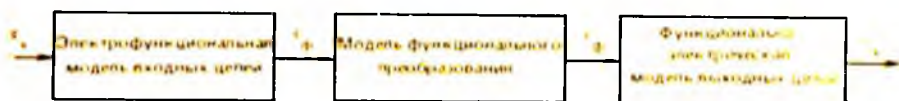


Рис. 8.3. Структура гибридной макромодел

сложен. Этот вопрос возникает потому, что процессы в цифровой и аналоговой частях схемы фактически протекают одновременно, тогда как модели этих частей из-за различий математического аппарата рассчитываются порознь. При этом предполагается, что при моделировании на интервале Δt одной части схемы другая часть на этом интервале не функционирует, хотя на самом деле из-за одновременности протекания процессов это допущение неверно.

Например, изменение электрических переменных при моделировании аналоговой части может вызвать изменение логических переменных на входе и, значит, выходе логической модели, поэтому моделирование на интервале Δt аналоговой части в предположении постоянства действующих на ее входе (см. рис. 8.2,а) логических y_l и соответствующих электрических переменных x_a будет некорректным.

Очевидным решением вопроса является поочередное моделирование на одном и том же интервале Δt_n сначала одной части схемы при постоянстве воздействия другой и затем наоборот, т. е. расчет сначала $y_l(t_n + \Delta t_n)$ при постоянном $y_a = y_a(t_n)$ и соответственно $x_l = x_l(t_n)$, а затем расчет $y_a(t_n + \Delta t_n)$ при постоянном $y_l = y_l(t_n)$ и соответственно $x_a = x_a(t_n)$ (см. рис. 8.2,а). Если на интервале Δt_n значения x_l и y_l не изменяются, то шаг Δt_n принимается, если же изменяются, то нужно уменьшить шаг Δt_n так, чтобы момент $t + \Delta t$ совпал с моментом изменения логических переменных. После этого новый шаг Δt_{n+1} начинается при новых значениях логических переменных.

Обычно в схмотехнической и логической моделях работают разные правила определения очередного шага Δt . В схмотехнической модели он переменный, равный шагу h решения дифференциальных уравнений, и может меняться в широких пределах, в логических моделях шаг постоянный, равный выбранному микротакту T_m при асинхронном моделировании или заданному такту T при синхронном потактовом моделировании. Поскольку варьирование шага допускается только в схмотехнической модели, будем считать, что $\Delta t = h$, причем h может превосходить или быть меньше T_m или T .

Из предыдущего ясно, что шаг h при гибридном электрологическом моделировании оказывается ограниченным не только традиционными соображениями точности расчета переходных процессов, но и условиями постоянства логических переменных x_l и y_l на данном шаге. При этом постоянство x_l легко определить из условия $x_l(t_n + h_n) = \psi(y_l(t_n))$, где ψ — электрологическая функция, а для определения постоянства y_l нужно промоделировать на том же интервале h_n цифровую часть схемы. При этом возможны три случая.

1. Если на интервале h_n все переменные x_l и y_l постоянны, то выполняется следующий шаг h_{n+1} .

2. Если на интервале h_n переменные x_l меняются, а y_l — постоянны или наоборот, то фиксируется момент t_k изменения x_l или y_l и аналоговая часть схемы рассчитывается заново от точки t_n до точки t_k . Точка t_k является начальной для следующего шага h_{n+1} , при этом если изменялось x_l , то цифровая часть моделируется с учетом новых значений $x_l(t_k)$, а если изменялось y_l , то аналоговая часть моделируется с учетом новых значений $x_a = \varphi(y_l(t_k))$.

3. Если меняются как x_l , так и y_l , нужно уменьшать шаг h_n до тех пор, пока один из типов переменных станет постоянным, после чего выполняются действия по соответствующему, описанному выше правилу.

Очевидно, если инерционность логической модели не учитывается, то изменение y_l будет одновременным с изменением x_l и моделировать цифровую часть нужно не на каждом интервале h_n , а лишь в случае изменения x_l , устанавливаемого по результатам схемотехнического моделирования.

Из изложенного ясно, что при электрологическом моделировании целесообразно ввести в программе некоторую общую временную шкалу — *календарь событий*, происходящих в аналоговой и цифровой частях схемы. На этой шкале нужно отмечать после каждого шага h моделирования аналоговой части и после каждого такта T моделирования цифровой части моменты появления всех событий, связанных с изменением y_a и y_l или, что то же, x_l и x_a . Шаг интегрирования дифференциальных уравнений в схемотехнической модели или микротакт моделирования T_m в логической модели (в случае логического моделирования с переменным микротактом) должны определяться не только исходя из точности моделирования внутри каждой модели, но прежде всего с учетом событий, зафиксированных на общей временной шкале-календаре.

Рассмотренные принципы электрологического моделирования остаются справедливыми не только когда цифровая часть — единое устройство, целиком моделируемое на логическом уровне, но и в случае распределенной цифровой части, когда на логическом уровне моделируются отдельные, непосредственно не связанные между собой элементы схемы или функциональные части гибридных макромоделей этих элементов (см. рис. 8.3). В этом случае каждый моделируемый на логическом или гибридном уровне элемент схемы вносит свои отметки на шкале-календаре событий, од-

нако при изменении $x_{\text{л}}$ пересчитываются логические части не всех макромоделей, а только тех, на входе которых произошло изменение $x_{\text{л}}$, при изменении же $y_{\text{л}}$ пересчитывается вся схмотехническая модель, включая схмотехнические части всех гибридных макромоделей, поскольку эта модель описывается общей системой дифференциальных уравнений.

8.5.2. ФЭТ-моделирование. Традиционные методы схмотехнического моделирования оказываются слишком медленными для устройств, содержащих сотни и тысячи транзисторов. В то же время функционально-логическое и даже гибридное функционально-электрическое моделирование оказывается хотя и быстрым, но недостаточно точным. В связи с этим в последние годы появились программы, в которых нет единого математического аппарата моделирования, а используется совокупность разнородных приемов функционально-логического и схмотехнического моделирования, обеспечивающих высокую скорость моделирования за счет допустимых (несколько процентов) потерь точности.

Общепринятого названия этот вид моделирования не имеет, в литературе его называют временным (timing), адекватным и т. д. Будем называть его *ФЭТ-моделированием* (функционально-электротабличным), подчеркивая тем самым разнообразие заложенных в нем принципов. Перечислим эти принципы.

1. Максимальное упрощение схмотехнических моделей. Для этого вольт-амперные нелинейные характеристики и другие нелинейные зависимости представляются таблицами с линейной интерполяцией, что эквивалентно кусочно-линейной аппроксимации. Кроме того, широко используются электрические макромоделли.

2. Максимальное упрощение алгоритмов схмотехнического моделирования. Для этого вместо традиционного метода Ньютона с полным циклом итераций до сходимости на каждом шаге решения дифференциальных уравнений неявными методами используется модифицированный метод Ньютона с одной итерацией на каждом шаге. Это обеспечивает для многих типов схем достаточную точность, более высокую, чем при функциональном моделировании, при сохранении устойчивости решения за счет уменьшения шага, если в этом возникает необходимость. Применение одной ньютоновской итерации эквивалентно методу интегрирования ОДУ, занимающему промежуточное положение между явным и неявным методами Эйлера.

3. Максимальное упрощение математической модели схемы. Для этого используется рассмотренный ранее принцип учета зон возбуждения (принцип латентности), в соответствии с которым на каждом шаге интегрирования ОДУ и на каждой итерации обрабатываются только те уравнения, в которых имеет место существенное изменение переменных.

4. Использование принципов функционального моделирования. Оно состоит в применении гибридных макромоделей с функциональной центральной частью (рис. 8.3) и в моделировании устройств с обратными связями на основе приемов, применяемых в

функциональном моделировании (см. гл. 4), а именно — обратные связи разрываются, рассчитывается выходной сигнал, а затем он подается на вход и т. д. до близкого совпадения формы сигналов на выходе в двух последовательных итерациях.

Применение указанных принципов позволяет ускорить расчет переходных процессов в цифровых сложных устройствах на несколько порядков при потере точности 10 ... 15%. Однако слабым местом ФЭТ-моделирования является эмпирический характер ряда приемов ускорения моделирования, не позволяющий четко определить границы их корректного применения. В конечном счете это снижает универсальность и алгоритмическую надежность ФЭТ-моделирования, делая его пригодным для расчета лишь достаточно узких классов схем с помощью специализированных программ.

ГЛАВА 9

СХЕМОТЕХНИЧЕСКОЕ МОДЕЛИРОВАНИЕ НЕЛИНЕЙНЫХ И ВЫСОКОДОБОРНЫХ ЛИНЕЙНЫХ РАДИОЧАСТОТНЫХ СХЕМ

9.1. ПОСТАНОВКА ЗАДАЧИ

Методы схемотехнического моделирования, рассмотренные в гл. 7, достаточно универсальны и в принципе позволяют рассчитывать любые схемы, математическая модель которых может быть представлена в виде системы обыкновенных дифференциальных уравнений. Вместе с тем существуют классы схем, расчет которых этими методами оказывается неэффективным в смысле вычислительных затрат. Типичным примером могут служить высокочастотные радиотехнические схемы. Как показано в § 4.1 (см. рис. 4.2), расчет характеристик этих схем — времени установления амплитуды высокочастотного переходного процесса, параметров низкочастотного модулирующего колебания и др. — путем обычного прямого численного интегрирования во временной области дифференциальных уравнений, описывающих высокочастотное колебание, требует слишком больших затрат времени и необходимы специальные методы расчета [32, 37—41].

Другим примером являются нелинейные радиотехнические схемы (смесители, модуляторы и др.), при расчете которых возникает ряд специальных задач, не решаемых на основе аппарата дифференциальных уравнений во временной области, например таких, как расчет нелинейных искажений, расчет спектра выходного сигнала и др. Эти задачи решаются не во временной, а в частотной области. Способ их решения существенно зависит от амплитуды гармонического входного сигнала и от типа схемы — линейного или нелинейного. Рассмотрим существо этих задач подробнее.

Пусть на вход простейшей нелинейной схемы (рис. 9.1,а) подаен сигнал $u_{\text{вх}}(t) = E + u_m \cos \omega t$, где E — постоянная составляющая сигнала. Тогда типы возникающих при этом задач можно классифицировать следующим образом.

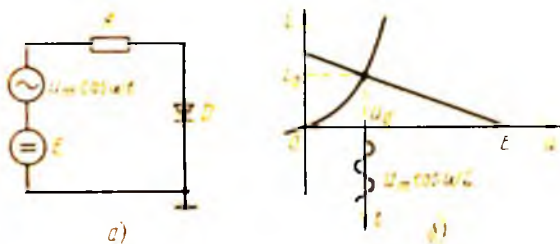


Рис. 9.1. Нелинейная схема (а) и графическая иллюстрация ее расчета в режиме малого сигнала (б)

1. *Расчет по постоянному току.* Его цель — определить положение рабочей точки на вольт-амперных характеристиках нелинейных элементов — в данном случае это точка (u_0, i_0) на характеристике диода (рис. 9.1,б). Методы расчета по постоянному току идентичны методам расчета статического режима, рассмотренным в гл. 7, при этом во входном сигнале учитывается только постоянная составляющая.

2. *Расчет амплитудно- и фазочастотных характеристик (АЧХ и ФЧХ) в линеаризованной схеме.* Методы их расчета аналогичны рассмотренным в гл. 7, однако из-за нелинейности схемы дифференциальные узловые проводимости зависят от режима по постоянному току. Поэтому сначала выполняется расчет по постоянному току, определяется рабочая точка (u_0, i_0) , а затем в этой точке нелинейности линеаризуются путем разложения в ряд Тейлора.

В данном случае, если положить для диода $i = I_0 [\exp(u/(m\varphi_T)) - 1]$, это разложение в точке $u = u_0$ будет иметь вид

$$i(u_0 + \Delta u) = i(u_0) + \left. \frac{di}{du} \right|_{u=u_0} \Delta u + \left. \frac{d^2 i}{du^2} \right|_{u=u_0} \frac{\Delta u^2}{2} + \dots =$$

$$= i_0 + \frac{e^{u_0/(m\varphi_T)}}{m\varphi_T} \Delta u + \frac{e^{u_0/(m\varphi_T)}}{(m\varphi_T)^2} \frac{\Delta u^2}{2} + \dots$$

Ограничиваясь линейным членом разложения, получаем, что дифференциальная проводимость диода y_d — это коэффициент при Δu :

$$y_d = \exp[u_0/(m\varphi_T)] / (m\varphi_T).$$

Это значение y_d используется в модели схемы при расчете амплитудно- и фазочастотных характеристик.

3. *Расчет нелинейных радиотехнических схем (НРТС) в нелинейном режиме малого сигнала.* В отличие от предыдущей задачи, где также сигнал мал, но схема считается линейной, в этой задаче учитывается влияние «нелинейности в малом».

В самом деле, линеаризация нелинейной характеристики диода по Тейлору эквивалентна ее замене касательной в точке (u_0, i_0) (рис. 9.1,б). Очевидно, при этом сигнал на диоде будет чисто моногармоническим и в нем не учитываются составляющие, обусловленные отклонением реальной характеристики от касательной. Эти составляющие образуют высшие по отношению к частоте входного сигнала ω гармоники с частотами 2ω , 3ω и т. д., создающие нежелательные нелинейные искажения, характеризующиеся такими величинами, как коэффициенты нелинейных искажений по мощности, амплитуде и т. д. Методика расчета малых амплитуд высших гармоник и составляет суть расчета НРТС в режиме малого сигнала и излагается в § 9.4.

4. *Расчет НРТС в нелинейном режиме большого сигнала.* В этой задаче в отличие от предыдущей, где высшие гармоники и комбинационные частоты предполагаются малыми, наоборот, они достаточно велики, так как играют полезную роль. К НРТС, работающим в режиме большого сигнала, относятся такие схемы, как смесители, модуляторы, умножители частоты и др., в которых нелинейность принципиально необходима для функционирования. Суть расчета этих схем составляет методика определения больших амплитуд основных высших гармоник и комбинационных частот, возникающих в схеме вследствие ее нелинейного характера.

Вследствие важности перечисленных задач для их решения разработан ряд специальных методов, рассматриваемых далее. Классификация некоторых из них дана на рис. 9.2.

5. *Расчет амплитуды установившегося режима периодических высокочастотных колебаний во временной области.* Рассмотрим колебательный RLC -контур (рис. 9.3,а). Пусть на его входе включен сигнал $E(t) = E \cos \omega t$, а начальные условия u_{C0}, i_{L0} — нулевые. Тогда амплитуда переходных процессов в контуре будет нарастать (рис. 9.3,б) до некоторого постоянного значения, после чего в контуре установится стационарный режим высокочастотных колебаний с постоянной амплитудой. Для этого режима характерен определенный сдвиг фазы колебаний между переменными u_C и i_L .

Разомкнем контур в некоторый момент t_0 , когда амплитуда колебаний установилась, и тут же замкнем его в момент t_1 , полагая, что за время размыкания u_C и i_L не изменились, т. е. $u_C(t_1) = u_C(t_0)$, $i_L(t_1) = i_L(t_0)$. Очевидно, далее процесс будет протекать сразу с установившейся амплитудой, соответствующей стационарному режиму. Если же $u_C(t_1) \neq u_C(t_0)$, $i_L(t_1) \neq i_L(t_0)$, то опять появится процесс установления амплитуды колебаний.

Таким образом, выбор начальных условий влияет на время установления амплитуды высокочастотных колебаний в радиотехнических схемах — линейных и нелинейных. В оптимальном можно так выбрать начальные условия, что расчет коли

временной области будет сразу соответствовать расчету режима со стационарной амплитудой колебаний. Очевидно, для определения характеристик этих колебаний достаточно рассчитать лишь точки для нескольких периодов колебаний, тогда как при неудачном задании начальных условий придется рассчитывать, особенно

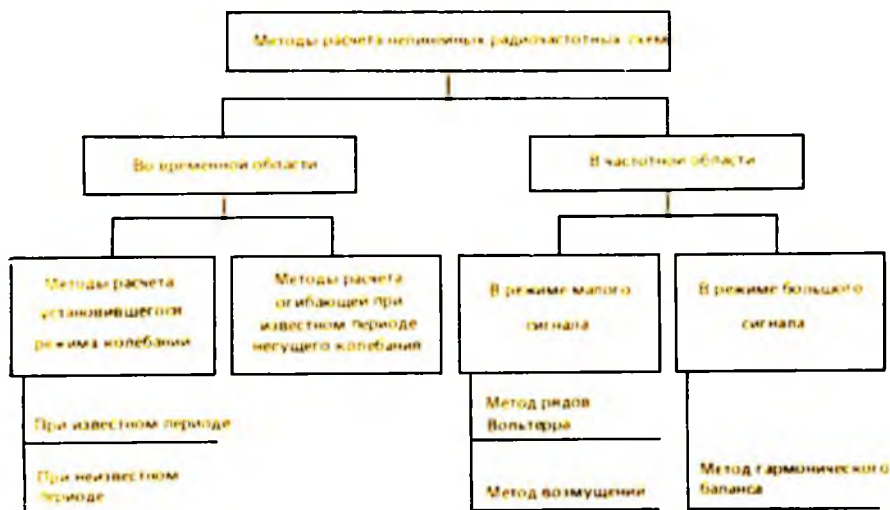


Рис. 9.2. Классификация методов расчета нелинейных радиочастотных схем

в высокочастотных схемах с медленным изменением огибающей, огромное количество периодов колебаний до момента установления их амплитуды. Следовательно, время расчета установившегося режима в высокочастотных схемах можно существенно сократить, правильно задав начальные условия.

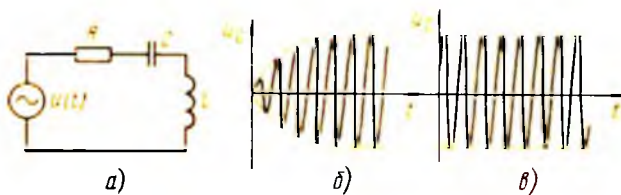


Рис. 9.3. RLC-контур (а) и переходный процесс в нем при произвольных начальных условиях u_{C0} , i_{L0} (б) и при согласованных начальных условиях (в)

В § 9.2 излагается алгоритм расчета оптимальных в указанном смысле начальных условий [32], а алгоритмы расчета НРТС в нелинейных режимах малого и большого сигналов описаны далее в § 9.4 и 9.5.

9.2. АЛГОРИТМЫ ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНЫХ НАЧАЛЬНЫХ УСЛОВИЙ

Пусть дифференциальное уравнение высокочастотного колебательного процесса с известным постоянным периодом T записано в нормальной форме

$$dx(t)/dt = f(x(t), t). \quad (9.1)$$

Интегрируя обе части уравнения от 0 до некоторого произвольного момента t , записываем это уравнение в эквивалентной интегральной форме

$$x_0 + \int_0^t f(x(t), t) dt = x(t), \quad (9.2)$$

где x_0 — начальное условие, которое нужно определить таким образом, чтобы (9.2) описывала процесс $x(t)$ без переходной компоненты установления амплитуды, т. е. значение интеграла, зависящее (после подстановки пределов) от величины $x_0 = x(0)$ было равно нулю.

Математически это требование можно определить следующим образом. Запишем (9.2) для конца периода

$$x_0 + \int_0^T f(x(t), t) dt = x(T). \quad (9.3)$$

Если оптимально выбрать начальные условия x_0 , то переходной компоненты не будет и колебательный процесс, начиная с первого периода, будет стационарным, т. е. $x(T) = x_0$, или с учетом (9.3)

$$x_0 + \int_0^T f(x(t), t) dt = x_0. \quad (9.4)$$

Разумеется, здесь не следует избавляться от x_0 , поскольку именно x_0 является той неизвестной, ради которой составлялось это уравнение.

Поскольку левая часть (9.4), в том числе интеграл, зависит от x_0 , то введем обозначение

$$x_0 + \int_0^T f(x(t), t) dt = F(x_0). \quad (9.5)$$

Тогда условие стационарности процесса с учетом (9.4) можно записать в виде уравнения относительно x_0

$$x_0 - F(x_0) = 0. \quad (9.6)$$

Можно показать, что решение (9.6) методом простых итераций $x_0^{(k+1)} = F(x_0^{(k)})$ эквивалентно прямому интегрированию уравнения (9.1) численным методом, причем скорость сходимости итераций соответствует скорости установления стационарной амплитуды высокочастотного процесса. Очевидно, этот способ решения может привести к большим затратам времени, поэтому будем решать

(9.6) более быстрым методом Ньютона, который запишем в векторном виде

$$\mathbf{x}_0^{(k+1)} = \mathbf{x}_0^{(k)} - (\mathbf{I} - \mathbf{F}'(\mathbf{x}_0))^{-1} (\mathbf{x}_0^{(k)} - \mathbf{F}(\mathbf{x}_0^{(k)})), \quad (9.7)$$

где $\mathbf{I}$ — единичная матрица; $\mathbf{F}'(\mathbf{x}_0) = \partial \mathbf{F}(\mathbf{x}_0) / \partial \mathbf{x}_0$ — матрица Якоби, имеющая смысл матрицы чувствительности «периодических» (т. е. со стационарной амплитудой) значений переменных к изменению начальных условий:

$$\mathbf{F}'(\mathbf{x}_0) = \begin{bmatrix} \frac{\partial F_1(\mathbf{x}_0)}{\partial x_{01}} & \dots & \frac{\partial F_1(\mathbf{x}_0)}{\partial x_{0n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n(\mathbf{x}_0)}{\partial x_{01}} & \dots & \frac{\partial F_n(\mathbf{x}_0)}{\partial x_{0n}} \end{bmatrix} \quad (9.8)$$

$\mathbf{x}_0 = (x_{01}, \dots, x_{0n})$ — вектор начальных условий; $\mathbf{F}(\mathbf{x}_0) = (F_1(\mathbf{x}_0), \dots, F_n(\mathbf{x}_0))$ — вектор результатов интегрирования за период T с начальными условиями $\mathbf{x}_0$.

Алгоритм (9.7) сводится к выполнению следующих операций.

1. Задаемся произвольным значением $\mathbf{x}_0^{(0)}$ и, численно интегрируя в (9.2) от $t=0$ до $t=T$, вычисляем в соответствии с обозначением (9.5) значение $\mathbf{F}(\mathbf{x}_0^{(0)})$.

2. Вычисляем матрицу $\mathbf{F}'(\mathbf{x}_0)$.

3. Вычисляем $\mathbf{x}_0^{(1)}$ по (9.7), проверяем условие сходимости и, если оно не выполняется, переходим снова к п. 1 с новым начальным условием $\mathbf{x}_0^{(1)}$ и т. д. до сходимости к оптимальному значению $\mathbf{x}^*$. Последняя итерация алгоритма (9.7) позволяет найти не только $\mathbf{x}^*$, но и форму колебания в стационарном режиме на интервале $(0, T)$. При необходимости прямое интегрирование в п. 1 может быть продолжено на последующие периоды стационарного режима.

Наиболее трудоемкой операцией в этом алгоритме является вычисление матрицы Якоби (9.8). Ее можно вычислять разными способами, однако поскольку аналитической формулы для зависимости $\mathbf{F}(\mathbf{x}_0)$ нет, то все способы основаны на численных методах. Простейший способ состоит в расчете компонентов матрицы методом приращений:

$$\left. \frac{\partial F_i(\mathbf{x}_0)}{\partial x_{0j}} \right|_{\mathbf{x}_0 = \mathbf{x}_0^{(k)}} \approx \frac{F_i(\mathbf{x}_0^{(k)} + \Delta \mathbf{x}_{0j}) - F_i(\mathbf{x}_0^{(k)})}{\Delta x_{0j}}, \quad (9.9)$$

где $\mathbf{x}_0^{(k)} + \Delta \mathbf{x}_{0j} = (x^{(k)}_{01}, x^{(k)}_{02}, \dots, x^{(k)}_{0j} + \Delta x_{0j}, x^{(k)}_{0,j+1}, \dots, x^{(k)}_{0n})$.

Вычисление всех элементов матрицы Якоби по (9.9) требует вначале интегрирования уравнений (9.1) в соответствии с (9.2) от $t=0$ до $t=T$ при начальном условии $\mathbf{x}_0^{(k)}$, что дает вектор $\mathbf{F}(\mathbf{x}_0^{(k)})$. Это интегрирование выполняется в п. 1 описанного выше алгоритма. Кроме того, для расчета всех приращений ΔF_i ($i=1, \dots, n$) в числителе (9.9) необходимо дополнительно выполнить n интегрирований на интервале $(0, T)$ с разными начальными условиями $\mathbf{x}_0^{(k)} + \Delta \mathbf{x}_{0j}$ ($j=1, \dots, n$). Каждое такое интегрирование позволяет рассчитать один столбец матрицы (9.8).

Таким образом, для схемы с n переменными, т. е. n начальными условиями, с учетом сходимости метода Ньютона за 3...5 итераций получим, что для расчета оптимальных начальных условий, обеспечивающих отсутствие переходной компоненты, необходимо проинтегрировать систему (9.1) на интервале $(0, T)$ $N = (3 \dots 5)(n+1)$ раз. Это эквивалентно по затратам времени прямому интегрированию N периодов подряд, не считая затрат на решение по алгоритму (9.7). Следовательно, данный способ расчета x^*_0 выгодно применять только для небольших схем, когда N еще мало по сравнению с количеством периодов колебания, приходящимся на переходный процесс.

Основное достоинство рассмотренного метода расчета установленного режима состоит в его универсальности, поскольку он применим при любых режимах работы схемы (большого и малого сигнала), а также при любых видах нелинейностей схем. Если в схеме существует несколько стационарных режимов с разными оптимальными начальными условиями, метод сходится к одному из них.

Хотя метод был изложен выше применительно к явной форме записи дифференциальных уравнений (9.1), однако он полностью применим и для неявной формы записи $F(x, x, t) = 0$, поскольку существо всех пунктов реализации алгоритма (9.7) сохраняется, а изменяется лишь методика интегрирования на интервале $(0, T)$ — она основана на дискретизации и алгебраизации неявной формы записи в соответствии с ДК-методом решения ОДУ (см. гл. 7).

Главный недостаток метода — большая трудоемкость вычисления матрицы $F'(x_0)$ методом приращений. В связи с этим для расчета $F'(x_0)$ применяют другие методы, в частности, метод, основанный на решении уравнений чувствительности (уравнений в вариациях). Суть этого метода вкратце сводится к следующему.

Рассматривая начальные условия x_0 как независимые параметры, продифференцируем (9.1) по x_0 :

$$\frac{\partial}{\partial x_0} \left(\frac{dx}{dt} \right) = \frac{df}{dx} \frac{\partial x}{\partial x_0} + \frac{\partial f}{\partial x_0}$$

Вследствие независимости x_0 и t порядок дифференцирования в левой части можно изменить:

$$\frac{\partial}{\partial x_0} \left(\frac{dx}{dt} \right) = \frac{d}{dt} \left(\frac{\partial x}{\partial x_0} \right)$$

Учитывая, что $df/dx_0 = 0$, так как x_0 непосредственно в уравнения не входит, и обозначая чувствительность переменных x к начальным условиям через $y = \partial x / \partial x_0$, получаем линейное относительно y дифференциальное уравнение чувствительности

$$\frac{dy}{dt} = \frac{df}{dx} y \quad (9.10)$$

Это уравнение позволяет найти значение чувствительности в любой момент времени, в том числе при $t = T$. Оно решается одновре-

менно с (9.1) на каждом шаге. Начальные условия $y_0 = \partial x / \partial x_0$ при $t=0$ очевидны; $y_0=1$, так как $x(0)=x_0$. Решение (9.10) совместно с (9.1) выгодно тем, что если для решения (9.1) используется неявный метод, требующий вычисления матрицы Якоби $\partial f / \partial x$, то эта же матрица одновременно участвует и при решении (9.10), так что вычислительные затраты получаются минимальными. Решая (9.10) совместно с (9.1) на интервале $(0, T)$ (п. 1 описанного выше алгоритма), получаем значения $y(T)$, т. е. элементы матрицы $F'(x_0)$, после чего можно сразу выполнять итерацию алгоритма (9.7).

Следует заметить, что (9.10) представляет собой в общем случае систему из $n_y \times n_p$ уравнений, где n_y — число уравнений в (9.1), а n_p — число параметров, по которым определяется чувствительность; в данном случае $n_y = n_p$. Таким образом, система (9.10) требует довольно значительной памяти.

Если (9.1) — система ОДУ в неявной форме $f(x, x, t) = 0$, то уравнение чувствительности приобретает вид

$$\frac{\partial f}{\partial x} \frac{dy}{dt} + \frac{\partial f}{\partial x} y = 0$$

и решается тоже методом дискретизации и алгебраизации, изложенным в гл. 7. Так, дискретизируя dy/dt по формуле метода Эйлера $dy/dt = (y_{n+1} - y_n) / \Delta t$ и полагая $y = y_{n+1}$, если хотим реализовать неявный метод Эйлера, или $y = y_n$, если явный, получаем соответственно

$$y_{n+1} = \left(\frac{1}{\Delta t} \frac{\partial f}{\partial x} + \frac{\partial f}{\partial x} \right)^{-1} \frac{1}{\Delta t} \frac{\partial f}{\partial x} y_n;$$

$$y_{n+1} = \left(\frac{1}{\Delta t} \frac{\partial f}{\partial x} \right)^{-1} \frac{1}{\Delta t} \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial x} \right) y_n.$$

Пример 9.1. Пусть схема описывается системой двух дифференциальных уравнений

$$\frac{dx_1}{dt} = f_1(x_1, x_2, t), \quad \frac{dx_2}{dt} = f_2(x_1, x_2, t) \quad (9.11)$$

с начальными условиями $x_{0,1}$ и $x_{0,2}$. Дифференцируя оба уравнения по $x_{0,1}$ и $x_{0,2}$, получаем следующую систему уравнений чувствительности $y_{ij} = \partial x_i / \partial x_{0,j}$:

$$\begin{aligned} \frac{dy_{11}}{dt} &= \frac{\partial f_1}{\partial x_1} y_{11} + \frac{\partial f_1}{\partial x_2} y_{21}, \\ \frac{dy_{12}}{dt} &= \frac{\partial f_1}{\partial x_1} y_{12} + \frac{\partial f_1}{\partial x_2} y_{22}, \\ \frac{dy_{21}}{dt} &= \frac{\partial f_2}{\partial x_1} y_{11} + \frac{\partial f_2}{\partial x_2} y_{21}, \\ \frac{dy_{22}}{dt} &= \frac{\partial f_2}{\partial x_1} y_{12} + \frac{\partial f_2}{\partial x_2} y_{22}. \end{aligned} \quad (9.12)$$

Решая (9.12) при начальных условиях $y_{11}(0) = y_{22}(0) = 1$; $y_{12}(0) = y_{21}(0) = 0$ на интервале $(0, T)$ совместно с (9.11), находим все $y_{ij}(T)$, после чего можно определить $x_{0,1}^{(k+1)}$, $x_{0,2}^{(k+1)}$ по (9.7).

9.3. РАСЧЕТ ПЕРИОДА СТАЦИОНАРНЫХ КОЛЕБАНИЙ

Если на радиотехническую схему подан внешний сигнал с периодом T , то он определяет частоту вынужденных колебаний в схеме. Однако если схема автономна, т. е. внешних воздействий нет, то период возникающих в такой схеме автоколебаний неизвестен. Для его определения можно использовать модификацию описанного в § 9.2 алгоритма из [32].

Воспользуемся тем обстоятельством, что в автоколебательных системах изменение начальных условий приводит только к сдвигу установившихся колебаний по оси времени без изменения их периода, определяемого только параметрами элементов системы. Поэтому можно любое одно из начальных условий x_{0i} задать произвольно, или, что то же, исключить из рассмотрения, а вместо него ввести новую неизвестную — период T . Тогда вектор неизвестных приобретает вид

$$y = x_{01}, \dots, x_{0,i-1}, T, x_{0,i+1}, \dots, x_{0n},$$

функция «периодических» значений $F(x_0)$ становится функцией

$$F(x_0, T) = x_0 + \int_0^T f(x, t) dt, \quad (9.13)$$

уравнение (9.6) приобретает вид

$$\Phi(y) = x_0 - F(x_0, T) = 0, \quad (9.14)$$

а алгоритм его решения относительно вектора x_0 записывается как

$$y^{(k+1)} = y^{(k)} - (I_i - F'(x_0^{(k)}, T^{(k)})^{-1}) (x_0^{(k)} - F(x_0^{(k)}, T^{(k)})). \quad (9.15)$$

Матрица I_i отличается от единичной тем, что на i -й позиции имеет нуль, так как $\partial x_i / \partial T = 0$ (производные от $\Phi(y)$ берутся по компонентам y , а не x_0), а матрица Якоби $F'(x_0, T)$ отличается от матрицы (9.8) i -м столбцом, имеющим вид

$$\left(\frac{\partial F_1(x_0, T)}{\partial T}, \frac{\partial F_2(x_0, T)}{\partial T}, \dots, \frac{\partial F_n(x_0, T)}{\partial T} \right)^T. \quad (9.16)$$

Все столбцы этой матрицы, кроме i -го, вычисляются так же, как в предыдущем алгоритме, причем в качестве T в $F(x_0, T)$ нужно брать k -е приближение $T^{(k)}$.

Вычисление элементов i -го столбца основано на следующем. Пусть переменными x_i являются переменные состояния u_{ci}, i_{Li} . Запишем известные соотношения

$$\frac{du_c(t)}{dt} = \frac{i_c(t)}{C}, \quad \frac{di_L(t)}{dt} = \frac{u_L(t)}{L}$$

в эквивалентном виде, заменяя t на T и вводя u_{c0}, i_{L0} :

$$\frac{du(T, u_{c0})}{dT} = \frac{i_c(T, u_{c0})}{C}, \quad \frac{di_L(T, i_{L0})}{dT} = \frac{u_L(T, i_{L0})}{L}. \quad (9.17)$$

Из (9.13) следует, что

$$dF_i(T, x_0)/dT = f_i(T, x_0). \quad (9.18)$$

Сравнивая (9.17), (9.18), видим, что для получения элементов столбца (9.16), когда $F_i = u_{ci}$ или $F_i = i_{Li}$, нужно вычислить правые части (9.17) в момент $t = T$, причем на k -й итерации нужно брать $t = T^{(k)}$.

В общем случае, когда x_i не являются переменными состояния и модель схемы имеет неявный вид $F(x, x, t) = 0$, элементы i -го столбца (9.16) вычисляются численно путем дискретизации производных.

Позиция « i » i -го столбца на каждой итерации выбирается для ускорения сходимости так, чтобы значения его элементов были максимальными. В остальном реализация алгоритма (9.15) аналогична реализации алгоритма (9.7).

9.4. РАСЧЕТ СХЕМ В ЧАСТОТНОЙ ОБЛАСТИ В РЕЖИМЕ МАЛОГО СИГНАЛА

При подаче на вход нелинейной радиотехнической схемы нескольких сигналов с частотами $\omega_1, \omega_2, \dots$ в схеме возникают колебания на высших гармониках $m_1\omega_1, m_2\omega_2, \dots$ и на комбинационных гармониках $m_1\omega_1 \pm m_2\omega_2 \pm \dots$. Если амплитуды этих новых гармоник малы по сравнению с амплитудами основных гармоник $\omega_1, \omega_2, \dots$, то говорят, что схема работает в режиме малого сигнала, в противном случае — в режиме большого сигнала.

Ниже рассмотрен один из основных способов расчета для малого сигнала. Оценка возникающих при этом нелинейных эффектов возможна либо непосредственно во временной области по степени искажения проходящих через цепь сигналов, что требует больших затрат времени и дает значительные погрешности, либо по характеристикам оператора передачи цепи. Оценки с помощью операторов передачи цепи давно используются при анализе линейных цепей, например расчет АЧХ и ФЧХ, являющихся компонентами оператора передачи линейной цепи $H(j\omega)$. Важным достоинством оценок, получаемых с помощью операторов передачи, является их независимость от свойств (спектра) передаваемых сигналов.

Оценки искажений в нелинейных *резистивных* цепях, возникающих при передаче через них гармонических сигналов $x(t)$, можно получить, используя оператор передачи вида $y(t) = f(x(t))$, где $y(t)$ — выходной сигнал. Представляя этот оператор в виде степенного полинома

$$y = y_0 + H_1x + H_2x^2 + H_3x^3 + \dots$$

где H_i — коэффициенты полинома, можно при известном входном сигнале $x(t)$ вычислить амплитуды высших гармоник с требуемой точностью. Например, если $x(t) = u_1 \cos \omega_1 t + u_2 \cos \omega_2 t$, то, подставляя $x(t)$ в этот полином, выполняя необходимые действия и груп-

пируя члены с частотами $\omega_1, \omega_2, 2\omega_1, 2\omega_2, \omega_1 \pm \omega_2$ и т. д., легко определить амплитуды колебаний на выходе с этими частотами, а далее по ним вычислить различные коэффициенты, характеризующие искажения сигнала. Обычно в качестве степенного полинома используется ряд Тейлора.

Для оценок искажений в нелинейных *инерционных* цепях с гармоническим входным сигналом чаще всего применяют оператор передачи в виде функционального ряда Вольтерра (ФРВ), который, в отличие от предыдущего, ставящего в соответствие $y(t)$ и $x(t)$ для одного и того же момента времени, определяет $y(t)$ как функцию $x(t)$ в предшествующий бесконечный интервал времени $y(t) = F(x(\tau)t)$, $\tau \leq t$. ФРВ является обобщением интеграла Дюамеля на случай нелинейной инерционной цепи и связывает выходной и входной сигналы в виде

$$\begin{aligned} y(t) &= \sum_{n=1}^{\infty} y_n(t) = \int_{-\infty}^{\infty} h_1(\tau_1) x(t-\tau_1) d\tau_1 + \\ &+ \int \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2) x(t-\tau_1) x(t-\tau_2) d\tau_1 d\tau_2 + \dots = \\ &= \sum_{n=1}^{\infty} \underbrace{\int \dots \int_{-\infty}^{\infty} h_n(\tau_1, \dots, \tau_n)}_{k \text{ раз}} \prod_{i=1}^n x(t-\tau_i) d\tau_i. \end{aligned}$$

Здесь $x(t)$ — входной сигнал; $h_n(\tau_1, \dots, \tau_n)$ имеет смысл нелинейной импульсной характеристики n -го порядка. При $n=1$ величина $h_1(\tau)$ — обычная импульсная характеристика. Добавление членов ряда при $n > 1$ соответствует учету нелинейных преобразований. Если входной сигнал и нелинейности схемы не слишком велики, то достаточно использовать лишь несколько первых членов ряда.

Преобразование Фурье от $h_n(\tau_1, \dots, \tau_n)$

$$\begin{aligned} H_n(f_1, f_2, \dots, f_n) &= \\ &= \int \dots \int_{-\infty}^{\infty} h_n(\tau_1, \dots, \tau_n) \exp[-j2\pi(f_1\tau_1 + \dots + f_n\tau_n)] d\tau_1 \dots d\tau_n \end{aligned}$$

называют передаточной функцией n -го порядка ФРВ. Тогда связь входного и выходного сигналов будет иметь вид

$$y(f_1, f_2, \dots, f_n) = \sum_{k=1}^n H_k(f_1, f_2, \dots, f_k) \prod_{i=1}^k x(f_i). \quad (9.19)$$

Для полигармонического входного сигнала вида $x(t) = \sum_i x_i \exp(j2\pi f_i t)$ выходной сигнал в виде ФРВ определяется выражением [38]

$$y(t) = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{n!}{m_1! \dots m_n!} H_n(f_1, \dots, f_n) \exp[j2\pi(f_1 + \dots + f_n)t]; \quad (9.20)$$

здесь множитель $n!/m_1! \dots m_n!$ соответствует числу сочетаний из индексов частот входных сигналов $f_1, \dots, f_n$ по n .

В качестве примера приведем некоторые выражения, определяющие спектральные составляющие выходного сигнала при подаче на вход цепи сигнала $x(t) = u \cos 2\pi f t$ — постоянная составляющая: $\frac{1}{2}u^2 \operatorname{Re}[H_2(f, -f)]$, составляющая частоты f : $u|H_1(f)| \cos 2\pi f t$, составляющая частоты $2f$: $\frac{1}{2}u^2|H_2(f, f)| \cos 2\pi 2f t$ и т. д. Если сигнал $x(t) = u_1 \cos 2\pi f_1 t + u_2 \cos 2\pi f_2 t$, то составляющая частоты f_1 : $u_1|H_1(f_1)| \cos 2\pi f_1 t$, частоты f_2 : $u_2|H_1(f_2)| \cos 2\pi f_2 t$, частоты $2f_1$: $\frac{1}{2}u_1^2|H_2(f_1, f_1)| \cos 2\pi 2f_1 t$, частоты $f_1 \pm f_2$: $u_1 u_2|H_2(f_1 \pm f_2)| \cos 2\pi (f_1 \pm f_2)t$ и т. д. По значениям амплитуд составляющих отклика цепи можно определить различные критерии нелинейных искажений, например коэффициент искажений по второй гармонике $K_{г2} = (u_{вх}/2)|H_2(f, f)|/|H_1(f)|$ или по третьей гармонике $K_{г3} = (u_{вх}^2/4)|H_3(f, f, f)|/|H_1(f)|$, коэффициент интермодуляции второго порядка $K_{им2} = u_2|H_2(f_1, \pm f_2)|/|H_1(f_1)|$ и т. д.

Для нахождения передаточных функций ФРВ любого порядка нужно решать систему линейных уравнений одного и того же вида, составляемую для узлов схемы

$$\mathbf{I}_n = \mathbf{Y}(f_1 + \dots + f_n) \mathbf{H}_n(f_1, \dots, f_n), \quad (9.21)$$

где $\mathbf{I}_n$ — вектор так называемых нелинейных токов n -го порядка (если $n \geq 2$), смысл которого поясняется ниже; $\mathbf{Y}(f_1 + \dots + f_n)$ — матрица узловых проводимостей схемы на частоте $f_1 + \dots + f_n$.

Передаточные функции ФРВ $H_n(f_1, \dots, f_n)$ удобно вычислять итерационно по следующему алгоритму [39]:

1. Расчет статического режима и определение коэффициентов разложения нелинейных характеристик элементов в ряд Тейлора в рабочей точке.

2. Формирование и решение матричного уравнения

$$\mathbf{I}_1 = \mathbf{Y}(f) \mathbf{H}_1(f), \quad (9.22)$$

где вектор тока $\mathbf{I}_1$ заполнен нулями, кроме единичного источника на входе.

3. Вычисление вектора нелинейного тока второго порядка $\mathbf{I}_2$, в котором значения элементов будут отличны от нуля только в узлах, к которым подключены нелинейные элементы.

4. Решение матричного уравнения

$$\mathbf{I}_2 = \mathbf{Y}(f_1 + f_2) \mathbf{H}_2(f_1, f_2). \quad (9.23)$$

5. Формирование нелинейного тока третьего порядка $\mathbf{I}_3$ и решение матричного уравнения $\mathbf{I}_3 = \mathbf{Y}(f_1 + f_2 + f_3) \mathbf{H}_3(f_1, f_2, f_3)$ и т. д. путем формирования нелинейных токов n -го порядка на основе предыдущих расчетов и решения (9.21).

На каждой k -й итерации схема возбуждается сигналом в виде суммы гармоник $x^{(k)}(t) = \sum_{n=1}^k \exp(j 2\pi f_n t)$, а отклик $y(t)$ отыскивается в результате последовательного вычисления передаточных функций H_k , входящих в представление отклика $y(t)$ в виде ФРВ (9.20). Методика расчета H_k поясняется нижеследующим примером.

Пример 9.2. 1. Пусть для схемы на рис. 9.4 с одной нелинейностью — диодом $i_d = \psi(u_d)$ в соответствии с п.1 описанного алгоритма рассчитан статический режим и в рабочей точке i_{d0} , u_{d0} ток диода представлен рядом Тейлора $i = g_1 u_d + g_2 u_d^2 + g_3 u_d^3 + \dots$, где g_1 , g_2 , g_3 — коэффициенты ряда.

2. Составим узловые уравнения схемы

$$i(t) = g_{\text{ист}} \Phi_1 + g_1 (\Phi_1 - \Phi_2) + g_2 (\Phi_1 - \Phi_2)^2 + \dots, \quad (9.24)$$

$$0 = \frac{\Phi_2}{R} + C \frac{d\Phi_2}{dt} - g_1 (\Phi_1 - \Phi_2) - g_2 (\Phi_1 - \Phi_2)^2 + \dots$$

где Φ_1 , Φ_2 — узловые потенциалы. В терминах ФРВ $i(t) = x(t)$, $\Phi(t) = (\Phi_1, \Phi_2) = y(t)$. На первой итерации, как было указано выше, $i^{(1)}(t) = \exp(j2\pi f t)$, $\Phi(t) = H_1(f) \exp(j2\pi f t)$. Подставляя эти выражения в (9.24), получаем члены, содержащие множители $\exp(j2\pi f t)$, $\exp(j2\pi \times 2f t) \dots$ и т. д. Приравнявая коэффициенты при $\exp(j2\pi f t)$ в правой и левой частях каждого уравнения, получаем уравнение вида (9.22)

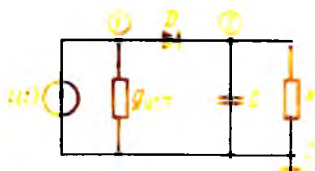


Рис. 9.4. Схема для иллюстрации методов ФРВ

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} g_1 + g_{\text{ист}} & -g_1 \\ -g_1 & g_1 + \frac{1}{R} + j2\pi f \end{bmatrix} \begin{bmatrix} H_1(1)(f) \\ H_1(2)(f) \end{bmatrix}, \quad (9.25)$$

здесь и далее $H_1(1)(f)$, $H_1(2)(f)$ — передаточные функции l -го порядка в узлах 1 и 2. Решая (9.25), находим вектор $H_1(f) = (H_1(1)(f), H_1(2)(f))$.

3, 4. На второй итерации $i^{(2)}(t) = \exp(j2\pi f_1 t) + \exp(j2\pi f_2 t)$, а $\Phi(t)$ из (9.20) определится как

$$\Phi^{(2)}(t) = H_1(f_1) \exp(j2\pi f_1 t) + H_1(f_2) \exp(j2\pi f_2 t) + H_2(f_1, f_2) \exp(j2\pi f_1 t) + \\ + H_2(f_1, f_2) \exp(j2\pi f_2 t) + 2H_2(f_1, f_2) \exp(j2\pi (f_1 + f_2) t) + \dots$$

Подставляя $i^{(2)}(t)$ и $\Phi^{(2)}(t)$ в (9.24) и приравнявая теперь коэффициенты при $\exp[j2\pi (f_1 + f_2) t]$, получаем уравнение вида (9.23)

$$\begin{bmatrix} -g_2 H_1(12)(f_1) H_1(12)(f_2) \\ g_2 H_1(12)(f_1) H_1(12)(f_2) \end{bmatrix} = \begin{bmatrix} g_1 + g_{\text{ист}} & -g_1 \\ -g_1 & g_1 + \frac{1}{R} + j2\pi (f_1 + f_2) C \end{bmatrix} \begin{bmatrix} H_2(1)(f_1, f_2) \\ H_2(2)(f_1, f_2) \end{bmatrix}, \quad (9.26)$$

Здесь $H_1(12) = H_1(1) - H_1(2)$, а левый столбец — это вектор нелинейных токов порядка I_2 , определяемый в общем случае вторыми производными разложения нелинейностей в ряд Тейлора и найденными на 1-й итерации значениями $H_1(f)$. Из (9.26) найдем вектор $H_2(f_1, f_2)$. Правила формирования нелинейных токов любого порядка можно формализовать [39] и тогда п. 3 будет отдельным пунктом данной методики.

5. Аналогично предыдущему при возбуждении схемы на третьей итерации сигналом $i^{(3)}(t) = \sum_{n=1}^3 \exp(j2\pi f_n t)$ получим систему $I_3 = Y(f_1 + f_2 + f_3) H_3(f_1, f_2, f_3)$,

где вектор нелинейного тока третьего порядка будет иметь вид $I_3 = g_3 H_1(f_1) \times \times H_1(f_2) H_1(f_3) + 2g_2 H_1(f_1) H_2(f_2, f_3)$ с известными из предыдущих итераций составляющими. Определяем отсюда $H_3(f_1, f_2, f_3)$ и т. д., после чего можно по H_1, H_2, H_3 вычислить отклик $\Phi(t)$ или найти различные характеристики нелинейных искажений.

Таким образом, изложенная методика позволяет заменить одноразовое решение нелинейных дифференциальных уравнений схемы во временной области многократным решением систем линейных дифференциальных уравнений, а в частотной области — решением соответствующих систем алгебраических уравнений вида (9.21), каждая из которых составляется и решается для определенных частот. При этом каждая нелинейность представляется по отношению к линейной части схемы током возбуждения соответствующего порядка, схема рассматривается как линейная и реакция $y(t)$ схемы получается суперпозицией действия отдельных гармоник сигнала.

Основной недостаток метода ФРВ — его применимость только при относительно малых сигналах, так как при больших сигналах ФРВ расходится.

9.5. РАСЧЕТ СХЕМ В ЧАСТОТНОЙ ОБЛАСТИ В РЕЖИМЕ БОЛЬШОГО СИГНАЛА

Расчет НРТС в режиме большого сигнала можно выполнить на основе метода гармонического баланса. Существо этого метода применительно к данной задаче состоит в следующем. Пусть на вход НРТС действует гармонический сигнал $u_{вх} = u_{вх} m \cos \omega_c t$. Тогда в нелинейной схеме возникнут колебания сложной формы.

Рассмотрим какой-либо нелинейный компонент схемы, описываемый компонентным уравнением $i = f(u)$. Если напряжение на нем изменяется во времени по закону $u = u(t)$, то закон изменения тока будет $i(t) = f(u(t))$. Перейдем из временной области в частотную. Для этого представим $u(t)$ и $i(t)$ разложением в ряд Фурье по гармоникам входного сигнала:

$$u(t) = u_0 + \sum_{k=1}^{\infty} u_k \cos k \omega_c t, \quad i(t) = i_0 + \sum_{k=1}^{\infty} i_k \cos k \omega_c t.$$

Тогда компонентное уравнение $i = f(u)$ принимает вид

$$i_0 + \sum_{k=1}^{\infty} i_k \cos k \omega_c t = f \left(u_0 + \sum_{k=1}^{\infty} u_k \cos k \omega_c t \right). \quad (9.27)$$

Выполнив преобразование f и приравнявая справа и слева постоянные составляющие и соответствующие гармоники, компонентное уравнение можно представить совокупностью уравнений

$$\begin{aligned} i_0 &= f_0(u_0) \\ i(\omega_c) &= f_1(u(\omega_c), u(2\omega_c), u(3\omega_c), u(4\omega_c), \dots) \\ i(2\omega_c) &= f_2(u(\omega_c), u(2\omega_c), u(3\omega_c), u(4\omega_c), \dots) \end{aligned} \quad (9.28)$$

Систему этих уравнений, связывающих амплитуды гармонических составляющих тока и напряжения, назовем *гармоническими компонентными уравнениями*. Отметим, что каждая гармоника тока $i(k\omega_c)$ зависит от всех гармоник напряжения $u(\omega_c)$, $u(2\omega_c)$ и т. д., так как вклад в $i(\omega_c)$ дают комбинационные частоты $2\omega_c - \omega_c$, $3\omega_c - 2\omega_c$ и т. д., а вклад в $i(2\omega_c)$ дают комбинационные частоты $3\omega_c - \omega_c$, $4\omega_c - 2\omega_c$ и др., образующиеся от взаимодействия $u(3\omega_c)$ и $u(\omega_c)$, $u(4\omega_c)$ и $u(2\omega_c)$ и т. д.

Таким образом, каждой переменной $u(t)$ будет соответствовать группа переменных u_0 , $u(\omega_c)$, $u(2\omega_c)$ и т. д. Соответственно, системе уравнений схемы

$$A i(t) = A f(u(t)) = 0,$$

где A — топологическая матрица узлов, будет соответствовать множество систем узловых уравнений, образующих общую систему узловых уравнений:

$$A i_0 = A f_0(u_0) = 0,$$

$$A i(\omega_c) = A f_1(u(\omega_c), u(2\omega_c), \dots) = 0, \quad (9.29)$$

$$A i(2\omega_c) = A f_2(u(\omega_c), u(2\omega_c), \dots) = 0.$$

Ограничиваясь в ряде Фурье конечным количеством гармоник N , получаем $N+1$ систему уравнений. Если в схеме l ветвей, т. е. l неизвестных напряжений, то модель схемы должна состоять из $(N+1)l$ уравнений относительно постоянных и гармонических составляющих $u^1_0, u^2_0, \dots, u^l_0, u^1(\omega_c), u^2(\omega_c), \dots, u^l(\omega_c), \dots, u^1(N\omega_c), u^2(N\omega_c), \dots, u^l(N\omega_c)$, где верхний индекс — номер ветви. Большое количество уравнений — недостаток данного метода. В то же время этот метод в отличие от метода ФРВ связан не с разложением в ряд Тейлора, а с разложением в ряд Фурье и поэтому его точность зависит не от амплитуды сигнала, а от количества N учитываемых гармоник.

Переходя в системе (9.29) от напряжений к узловым потенциалам $u_{mn}(k\omega_c) = \varphi_m(k\omega_c) - \varphi_n(k\omega_c)$ и решая эту систему, получаем спектральные составляющие в каждом узле схемы, обусловленные входным сигналом на частоте ω_c . Очевидно, метод можно применять при любом количестве входных сигналов с частотами $\omega_{c1}, \omega_{c2}, \dots$, если иметь соответствующие гармонические компонентные уравнения. Для этого необходимо представить $u(t)$ разложением в ряд Фурье на частоте, кратной частотам входных сигналов.

Пример 9.3. Пусть на НРТС действуют два гармонических входных сигнала $u_1(t) = v_{01} + v_{m1} \cos \omega_1 t$ и $u_2(t) = v_{02} + v_{m2} \cos \omega_2 t$, где v_{01}, v_{02} — постоянные составляющие.

Обозначим $\Phi_{\omega i}$ комплексные амплитуды узловых потенциалов на частоте ω_i и проследим, как видоизменяются уравнения схемы при решении различных задач в частотной области.

1. Уравнения схемы в линейном режиме

$$Ai(\varphi_0) = 0,$$

$$Ai(\varphi_0, \varphi_{\omega_1}) = 0,$$

$$Ai(\varphi_0, \varphi_{\omega_2}) = 0.$$

Первое уравнение соответствует расчету по постоянному току для определения положения рабочей точки φ_0 , а второе и третье, решаемые независимо друг от друга и в произвольном порядке, — расчету линеаризованной в этой точке схемы на частотах ω_1 и ω_2 .

2. Уравнения схемы в режиме большого сигнала

$$Ai_{\omega_1}(\varphi_0, \varphi_{\omega_1}, \varphi_{\omega_2}, \varphi_{2\omega_1}, \varphi_{2\omega_2}, \varphi_{\omega_1+\omega_2}, \varphi_{\omega_1-\omega_2}, \dots) = 0,$$

$$Ai_{\omega_2}(\varphi_0, \varphi_{\omega_1}, \varphi_{\omega_2}, \varphi_{2\omega_1}, \varphi_{2\omega_2}, \varphi_{\omega_1+\omega_2}, \varphi_{\omega_1-\omega_2}, \dots) = 0.$$

.....

Количество систем уравнений зависит от количества N высших гармоник и комбинационных частот, которое необходимо рассчитать в схеме.

3. Расчет схемы в нелинейном режиме малого сигнала. Для этого нужно сначала найти рабочую точку, линеаризовать в ней схему, выполнить расчет линеаризованной схемы в линейном режиме (как в п.1) на частотах ω_1 и ω_2 и найти гармонические напряжения на линеаризованных нелинейных элементах, соответствующие этим частотам. Далее по этим напряжениям нужно найти все гармонические и комбинационные составляющие токов нелинейных элементов, порождаемые найденными гармоническими напряжениями. После этого, рассматривая найденные токи как новые источники сигналов, заменяющие собой нелинейные элементы, и вновь рассчитывая линеаризованную схему в линейном режиме, можно найти спектр выходных сигналов.

Таким образом, данный метод позволяет выполнить все виды расчетов НРТС.

Остается рассмотреть вопрос о способе получения гармонических компонентных уравнений вида (9.28). Для этого нужно разложить функцию $i(t) = f(u(t))$ по гармоническим составляющим функции $u(t)$. Это разложение зависит от конкретного вида функции. В частности, его можно легко выполнить для распространенного случая — вольт-амперной характеристики p — n -перехода:

$$i = I_0 (e^{(u/m)\varphi_T} - 1).$$

Пусть напряжение на диоде $u = v_1 \cos \omega_1 t + v_2 \cos \omega_2 t$. Тогда

$$i = I_0 \left(\exp \left(\frac{v_1 \cos \omega_1 t + v_2 \cos \omega_2 t}{m\varphi_T} \right) - 1 \right).$$

Используя разложение экспоненты по бесселевым функциям J_n от мнимого аргумента

$$e^{a \cos x} = J_0(a) + 2J_1(a) \cos x + 2J_2(a) \cos 2x + \dots,$$

получаем для тока диода

$$i = I_0 \left[\left(J_0 \left(\frac{v_1}{m\varphi_T} \right) + 2 \sum_{p=1}^{\infty} J_p \left(\frac{v_1}{m\varphi_T} \right) \cos p \omega_1 t \right) \left(J_0 \left(\frac{v_2}{m\varphi_T} \right) + \dots \right) \right]$$

$$+ 2 \sum_{n=1}^{\infty} J_n \left(\frac{v_2}{m\varphi_T} \right) \cos n \omega_2 t \Big) - 1 \Big|.$$

Перемножая, находим гармонические компонентные уравнения диода для частот $p\omega_1 \pm n\omega_2$, где $p, n=0, 1, 2, \dots$:

постоянная составляющая $i_{00} = I_0 J_0(v_1/m\varphi_T) J_0(v_2/(m\varphi_T))$;

p -е гармоники частоты ω_1 : $i_{p\omega_1} = 2I_0 J_p(v_1/m\varphi_T) J_0(v_2/(m\varphi_T))$;

n -е гармоники частоты ω_2 : $i_{n\omega_2} = 2I_0 J_0(v/(m\varphi_T)) J_n(v_2/(m\varphi_T))$;

комбинационные составляющие с частотой $p\omega_1 \pm n\omega_2$: $i_{pn} = 2I_0 J_p(v_1/(m\varphi_T)) J_n(v_2/(m\varphi_T))$.

При воздействии не двух, а произвольного числа частот N , т. е. напряжения произвольной формы $u(t)$, представленного конечным рядом Фурье, $u(t) = u_0 + \sum_{k=1}^N v_k \cos k\omega t$, можно получить общую формулу

$$i_{pn \dots k} = 2 I_0 J_p \left(\frac{v_1}{m\varphi_T} \right) J_n \left(\frac{v_2}{m\varphi_T} \right) \dots J_k \left(\frac{v_k}{m\varphi_T} \right).$$

Аналогичный подход используется для построения гармонических компонентных уравнений более сложных элементов, описываемых экспоненциальными зависимостями или их комбинациями [41].

9.6. БЫСТРОЕ ПРЕОБРАЗОВАНИЕ ФУРЬЕ

Непрерывному прямому преобразованию Фурье

$$F(\omega) = \int_{-T/2}^{T/2} x(t) e^{-j\omega t} dt$$

можно поставить в соответствие его дискретный аналог. Пусть процесс $x(t)$ измерен на интервале $(0, T)$ в N равноотстоящих на Δt точках $t_0, t_1, \dots, t_k, \dots, t_{N-1}$. Тогда дискретное преобразование Фурье (ДПФ) описывается соотношением

$$F(\omega) = \sum_{k=0}^{N-1} x(t_k) e^{-j\omega t_k}. \quad (9.30)$$

Вводя обозначения $t_k = k\Delta t$, где k — автоматное время; $\omega = p(2\pi/T)$, где p — безразмерная частота; $T = N\Delta t$, можно переписать (9.30) в виде

$$F(p) = \sum_{k=0}^{N-1} x_k e^{j \frac{2\pi}{N} pk}, \quad p = 0, 1, \dots \quad (9.31)$$

Дискретное преобразование Фурье можно использовать для расчета спектра НРТС. Для этого нужно достаточно точно моделировать схему на интервале $(0, T)$, выбрать равноотстоящие точки t_k и по (9.31) рассчитать точки спектра $F(0), F(1), \dots$

Различают два способа численной реализации этой формулы — непосредственный и специальный, получивший название быстрого преобразования Фурье (БПФ).

Непосредственное ДПФ требует больших затрат времени, поэтому его можно применять лишь в тех случаях, когда эта операция встречается в программе редко, если же ее нужно выполнять часто или максимально быстро, как, например, в системах, работающих в реальном масштабе времени, то нужно применять БПФ.

Существуют различные алгоритмы БПФ, рассмотрим один из них [42]. Обозначим

$$W_N^{pk} = \exp \left(-j \frac{2\pi}{N} pk \right).$$

Разделим N отсчетов $x(0), \dots, x(N-1)$ на две равные половины $x(0), \dots, x(N/2-1)$ и $x(N/2), \dots, x(N-1)$. Тогда ДПФ (9.31) можно представить в виде

$$F(p) = \sum_{k=0}^{N/2-1} \left(x(k) W_N^{pk} + x\left(k + \frac{N}{2}\right) W_N^{p(k+N/2)} \right) = \sum_{k=0}^{N/2-1} x'_p(k) W_N^{pk}, \quad (9.32)$$

где $x'_p(k) = x(k) + x(k+N/2) W_N^{pN/2}$ — сложный отсчет. Учитывая, что для четных частот ($p=2r$) $W_N^{pN/2} = W_N^{rN} = 1$, а для нечетных частот ($p=2r+1$) $W_N^{pN/2} = -1$, видим, что сложный отсчет для четных частот получается суммированием, а для нечетных частот — вычитанием соответствующих значений $x(k)$. В результате из N первоначальных отсчетов находим две группы по $N/2$ сложных отсчетов — для четных и нечетных значений p . Далее аналогичная процедура половинных разбиений применяется для каждой из полученных групп сложных отсчетов, что приводит к четырем группам новых сложных отсчетов и соотношению

$$F(p) = \sum_{k=0}^{N/4-1} x''_p(k) W_N^{pk},$$

где $x''_p(k) = x'_p(k) + x'_p(k+N/4) W_N^{pN/4}$ — новый сложный отсчет, а весовые коэффициенты определяются как

$$W_N^{pN/4} = \begin{cases} 1 & p = 4r, \\ \exp(-j\pi/2) & p = 4r+2, \\ \exp(-j\pi/4) & p = 4r+1, \\ \exp(-j3\pi/4) & p = 4r+3. \end{cases}$$

Группируя новые отсчеты $x''_p(k)$ по признаку равных весовых коэффициентов, получаем четыре группы по $N/4$ отсчетов в каждой. Продолжая этот процесс до тех пор, пока не сформируется N групп по одному сложному отсчету в каждой, найдем N точек спектра $F(0), \dots, F(N-1)$. Чтобы группы при каждом делении были одинаковыми, обычно берут $N=2^n$, тогда БПФ выполняется за n описанных процедур (итераций). Очевидно, общее число операций в БПФ гораздо меньше, чем в непосредственном ДПФ, так

как в последующих итерациях учитываются действия, выполненные в предыдущих. Отметим также, что если в непосредственном ДПФ спектр вычисляется последовательно точка за точкой, то в БПФ расчет выполняется параллельно и спектр образуется полностью только на последней итерации. В настоящее время в связи с широким применением преобразования Фурье в автоматизированных системах научных исследований для его выполнения разработаны спецвычислители, в которых алгоритмы БПФ реализуются аппаратно.

ГЛАВА 10

МОДЕЛИРОВАНИЕ ПОЛУПРОВОДНИКОВЫХ ПРИБОРОВ

10.1. ОСНОВНАЯ ЗАДАЧА МОДЕЛИРОВАНИЯ КОМПОНЕНТОВ И КЛАССИФИКАЦИЯ МОДЕЛЕЙ

Математические модели активных и пассивных компонентов радиоэлектронных схем: транзисторов, диодов, резисторов, конденсаторов, катушек индуктивностей, источников питания и внешних сигналов — в значительной степени определяют эффективность использования САПР [43—45]. От них зависят как адекватность результатов моделирования, так и затраты машинного времени. Поэтому, несмотря на большое внимание, которое с начала развития автоматизированного проектирования уделялось моделям компонентов, 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 компонентов радиоэлектронных схем, включающие модели типовых элементов. Такими моделями можно пользоваться, автоматически включая их в общую математическую модель устройства. Однако при этом необходимо представлять особенности и ограничения используемых моделей.

Модели компонентов радиоэлектронных схем классифицируются по ряду разнохарактерных признаков: назначению, способу получения параметров, принципу описания процессов, форме представления, степени учета воздействия внешних условий. Основными классификационными признаками следует считать характеристики функционирования (см. рис. 3.1). По этим признакам модели подразделяют на динамические и статические, нелинейные и линейные, для большого и малого сигналов, высокочастотные и низкочастотные. Следует отметить, что эти признаки служат для организации иерархических рядов моделей, различающихся затратами машинного времени, зависящими от сложности, но допускающих пересчет параметров и переход от одной модели к другой в процессе моделирования. Очевидно, что наиболее точной и

сложной в таком иерархическом ряду является динамическая нелинейная высокочастотная модель для большого сигнала.

По принципу описания внутренних процессов в компонентах и по системе исходных параметров модели делятся на электрические, физико-топологические и технологические. В электрических моделях для описания внутренних процессов применяются токи и напряжения, а также общепринятые электрические параметры, такие, например, как коэффициент усиления, крутизна характеристики, сопротивление. В физико-топологических моделях описываются внутренние процессы перераспределения зарядов в одной или нескольких областях полупроводниковой структуры и используются такие параметры, которые характеризуют геометрические размеры областей, профиль примесей в структуре компонента, подвижность носителей заряда и т. п. Технологические модели включают описания технологических операций изготовления компонента — время и температуру диффузии, характеристики диффузанта и исходного материала.

Значение физико-топологических и технологических моделей возрастает по мере развития микроэлектроники. Однако в схемотехническом проектировании они находят непосредственное применение лишь в специализированных программах при отработке самих компонентов или малых фрагментов ИС, а также служат для исследований специальных условий их работы.

Электрические модели, получившие широкое развитие при расчете схем на дискретных компонентах, достаточно эффективны и при моделировании ИС. В этом случае для связи с условиями изготовления при определении параметров электрических моделей компонентов используются физико-топологические или даже технологические модели компонентов.

Имеет смысл выделить следующие формы представления моделей компонентов: эквивалентные схемы, аналитические выражения, системы уравнений и логических условий, таблицы. В современных системах АСХП часто используются смешанные формы представления. Например, модель представлена эквивалентной схемой, содержащей управляемые элементы (источники тока, резисторы, конденсаторы), часть которых задана аналитическими выражениями, а другая часть — таблицами.

Модели компонентов могут учитывать воздействия внешних условий, таких как температура, давление, влажность, радиация, эффекты старения. Описания указанных воздействий играют существенную роль при моделировании схем, предназначенном для замены натурного макетирования. Однако эти описания могут включаться в математическую модель схемы как отдельные управляемые «компоненты» и не обязательно должны содержаться внутри моделей компонентов исходной схемы.

Важными характеристиками каждой модели компонента являются ее точность и оценки требуемых машинных ресурсов. Однако эти характеристики зависят не только от структуры модели и выбранных уравнений режимно-зависимых элементов, но и от чис-

ленных значений параметров модели, а также от конкретной реализации в программе АСХП. Об этом необходимо помнить при проведении моделирования с целью получения не только качественных, но и правильных количественных оценок.

10.2. ЭЛЕКТРИЧЕСКИЕ МОДЕЛИ

10.2.1. Модели пассивных элементов. При моделировании схем на дискретных компонентах в качестве моделей пассивных компонентов используются, как правило, константы, соответствующие их номиналам. В ИС для повышения точности расчетов в ряде случаев целесообразно применять более сложные модели.

Резисторы ИС — это либо определенная область одного из диффузионных слоев в монокристаллических схемах, либо пленочные резисторы в гибридных схемах, либо резисторы на основе МДП-структур (в МДП ИС).

При моделировании интегральных резисторов следует учитывать, что они представляют собой распределенные RC -структуры. Паразитная емкость RC -структуры определяется зарядной емкостью изолирующего p — n -перехода в монокристаллических схемах, а при диэлектрической изоляции и для пленочных резисторов — емкостью резистивного слоя относительно подложки. Влияние паразитной распределенной емкости обычно описывается введением в модель шунтирующего конденсатора с постоянным значением емкости, равным среднему значению емкости распределенной RC -структуры. Иногда параллельно резистору в модели, кроме конденсатора, подключают паразитный транзистор. Параметры такой модели рассчитываются из технологических данных.

Распределенные RC -структуры находят специальное применение в гибридных ИС благодаря функциональному совмещению резисторов и конденсаторов, что позволяет уменьшить размеры и массу этих ИС.

В ИС RC -структуры используются в виде ускоряющих, интегрирующих и дифференцирующих цепей (рис. 10.1, a — $в$), в виде фильтров и других функциональных элементов. В машинных рас-

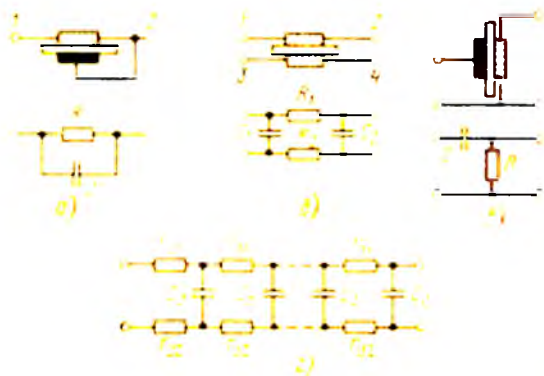


Рис. 10.1 RC -структуры:
 a — ускоряющая цепь;
 $б$ — линия задержки
 или интегрирующая
 цепь; $в$ — дифференци-
 рующая цепь в виде
 сосредоточенных моде-
 лей; $г$ — распределен-
 ная модель

четах можно использовать многосекционную модель RC -структуры (рис. 10.1,2) или приближенную модель с сосредоточенными параметрами, которые аппроксимируют требуемые характеристики [45].

В моделях интегральных конденсаторов следует учитывать зависимость емкости от режима, а также пробивное напряжение, которое зависит от технологии и может оказаться недостаточно большим. Паразитным элементом диффузионных и пленочных конденсаторов является сопротивление потерь материала обкладки.

10.2.2. Модели полупроводниковых диодов. Полупроводниковые диоды применяются как в дискретных, так и в интегральных цифровых и аналоговых схемах. При этом весьма разнообразны требования к их статическим и динамическим параметрам: напряжению отпираания, мощности, инерционности. В ИС обычно применяют в качестве диодов транзисторы в одном из пяти возможных диодных включений, имеющие различные характеристики.

Эквивалентная схема модели диода показана на рис. 10.2,а. Она содержит управляемый источник тока I_d , сопротивление диода R_d при прямом смещении, складывающееся из сопротивлений областей базы диода, омических контактов и выводов, сопротивление утечки R_y и емкость диода C_d .

Источник тока, управляемый напряжением, может быть описан выражением, соответствующим кусочно-линейной аппроксимации (рис. 10.2,б) характеристики диода, и рассматриваться в этом случае как упрощенная статическая модель диода, где E_0 , α_1 , α_2 — параметры модели.

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

$$I = I_0 \left| \exp \left(\frac{U}{m \varphi_T} \right) - 1 \right|, \quad (10.1)$$

$$C_d = C_{бар} + C_{диф}, \quad (10.2)$$

$$C_{бар} = C_\phi (\varphi - U)^{-1/2}, \quad (10.3)$$

$$C_{диф} = \left[I_0 \exp \left(\frac{U}{m \varphi_T} \right) \right] \left(\frac{\tau}{m \varphi_T} \right)^{-1}, \quad (10.4)$$

где I_0 — тепловой ток; U — напряжение на p — n -переходе; m — поправочный коэффициент, учитывающий отклонение реальной характеристики от идеальной теоретической характеристики p — n -перехода; φ_T — температурный потенциал; φ — контактная разность потенциалов, определяемая технологией изготовления; C_d , $C_{бар}$, $C_{диф}$ — суммарная, барьерная и диффузионная емкости соответственно; τ — временной коэффициент, учитывающий предельную частоту работы диода.

Параметры полупроводниковой структуры, входящие в модель, определяются из физических соображений: для нормальной темпе-

ратуры $\varphi_T = 0,026$ В, для кремниевых $p-n$ -переходов $\varphi = 0,7 \dots 0,75$ В, а для германиевых $\varphi = 0,4 \dots 0,6$ В.

Параметры I_0 , m , R_d вычисляются из условия аппроксимации статической характеристики диода (рис. 10.2, в) выражением

$$I_j = I_0 \left(\exp \frac{U_j - I_j R_d}{m \varphi_T} \right), j = 1, 2, \dots, N, \quad (10.5)$$

где (I_j, U_j) — соответствующие j -й экспериментальной точке характеристики диода значения тока и напряжения на диоде; N — число известных экспериментальных точек. Параметры I_0 , m , R_d вычисляются, например, методом наименьших квадратов.



Рис. 10.2. Модель полупроводникового диода (а) и представление ее управляемого источника кусочно-линейной характеристикой (б) и нелинейной характеристикой (в)

Барьерная емкость $C_{бар}$ моделирует приращение пространственного заряда при изменении напряжения на $p-n$ -переходе. В предположении, что там отсутствуют подвижные носители заряда, а также для ступенчатой формы перехода $C_{бар}$ описывается выражением (10.3). Для современных приборов характерно плавное распределение примеси в $p-n$ -переходе, которое точнее описывается при использовании в этом выражении другого показателя степени вместо квадратного корня. Значение показателя степени ν (10.3) может быть получено путем подбора наилучшей аппроксимации экспериментальной зависимости емкости от напряжения при обратном смещении перехода.

Диффузионная емкость $C_{диф}$ отражает влияние перераспределения подвижных носителей. Следует напомнить, что диффузионная емкость является доминирующей при прямом смещении, а барьерная — при обратном. В связи с этим применение формулы (10.3) ограничивается при некотором значении $U > 0$. Введение ограничения необходимо при машинной реализации модели, так как возможная в итерационных процедурах ситуация $\varphi \leq U$ вызывает аварийный останов. Вместо введения ограничения по напряжению при расчете барьерной ем-

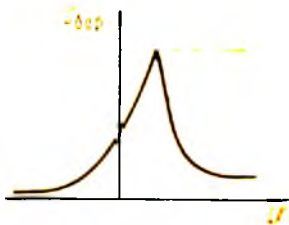


Рис. 10.3. Аппроксимация зависимости барьерной емкости $p-n$ -перехода от напряжения, применяемая в моделях компонентов

жости может использоваться кусочно-нелинейная аппроксимация зависимости $C_{бар}(U)$ или более сложная модель, например, в виде следующего выражения:

$$C_{бар} = \frac{C_0 \left(1 + \frac{a}{1-a} \frac{b}{x^N + b} \right)}{(x^2 + b)^{N/2}}, \quad (10.6)$$

где C_0 , a , b , x , N — параметры модели.

На рис. 10.3 приведена зависимость барьерной емкости от напряжения, рассчитанная по (10.6).

10.2.3. Модели биполярных транзисторов. В настоящее время промышленность выпускает большое число разнообразных типов транзисторов различного назначения. Кроме того, транзисторы являются одним из основных элементов ИС, причем их относительное число в РЭУ непрерывно растет, так как применение активных элементов более выгодно функционально, а стоимость их уменьшается в процессе развития технологии микроэлектроники. Следует отметить, что практически все программы АСХП имеют встроенные модели биполярных транзисторов (за исключением программ специализированного назначения). При этом, несмотря на множество работ, посвященных созданию моделей биполярных транзисторов, не удается получить одну «наилучшую» модель, пригодную для решения любых задач.

В качестве простейших моделей биполярных транзисторов используют схемы замещения уравнений четырехполюсника, представляющие собой формальную аппроксимацию его характеристик. Наибольшее распространение получили уравнения с y -параметрами и с h -параметрами (рис. 10.4):

$$\left. \begin{aligned} i_1 &= y_{11} u_1 + y_{12} u_2 \\ i_2 &= y_{21} u_1 + y_{22} u_2 \end{aligned} \right\}, \quad \left. \begin{aligned} u_1 &= h_{11} i_1 + h_{12} i_2 \\ i_2 &= h_{21} i_1 + h_{22} i_2 \end{aligned} \right\} \quad (10.7)$$

Такие модели с большой точностью описывают характеристики транзистора в линейном малосигнальном режиме, однако трудности их практического применения связаны с необходимостью в каждом конкретном случае определять численные значения параметров с учетом определенных рабочих точек.

При моделировании линейных электронных схем после расчета режима рабочей точки для сокращения машинного времени используются малосигнальные модели биполярных транзисторов. Эквивалентные схемы подобных моделей для включения транзистора с управлением по эмиттеру и по базе показаны на рис. 10.5, а, б. Параметры приведенных моделей — коэффициенты передачи α или β , сопротивления базы r_b , коллектора r_k , эмиттера r_a , коэффициент обратной связи по напряжению $\mu_{эк}$, емкости коллекторного и эмиттерного переходов C_k , C_a — соответствуют стандартизованным справочным значениям параметров биполярных транзисторов. При автоматизированном проектировании указанные параметры определяются, как правило, расчетным путем из нелинейных моде-

лей, а не являются исходными данными. Это необходимо для обеспечения моделирования приборов в различных режимах с наименьшими затратами труда человека на подготовку данных. Для интегральных транзисторных структур при получении параметров моделей необходимо также учитывать расположение и геометрическую форму областей, а также технологические данные.

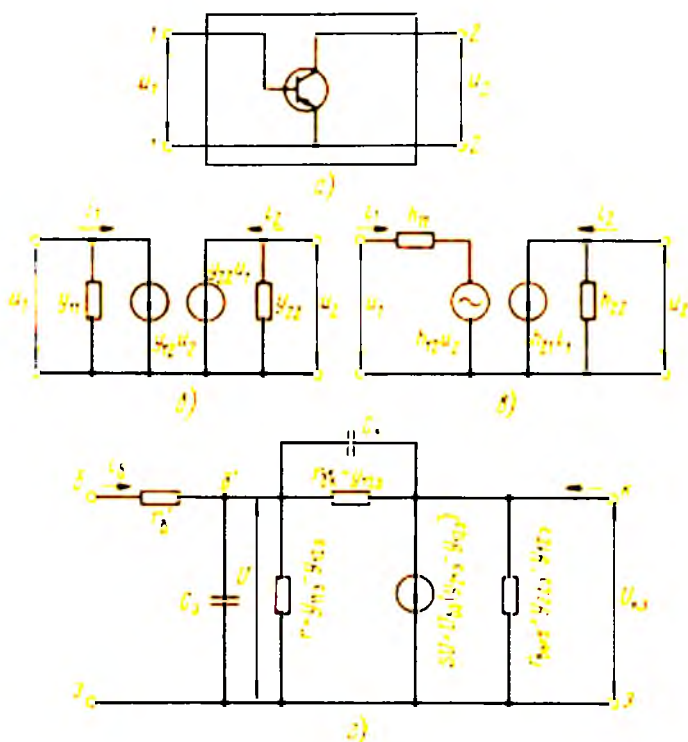


Рис. 10.4. Формальные модели транзистора как четырехполюсника

Пример 10.1. Линейная модель биполярного транзистора [44], предназначенная для автоматизированного проектирования приемно-усилительных устройств, приведена на рис. 10.5,а. Эта модель описывает работу маломощных СВЧ-транзисторов и включает ряд элементов, отражающих их характерные особенности: технологические, конструктивные, режимные.

Таковыми элементами являются: последовательные сопротивления $r_{бв}$ и $r_{эв}$, обусловленные сопротивлениями выводов и контактными явлениями, причем $r_{бв}$ сравнимо с омическим сопротивлением базы $r'_б$, а $r_{эв}$ — с дифференциальным сопротивлением эмиттера $r_э$ при больших токах; последовательное сопротивление в коллекторной цепи r_k , обусловленное растеканием тока в толще полупроводника и достигающее нескольких десятков Ом; сопротивление $R_{ш}$, шунтирующее коллекторный переход, моделирующее эффект Эрли; паразитная емкость $C_{эк0}$ между металлизированной площадкой для пайки эмиттерно-

то вывода и коллектором, оказывающая существенное влияние на устойчивость транзистора при включении с общей базой; межэлектродные емкости $C_{бэ}$, $C_{эк}$, $C_{бк}$. Кроме того, индуктивности выводов разделены на внутренние $L_{бв}$ и $L_{эв}$, определяемые индуктивностями токоведущих дорожек или золоченых проволочек, и внешние $L_б$, $L_э$, $L_к$, представляющие индуктивности внешних выводов, длины которых устанавливаются в соответствии с указанными в технических условиях минимальными расстояниями до места пайки.

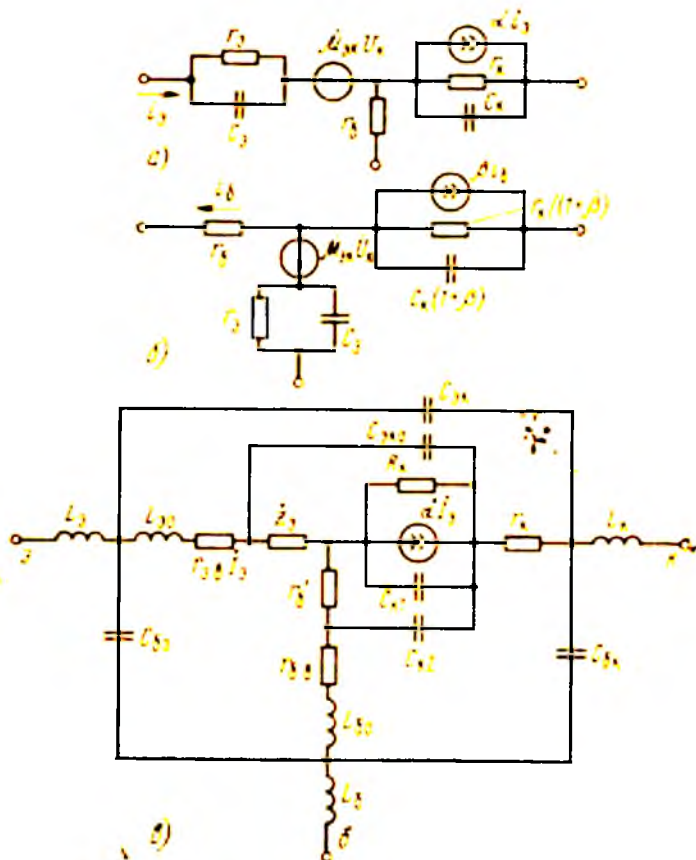


Рис. 10.5. Линейные модели биполярного транзистора:
а, б — для эмиттерного и базового управления; в — высокочастотная модель

В данной модели биполярного транзистора условно выделены две группы параметров: параметры собственно транзисторной структуры и параметры, характеризующие корпус. Основные параметры первой группы следующие: α — коэффициент передачи по току на высокой частоте для включения транзистора по схеме с общей базой; $z_б$ — полное комплексное сопротивление эмиттерного перехода; $r'_б$ — омическое сопротивление базы; $C_{к1}$ — активная емкость коллекторного перехода; $C_{к2}$ — пассивная емкость коллекторного перехода, а так-

же указанные выше параметры $r_{бв}$, $r_{эв}$, r_k , R_k , $C_{эко}$. Остальные параметры $L_{бэ}$, $L_{эо}$, $L_б$, $L_э$, L_k , $C_{бэ}$, $C_{эк}$, $C_{бк}$ составляют группу паразитных параметров корпуса.

Таким образом, данная модель имеет 18 параметров, два из которых являются частотно-зависимыми:

$$\alpha = \frac{r_{э0}}{1 + jf/f_T}, \quad \beta = \frac{r_{б0}}{1 + jf/f_T}, \quad (10.8)$$

где α_0 — низкочастотное значение α ; f_T — граничная частота усиления по току; $r_{э0} = 0,026/I_{э}$ — дифференциальное сопротивление эмиттерного перехода при токе $I_{э}$, соответствующем рабочей точке.

Кроме того, шесть параметров этой модели существенно зависят от режима работы по постоянному току. Для определения параметров модели по справочным данным и характеристикам разработан специальный алгоритм, результатом работы которого является запись полученных значений параметров в матрицу коэффициентов системы уравнений модели, составленную для приведенной эквивалентной схемы. Затем вычисляются y -параметры по стандартному алгоритму анализа электронных схем.

В качестве нелинейных моделей биполярного транзистора наибольшее распространение получили различные модификации модели Эберса — Молла, которая была предложена для дискретного транзистора в 1954 г., т. е. в самом начале развития транзистор-

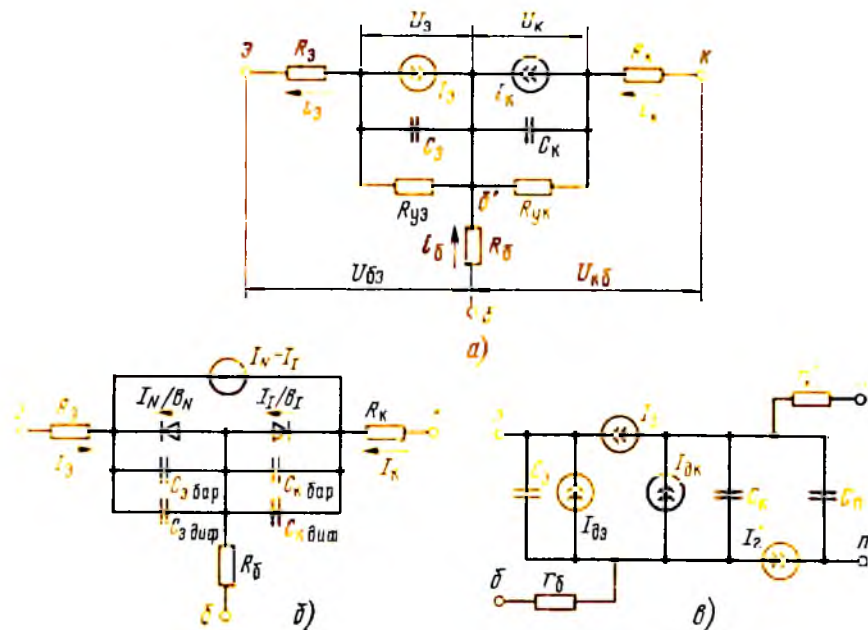


Рис. 10.6. Модифицированные модели Эберса — Молла:

а и б — инжекционная и передаточная для двухпереходного транзистора; в — для трехпереходного транзистора

ной техники (транзистор теоретически предсказан Шокли в 1948 г.). Рассмотрим один из вариантов такой модели, в основе которой лежит представление транзистора как системы двух взаимодействующих диодов. Эквивалентная схема модели приведена на рис. 10.6,а. В ней используются следующие обозначения: $I_{\text{э}}, I_{\text{к}}$ — источники внутренних токов, управляемые напряжениями на p — n -переходах; $r_{\text{э}}, r_{\text{б}}, r_{\text{к}}$ — сопротивления объемных областей эмиттера, базы и коллектора соответственно, а также их внешних выводов; $r_{\text{уэ}}, r_{\text{ук}}$ — сопротивления утечки; $C_{\text{э}}, C_{\text{к}}$ — емкости эмиттерного и коллекторного переходов.

Для расчета токов $I_{\text{э}}(U_{\text{э}}, U_{\text{к}})$ и $I_{\text{к}}(U_{\text{э}}, U_{\text{к}})$ используются модифицированные уравнения Эберса — Молла:

$$I_{\text{э}} = I_{\text{э0}} \left(\exp \frac{U_{\text{э}}}{m_{\text{э}} \varphi_{\text{T}}} - 1 \right) - \frac{B_{\text{I}}}{B_{\text{I}} + 1} I_{\text{к0}} \left(\exp \frac{U_{\text{к}}}{m_{\text{к}} \varphi_{\text{T}}} - 1 \right), \quad (10.9)$$

$$I_{\text{к}} = I_{\text{к0}} \left(\exp \frac{U_{\text{к}}}{m_{\text{к}} \varphi_{\text{T}}} - 1 \right) - \frac{B_{\text{N}}}{B_{\text{N}} + 1} I_{\text{э0}} \left(\exp \frac{U_{\text{э}}}{m_{\text{э}} \varphi_{\text{T}}} - 1 \right), \quad (10.10)$$

где $U_{\text{э}}$ и $U_{\text{к}}$ — напряжения на эмиттерном и коллекторном переходах (входные параметры модели); B_{N} и B_{I} — статические коэффициенты передачи по току в схеме с общим эмиттером; $m_{\text{э}}$ и $m_{\text{к}}$ — поправочные коэффициенты; $I_{\text{к0}}$ и $I_{\text{э0}}$ — обратные токи, определяемые при замкнутых накоротко цепях эмиттер — база и коллектор — база.

Поскольку для большинства транзисторов коэффициенты передачи существенно зависят от тока, в модель включаются уравнения, аппроксимирующие эти зависимости, например полиномами третьей степени [43]

$$B_{\text{N}} = B_{\text{N0}} + B_{\text{N1}} i_{\text{э}} + B_{\text{N2}} i_{\text{э}}^2 + B_{\text{N3}} i_{\text{э}}^3 \quad (10.11)$$

В ряде случаев в модель вводят также зависимость коэффициентов передачи от напряжения на переходах, задающую наклон коллекторных характеристик транзистора.

Для расчета емкостей модели, которые существенно зависят от напряжения и тока, используются следующие соотношения:

$$C_{\text{ээ}} = \frac{C_{\text{э0}}}{(\varphi_{\text{э}} - U_{\text{э}})^{0.5}} + \frac{I_{\text{э0}} r_{\text{э}} \exp [U_{\text{э}} / (m_{\text{э}} \varphi_{\text{T}})]}{2 \pi f_{\text{T}} m_{\text{э}} \varphi_{\text{T}}} \quad (10.12)$$

$$C_{\text{кк}} = \frac{C_{\text{к0}}}{(\varphi_{\text{к}} - U_{\text{к}})^{0.5}} + \frac{I_{\text{к0}} r_{\text{к}} \exp [U_{\text{к}} / (m_{\text{к}} \varphi_{\text{T}})]}{2 \pi f_{\text{T}} m_{\text{к}} \varphi_{\text{T}}} \quad (10.13)$$

В этих выражениях представлены суммарные барьерные и диффузионные емкости соответствующих p — n -переходов, смысл параметров которых рассмотрен выше для модели полупроводниковых диодов (см. п. 10.2.2).

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 или понизить степень полинома в (10.11), считать емкости постоянными, а в ряде схем вообще пренебречь ими при анализе на низких частотах, для маломощных транзисторов не учитывать сопротивление эмиттера и т. п. При моделировании высокочастотных схем в модель следует ввести индуктивности выводов. Для повышения точности необходимо учитывать зависимость сопротивления базы от режима, частотные зависимости коэффициентов передачи, лавинное умножение носителей в $p-n$ -переходе, а также моделировать влияние таких внешних факторов, как температура, излучение, давление на соответствующие параметры.

Другой часто используемый в программах АСхП вариант модели биполярного транзистора (рис. 10.6,б) известен как передаточная модель или как модель Гуммеля — Пуна. Если в первой рассмотренной модели в качестве токов управляемых источников приняты токи, инжектируемые $p-n$ -переходами, то в передаточной модели приняты токи, собираемые $p-n$ -переходами [43]:

$$I_N = I_S [\exp(U_a/\varphi_T) - 1], \quad (10.14)$$

$$I_I = I_S [\exp(U_k/\varphi_T) - 1], \quad (10.15)$$

где I_S — параметр модели, называемый током «отсечки». Этот параметр определяется по характеристикам транзистора. В цепь базы ответвляется ток

$$I_6 = I_N/B_N + I_I/B_I, \quad (10.16)$$

а выражения для токов через выводы эмиттера и коллектора имеют вид:

$$I_a = -(1 + 1/B_N)I_N + I_I, \quad (10.17)$$

$$I_k = I_N - (1 + 1/B_I)I_I. \quad (10.18)$$

Зависимости барьерных емкостей от напряжения описываются в модели Гуммеля — Пуна выражением (10.6), а в других модификациях передаточной модели так же, как и в инжекционной модели. Для диффузионных емкостей используется следующее представление:

$$C_{э \text{ диф}} = (\tau_N/\varphi_T) (I_N + I_S), \quad C_{к \text{ диф}} = (\tau_I/\varphi_T) (I_I + I_S), \quad (10.19)$$

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 и определяется степенью адекватности используемых параметров.

Пример 10.2. Рассмотрим модель интегрального трехпереходного транзистора, предложенную И. П. Норенковым [9]. Эквивалентная схема модели

показана на рис. 10.6,в. Нелинейные ветви схемы описываются следующими уравнениями:

$$C_i = C_{\text{б}0i} [\varphi_0 / (\varphi_0 - U_i)]^{n_i} + \frac{\tau_i}{m_i \varphi_T} (I_{\partial i} + I_{Ti}),$$

$$I_{\partial i} = I_{Ti} \{ \exp [U_i / (m_i \varphi_T)] - 1 \},$$

$$I_T = B I_{\partial \text{э}} - B_{\text{п}} I_{\partial \text{к}}, \quad I_{\Gamma} = B_{\text{п}} I_{\partial \text{к}},$$

где индекс i принимает значения {э, к, п} и характеризует принадлежность параметра переходу эмиттера, коллектора или подложки; $C_{\text{б}0i}$ — барьерная емкость при нулевом напряжении; n_i — коэффициент, равный 0,5 или 0,33 в зависимости от характера перехода; τ_i — постоянная времени перехода; I_{Ti} — тепловой ток; $r_{\text{б}}$ и $r'_{\text{к}}$ — объемные сопротивления тел базы и коллектора; B и $B_{\text{п}}$ — нормальный и инверсный коэффициенты передачи тока базы основного транзистора; $B_{\text{п}}$ — коэффициент передачи тока паразитного транзистора.

Ветвь $I_{\partial \text{э}}$ отражает статическую вольт-амперную характеристику эмиттерного перехода в виде зависимости части тока, ответвляющейся в базу, от напряжения на переходе. Аналогичный смысл имеет ветвь $I_{\partial \text{к}}$. Источник I_T представляет ток неосновных носителей, инжектируемых в базу, а источник I_{Γ} — ток неосновных носителей, инжектируемых в коллектор коллекторным переходом при условии, что переход коллектор-подложка изолирующий и всегда закрыт. Остальные ветви имеют тот же смысл, что и в ранее описанных моделях.

Рассмотренные модели биполярного транзистора отражают наиболее общие принципы его работы. Существует множество разновидностей биполярных структур, для которых требуется модификация моделей, направленная на учет их специфических особенностей. К ним относятся торцевой, или горизонтальный, транзистор, в модели которого необходимо учитывать два паразитных вертикальных транзистора. Другим примером является многоэмиттерный транзистор, широко используемый в ИС, имеющий паразитный межэмиттерный транзистор. К специальным транзисторным структурам относятся также транзистор Шотки, транзистор с инжекторным переходом (многоколлекторный), однопереходный транзистор и т. д. Приведенные примеры показывают, что невозможно получить точную универсальную электрическую модель биполярных транзисторов. Для автоматизированного проектирования РЭУ необходима разработка иерархического ряда моделей, а также диалоговых и автоматических процедур формирования моделей, имеющих требуемые свойства.

10.2.4. Модели МДП-транзисторов. Эти приборы имеют широкое распространение, в особенности в качестве элементов ИС и БИС. МДП-транзисторы относятся к униполярным приборам, модели которых проще, чем биполярных транзисторов. Основным элементом моделей является источник тока канала, функциональная зависимость которого от напряжений на выводах характеризует активное действие униполярного транзистора.

Для приближенных расчетов достаточно упрощенная модель МДП-транзистора (рис. 10.7) для малых сигналов, в которой не учитывается влияние объемных сопротивлений стока (с), затвора (з) и подложки (п), а также воздействие подложки на изменение тока канала. Параметры малосигнальной модели определяются в заданной рабочей точке приращениями токов и напряжений в пологой области характеристик МДП-транзистора, т. е. источник тока, задающий приращение тока канала, $i_k = S U_{зи}$, где S — крутизна вольт-амперной характеристики по управляющему затвору при заданном напряжении на стоке. Влияние внутренней обратной связи, возникающей через объемное сопротивление истока (и), учитывается переменной величиной крутизны [45].

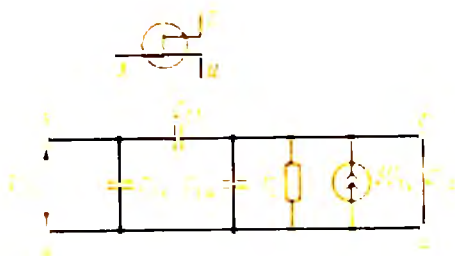


Рис. 10.7. Эквивалентная схема модели МДП-транзистора для малого сигнала

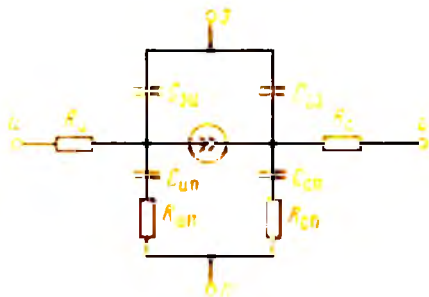


Рис. 10.8. Эквивалентная схема модели МДП-транзистора для большого сигнала

Модель интегрального МДП-транзистора для режима большого сигнала показана на рис. 10.8 [37]. Основным элементом модели является источник тока I_c , функциональная зависимость которого от напряжений на выводах характеризует активное действие транзистора. 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 для крутой и пологой областей вольт-амперной характеристики описывается выражениями:

$$I_c = \begin{cases} \left| \frac{M_0}{1 + \theta(U_{зи} - U_0)} \right| (U_{зи} - U_0) U_{си} - \frac{U_{си}^2}{2} (1 + \delta) & \text{при } U_{си} \leq U_{нас}, U_{зи} > U_0; \\ I_{нас} \left(1 - \frac{|U_{си}^2 + 2A(1 + BI_{нас})(U_{си} - U_{нас})|^{1/2} - U_{нас}}{A(1 + BI_{нас})} \right) & \text{при } U_{си} > U_{нас}, U_{зи} > U_0, \end{cases} \quad (10.20)$$

где точка $(I_{нас}, U_{нас})$ соответствует напряжению на стоке, при котором заряд в инверсионном слое равен нулю:

$$I_{нас} = \frac{M_0 U_{си}}{1 + \theta(U_{зи} - U_0)} \{ [U_{си}^2 (1 + \delta)^2 + (U_{зи} - U_0)^2]^{1/2} - U_{си} (1 + \delta) \}. \quad (10.21)$$

$$U_{\text{нас}} = \frac{U_{\text{зи}} - U_0}{1 + \delta} + U_{\text{си}} \left[\left(\frac{U_{\text{зи}} - U_0}{1 + \delta} \right)^2 + U_{\text{си}}^2 \right].$$

Коэффициенты δ , $U_{\text{си}}$, M_0 , θ , A , B в приведенных выражениях являются статическими параметрами модели. Динамическими параметрами служат значения емкостей затвор—сток $C_{\text{зс}}$, затвор—исток $C_{\text{зи}}$, сток—подложка $C_{\text{сп}}$, исток—подложка $C_{\text{ип}}$, которые, как правило, принимаются постоянными.

В данной модели при соответствующем определении параметров учитываются практически все известные физические эффекты в работе МДП-транзистора: зависимость подвижности носителей в канале от напряженности электрического поля, влияние потенциала подложки, конечность выходного сопротивления в области насыщения, паразитные сопротивления истока и стока. Эта модель справедлива также и при инверсном режиме работы МДП-транзистора, когда исток и сток меняются местами. Следует отметить, что в выражения для указанных параметров моделей МДП-транзисторов входят внутренние параметры структуры прибора, такие как длина и ширина канала, подвижность носителей, удельная емкость диэлектрика под затвором, концентрация примеси, толщина диэлектрика.

10.3. ФИЗИКО-ТОПОЛОГИЧЕСКИЕ МОДЕЛИ

Разработчикам схем нужны лишь количественные описания внешних электрических характеристик приборов, которые имели бы требуемую точность и были достаточно простыми. Сложные модели полупроводниковых элементов ИС, учитывающие особенности структуры: геометрические размеры и форму областей, параметры p — n -переходов, изолирующие слои, расположение и форму контактных площадок (рис. 10.9), концентрацию и подвижность носителей заряда, электрофизические характеристики материалов — предназначены в основном для разработчиков приборов. Однако с целью учета схемных критериев при отработке небольших компоновочных блоков ИС используют схемотехническое проектирование с применением физико-топологических моделей. Кроме того, подобные модели могут быть полезны для получения параметров моделей более простой структуры или макромоделей фрагментов ИС.

В основе физико-топологических моделей лежат уравнения непрерывности потоков носителей заряда: электронов и дырок. Это уравнения переноса и Пуассона, которые решаются численными методами в одномерном или двухмерном представлении на дискретной сетке, построенной на сечении анализируемой области полупроводника. В уравнения входят электростатический потенциал Ψ и квазиуровни Ферми Φ_n , Φ_p , определяемые концентрацией носителей, такими параметрами, как подвижность, время жизни, а также распределением концентрации примесей по структуре полупроводника в рассматриваемой области. Кроме того, при

решении необходимо задание граничных условий, характеризующих поведение носителей заряда на контактах и поверхностях раздела областей [46].



Рис. 10.9. Структура биполярного транзистора (а) и МДП-транзистора (б)

При анализе переходных процессов пространственная дискретизация и неявное интегрирование по времени получаемых уравнений в частных производных, которые представляют собой модель прибора, для каждого момента времени дают нелинейную систему уравнений вида

$$\begin{aligned} F_{\Psi}(\Psi^m, \Phi_n^m, \Phi_p^m, t^m) &= 0, \\ F_n(\Psi^m, \Phi_n^m, \Phi_p^m, t^m) &= 0, \\ F_p(\Psi^m, \Phi_n^m, \Phi_p^m, t^m) &= 0, \end{aligned} \quad (10.22)$$

где t^m — момент времени, соответствующий m -му шагу. При анализе статических режимов система аналогичного вида получается для каждого набора внешних смещений на выводах прибора.

В обоих случаях при моделировании приходится решать систему из $3N$ нелинейных уравнений, где N — число точек сетки, составляющее в двумерном случае обычно от 300 до 4000. Решение, причем многократное, такой большой системы уравнений является наиболее трудоемкой процедурой в процессе моделирования полупроводниковых приборов.

В наиболее простом варианте каждое из дифференциальных уравнений (10.22) решается независимо. Сначала решается первое уравнение (уравнение Пуассона) в предположении известных

квазиуровней Ферми, для которых задается начальное приближение. Полученные значения электростатического потенциала Ψ подставляются в уравнения непрерывности (второе и третье уравнения в (10.22)). Этот процесс последовательно повторяется до тех пор, пока для всех неизвестных переменных не будут получены совпадающие с заданной точностью последовательные значения

(рис. 10.10). Этот метод имеет преимущество в многомерном моделировании, состоящее в том, что он экономичен (относительно памяти ЭВМ) и хорошо сходится во всех случаях, когда указанные три основных уравнения слабо связаны между собой. При удачном выборе начального приближения для квазипотенциала Ферми основных носителей результат уже первого цикла итерационного процесса оказывается близким к искомому решению.

Каждая из трех систем (10.22) чаще всего нелинейна, поэтому для их решения применяются итерационные методы. Обычно выбирается метод Ньютона, обладающий квадратичной сходимостью.

Для решения задач моделирования полупроводниковых приборов разработаны также методы одновременного решения указанных систем уравнений, в которых применяются LU -разложение, упорядочение строк и столбцов матрицы Якоби, специальные

методы работы с ленточными матрицами, а также ряд других приемов с целью преодоления проблем сходимости и повышения эффективности моделирования.

10.4. ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ МОДЕЛЕЙ

10.4.1. Особенности получения параметров моделей. Определение параметров моделей является важнейшей задачей, от решения которой зависит эффективность внедрения программ АСХП в практику проектирования. Справочные данные и технические условия на приборы не содержат полной информации для определения параметров моделей, применяемых в программах АСХП. Цель настоящего раздела состоит в том, чтобы показать, что методики и программы определения параметров моделей должны

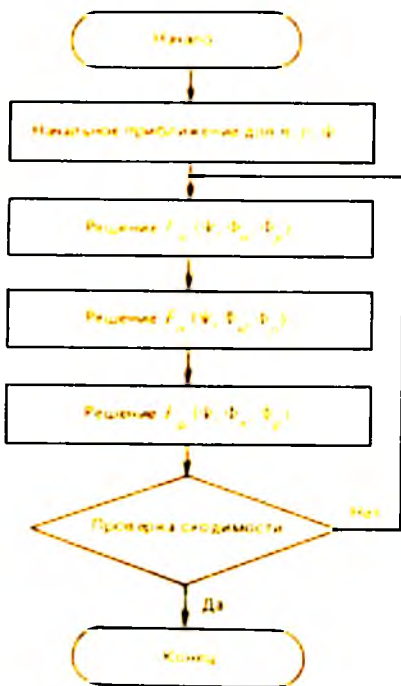


Рис. 10.10. Схема алгоритма Гуммеля для численного решения уравнений физических процессов

быть составной частью САПР. Это связано с созданием измерительно-вычислительных комплексов и специального программного обеспечения. Конкретные методики определения параметров для практически используемых в АСхП моделей, имеющих, как отмечалось выше, порядка десятков параметров для активных элементов, рассматриваются в специальной литературе по моделям, а также должны входить в комплект документации на САПР.

В связи с технологическим разбросом характеристик реальных полупроводниковых приборов (например, для многих дискретных транзисторов допускается разброс коэффициента усиления от 20 до 80) нет гарантии, что параметры модели, полученные в результате измерений и аппроксимации характеристик одного прибора, будут справедливы и для других приборов данного типа. Для параметров моделей полупроводниковых приборов наиболее достоверными являются их среднестатистические значения, рассчитанные по достаточно представительной выборке приборов.

Для получения параметров моделей статистической обработкой необходимо провести с достаточно высокой точностью большое количество измерений. Например, при расчете параметров вольт-амперной характеристики диода, позволяющих аппроксимировать эту характеристику в нелинейной модели с точностью порядка 1%, требуется приблизительно 20 экспериментальных точек, а при получении параметров статической нелинейной модели биполярного транзистора — порядка 150...200 экспериментальных точек. В связи с этим задача получения параметров моделей должна решаться с использованием современных технических средств автоматизированных измерений и средств вычислительной техники, используемых для хранения и обработки информации.

В ряде случаев, в частности для элементов ИС, при получении параметров моделей используются не экспериментальные данные, а результаты моделирования по более точным моделям. Например, результаты моделирования характеристик прибора на основе физико-топологической модели используются при определении параметров электрической модели, применяемой в программах АСхП, или результаты моделирования характеристик по сложной нелинейной модели прибора используются при определении параметров простой линейной модели в рабочей точке после расчета статического режима схемы, которые затем применяются при моделировании в режиме малого сигнала по программам линейного анализа.

Однако необходимо помнить, что в любом методе определения параметров моделей критерием их достоверности является степень совпадения результатов моделирования с реальными характеристиками. На основе этого положения разрабатываются методики определения параметров моделей полупроводниковых приборов, в которых часть величин находится методами формальной оптимизации, а в качестве критериев оптимальности выбирается заданная количественная мера оценки совпадения расчетных и экспериментальных характеристик.

10.4.2. Процедуры определения параметров моделей. Анализ параметров электрических моделей различных полупроводниковых приборов, наиболее употребительных в АСхП (выпрямительных, импульсных и других диодов, стабилитронов, биполярных транзисторов, полевых и МДП-транзисторов, тиристоров), а также методик получения параметров моделей, показывает, что можно выделить ряд основных задач, решение которых необходимо при формировании информационного обеспечения САПР.

1. Получение статических вольт-амперных характеристик типа $U(I)$, $I(U)$, $U_{\text{вых}}(U_{\text{вх}})$, $I_{\text{вых}}(I_{\text{вх}})$ и динамических вольт-фарадных $C(U)$ или импульсных характеристик. Характеристики определяются в результате либо измерений на лабораторных макетах или на автоматизированных измерительных системах, управление которыми обеспечивается программным способом, либо моделирования на ЭВМ с использованием моделей более высокой степени сложности и точности.

2. Документальное оформление результатов измерений или моделирования для достаточно представительной выборки приборов с использованием автоматизированных средств печати и графической обработки данных. Наиболее эффективной является непосредственная связь автоматизированных установок с ЭВМ.

3. Обработка результатов измерений или моделирования по специальным программам на ЭВМ с целью расчета параметров моделей по принятым методикам, статистической обработки, оценки точности заданных моделей и полученных наборов параметров, оптимизации набора параметров.

4. Организация хранения и поиска наборов параметров для заданных типов электрорадиоэлементов и выбранных моделей, а также выбора или пересчета параметров на основании режимных зависимостей, температуры и других внешних условий.

На рис. 10.11 приведена одна из возможных структур программного обеспечения для решения рассмотренных задач, в которой выделено пять уровней программ. Дугами указаны программы, обеспечивающие решение задач внутри одного уровня, а стрелками показана последовательность алгоритмических переходов между уровнями.

На уровне I формируется библиотека программ автоматизированных испытаний, управляющих источниками питания и сигналов в процессе измерений. Это программы многократного использования, так как параметры моделей получают из ограниченного набора характеристик. Результаты испытаний записываются в базу данных под шифром типа прибора и затем используются при расчете, анализе и оптимизации параметров соответствующих моделей на остальных уровнях.

Ко II уровню относятся программы, реализующие конкретные методики расчета параметров, а также вспомогательные программы, реализующие многократно встречающиеся процедуры, например расчета коэффициентов аппроксимирующих функций, решения

линейных уравнений, специальных систем нелинейных уравнений и т. д.

Программы уровня III предназначены для анализа параметров моделей. Это реализация процедур статистической обработки данных: расчет математического ожидания, дисперсии, оценок доверительных интервалов, коэффициентов корреляции параметров, а:

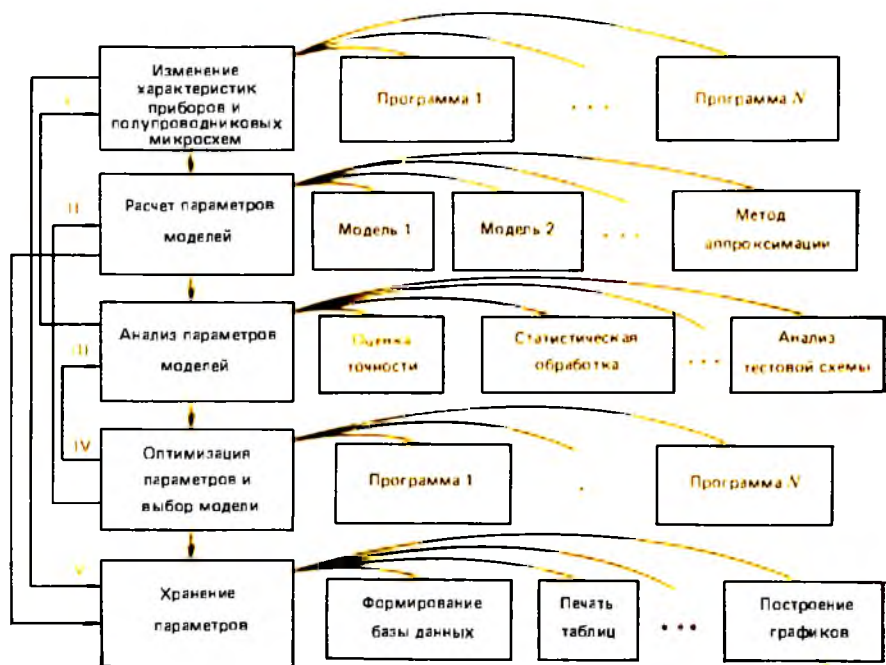


Рис. 10.11. Структура программного обеспечения для определения параметров моделей

также процедур оценки точности совпадения экспериментальных и аппроксимирующих функций: абсолютные и относительные отклонения, среднеквадратическая ошибка, минимаксные критерии. Кроме того, к этому уровню относятся процедуры анализа тестовых схем для оценки полного набора параметров модели.

К процедурам оптимизации уровня IV относятся как программы формальной параметрической оптимизации, так и процедуры, реализующие переход к более сложным и точным моделям в иерархической структуре моделей, предусмотренных в системе. Кроме того, следует обеспечить возможность автоматического формирования кусочных и табличных моделей путем получения параметров моделей для ряда поддиапазонов рабочих токов и напряжений или таблиц для заданных диапазонов.

Программы уровня V предназначены для обеспечения записи и выдачи информации о параметрах моделей и их оценках для

различных типов приборов. Они реализуют процедуры распечатки документации в виде таблиц и графиков, выдачи графической информации на специальные устройства, формирования и обслуживания базы данных параметров моделей на магнитных дисках и лентах.

10.4.3. Структура информационного обеспечения АСхП. Создание информационного обеспечения в виде базы данных, содержащей описания моделей полупроводниковых приборов, численные значения параметров этих моделей и их оценки для широкого круга типов полупроводниковых приборов с учетом их статистического характера, является важной задачей практического внедрения САПР РЭА. Наиболее рационально создание диалоговых информационных систем, способных взаимодействовать как с человеком, так и с другими подсистемами САПР.

Диалоговая справочная система по параметрам моделей полупроводниковых приборов должна обеспечивать режимы считывания, записи и коррекции информации, формирование и выдачу каталогов, подготовку к работе информационных файлов. В режиме считывания основные функции следующие: получение зна-

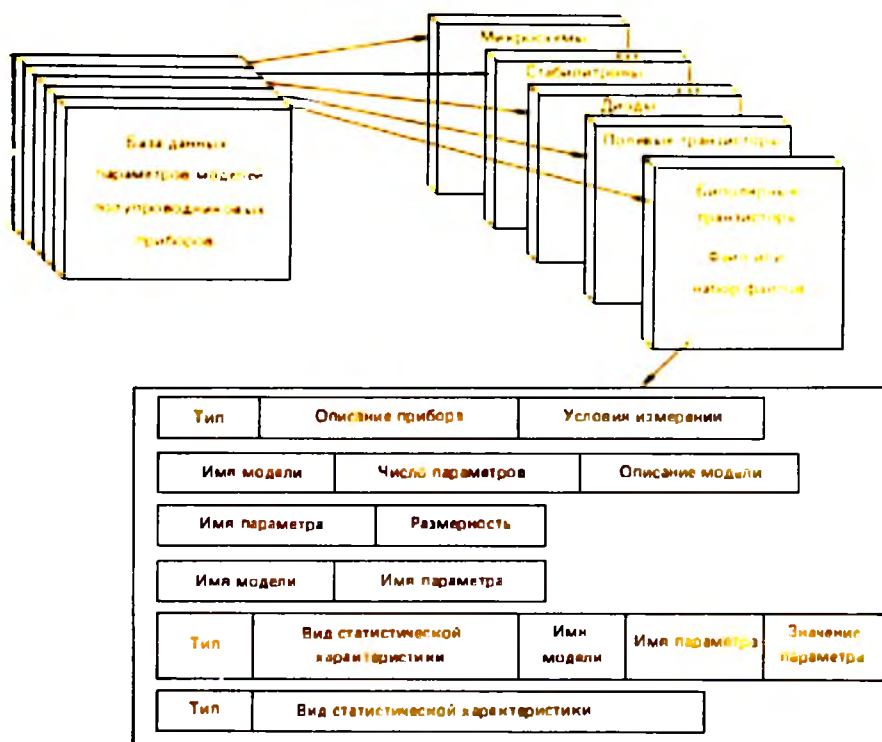


Рис. 10.12. Структура информационного обеспечения АСхП, содержащего описания моделей и их параметры

чения одного или группы параметров заданной модели прибора определенного типа, получение статистических характеристик значений параметров для партии приборов определенного типа, подборка прибора с определенными классификационными признаками, нахождение прибора с требуемыми значениями одного или нескольких параметров, получение классификационного описания прибора и сведений о наличии параметров моделей и их оценках, получение данных об условиях измерений базовых характеристик при определении параметров моделей, получение описаний моделей или их расчетных модулей.

Данные, которые хранятся в подобной информационной системе, следующие: типы приборов, классификационное описание приборов, которое включает сведения о применении, способе изготовления, типе проводимости, мощности рассеяния и т. п., данные об условиях исследования характеристик при определении параметров моделей, наименования моделей, описания моделей текстовые, перечни моделей, численные значения параметров, статистические характеристики параметров. На основе анализа связей указанной информации строится ее логическая структура, которая затем оптимизируется с целью устранения избыточности (рис. 10.12). Эта логическая структура может различными способами отображаться в конкретную базу данных, что связано с используемыми средствами вычислительной техники, операционными системами и системами управления базами данных, принятыми в рамках САПР. В простейшем случае файловая структура базы данных получается прямым отображением логической структуры на физический носитель: магнитный диск или ленту. Тогда каждому отношению, приведенному на рис. 10.12, ставится в соответствие физический файл.

ГЛАВА 11

МАКРОМОДЕЛИ

ИНТЕГРАЛЬНЫХ МИКРОСХЕМ

11.1. ПОСТАНОВКА ЗАДАЧИ

11.1.1. Понятие макромоделей в АСхП. Макромоделей, используемые в программах АСхП, — это упрощенные представления функциональных узлов ИС, которые с достаточной для конкретного применения точностью отражают статические и динамические характеристики на внешних выводах ИС.

Интерес к макромоделям связан с развитием технологии микроэлектроники, что позволило изготавливать ИС, содержащие $10^2 \dots 10^5$ вентилях на кристалле. Для столь больших многокомпонентных схем методы проектирования и программы для ЭВМ, основанные на моделях *R*-, *L*-, *C*-элементов, транзисторов, диодов, не обеспечивают проведение схмотехнического анализа и, тем более, оптимизации параметров. Размер задач схмотехнического проек-

тирования становится в этом случае столь велик, что превышает возможности современных САПР по объему требуемой памяти и необходимым затратам времени на моделирование.

В связи с этим возникает необходимость снижения размера задач схемотехнического проектирования. Одним из наиболее рациональных подходов является составление моделей ИС не из моделей традиционных компонентов электронных цепей, а из моделей подсистем упрощенной структуры — макромоделей. Макромодель подсистемы описывается системой уравнений значительно меньшего размера или существенно более простым алгоритмом расчета выходных переменных по сравнению с развернутым компонентным представлением той же подсистемы, что и обеспечивает уменьшение размера задачи моделирования в целом.

Снижение вычислительных затрат при использовании макромоделей ИС и фрагментов БИС достигается ценой некоторого уменьшения точности моделирования. Однако время моделирования уменьшается при этом в несколько раз или даже в несколько десятков раз, тогда как точность — всего лишь на 5...10%, что в большинстве практических применений вполне допустимо.

11.1.2. Принципы построения макромоделей. Как отмечалось в гл. 1, применение схемотехнических макромоделей можно рассматривать как компромисс между требованиями повышения точности функционально-логического уровня проектирования и снижения вычислительных затрат на схемотехническом уровне проектирования (рис. 11.1).

Макромодели как цифровых, так и аналоговых элементов разделяются на два типа: физические макромоделели в виде упрощенных электрических схем замещения, для которых расчет проводится на основе законов Кирхгофа, и информационные макромоделели, уравнения которых имеют вид функционального описания $y=f(x)$, где y и x — векторы выходных и входных переменных соответственно. Это деление отражает основные способы создания макромоделей. Сущность указанных типов макромоделей рассмотрена в п. 4.2.4, 4.5.2 и 8.4.2.

Наиболее целесообразным оказывается представление в виде макромоделей подсистем, топология и свойства которых многократно повторяются. Из таких макромоделей организуются библиотеки в САПР, для них решаются задачи подбора параметров и унификации сопряжения с обычными схемотехническими описаниями.

11.1.3. Иерархия и типовая структура макромоделей. На практике используются различные уровни декомпозиции функциональных узлов на подсистемы, для которых применяются макромоделели. Макромодели разрабатываются как для простых узлов с небольшим числом элементов, например для логического вентиля или дифференцирующего каскада (макромодель I на рис. 11.1), так и для более сложных узлов, кончая подсистемами типа регистров, сумматоров или трактов преобразования аналоговой информации (макромодель II на рис. 11.1).

Таким образом, существует иерархическое деление макромоделей по уровню сложности описываемых узлов. Создание и использование макромоделей элементарных подсхем оказывается достаточно простым, а сами макромоделли являются универсальными. Однако выигрыш во времени вычислений и затратах памяти при

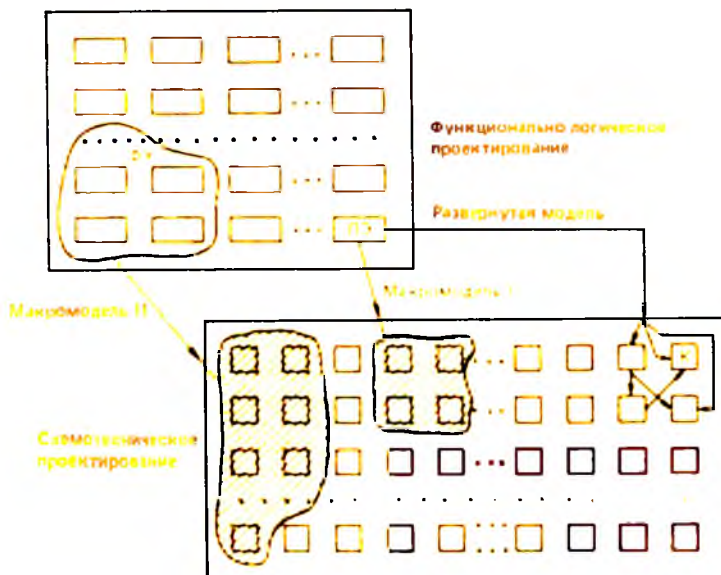


Рис. 11.1. К формированию макромоделей для программ АСхП:

К — компонент; ЛЭ — логический элемент; ФУ — функциональный узел

моделировании не так велик (в 2...20 раз). При замене макро-моделью сложных многокомпонентных узлов выигрыш возрастает (затраты сокращаются в 20...100 раз), но, как правило, необходима большая работа по выбору и определению значений параметров макромоделли.

Несмотря на то что большинство разработанных макромоделей связано с конкретными устройствами разного уровня сложности, можно выделить типовую обобщенную структуру макромоделли схемотехнического проектирования (рис. 11.2,а), которая характерна как для физических (рис. 11.2,б), так и для информационных (рис. 11.2,в) макромоделей цифровых и аналоговых под-схем.

В качестве простого примера, иллюстрирующего принципы и методы построения макромоделей, применимые как для цифро-вых, так и для аналоговых узлов, здесь и далее в § 11.1—11.3 рассматриваются различные представления логического элемен-та — двухвходового вентиля И—НЕ (рис. 11.3).

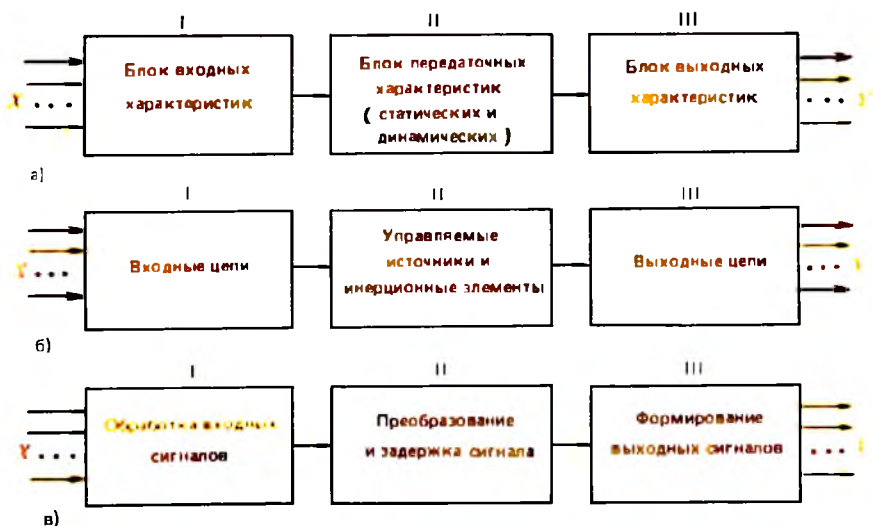


Рис. 11.2. Типовая структура макромоделей для программ АСхП:
 а — общий вид; б — для физической макромодели; в — для информационной макромодели

Пример 11.1. Макромодель ТТЛ-вентиля И — НЕ, структура которой соответствует типовой, показана на рис. 11.4. По форме представления — это физическая макромодель, непосредственно совместимая с программами АСхП, предусматривающими составление уравнений по методу узловых потенциалов. Для ее реализации необходимо задать функции, описывающие управляемые источники тока I_A , I_B и управляемые источники напряжения E_1 , E_2 , а также параметры пассивных компонентов R_1 , R_2 , C_1 и постоянного источника тока I_0 .

В данной макромодели предполагаются известными уровни напряжения логического нуля U_0 ($U_0 = 0,3$ В) и логической единицы U_1 ($U_1 = 3,1$ В). В этих пределах изменяется выходное напряжение макромодели. Возможные значения входных напряжений, на которые реагирует данная макромодель, опре-

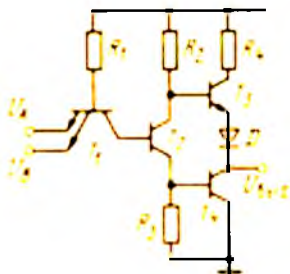


Рис. 11.3. Принципиальная схема двухвходового ТТЛ-вентиля И — НЕ

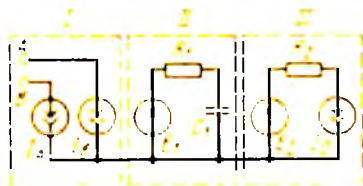


Рис. 11.4. Физическая макромодель ТТЛ-вентиля И — НЕ:
 I II III — типовые блоки

деляются минимально допустимым напряжением логической единицы $U_{1\text{мин}}$ ($U_{1\text{мин}}=1,9$ В) и максимально допустимым напряжением логического нуля $U_{0\text{макс}}$ ($U_{0\text{макс}}=0,8$ В). Указанные значения соответствуют диапазонам, в которых описаны функции $I_A(U_A)$ и $I_B(U_B)$. При выходе из заданных диапазонов значений входных напряжений состояние источников входных сигналов макромодели и, следовательно, состояния всех ее элементов не изменяются.

Источники тока $I_A(U_A)$ и $I_B(U_B)$, образующие блок I типовой структуры макромодели (рис. 11.4), могут быть либо представлены диодами, либо описаны в виде таблиц соответствия «напряжение — ток».

Блок II типовой структуры в данной макромодели представлен управляемым источником напряжения E_1 и инерционной цепью R_1C_1 (см. рис. 11.4).

Значение напряжения управляемого источника E_1 является функцией входных напряжений U_A , U_B и определяется в соответствии с таблицей истинности вентиля. Если оба входных сигнала превышают порог $U_{1\text{мин}}$, то $E_1 = -U_0$. Если один из входных сигналов меньше $U_{0\text{макс}}$, то $E_1 = U_1$. В остальных случаях значение напряжения E_1 равно разности уровня напряжения логической единицы и минимального из входных напряжений. Постоянная времени цепи R_1C_1 определяет задержку сигнала. Параметры цепи R_1 и C_1 выбираются из условия соответствия экспериментальной задержке.

Блок III типовой структуры макромодели составляют управляемый источник напряжения E_2 , а также выходное сопротивление R_2 и источник опорного тока I_0 . В рассматриваемой макромодели выходное напряжение E_2 равно напряжению на емкости C_1 , причем $C_1=550$ пФ при $E_1 \geq E_2$ и $C_1=300$ пФ при обратном соотношении. Выходное сопротивление определяет резистор $R_2 = 30$ Ом. Генератор тока I_0 введен для частичного учета изменения режимов выходной цепи вентиля.

Следует отметить, что введение в макромодель RC -цепей с изменяющимися параметрами позволяет без изменения структурной схемы макромодели приближенно воспроизводить форму выходного сигнала.

Очевидно, что при использовании в программах АСХП рассмотренной макромодели не требуется специальных приемов по формированию уравнений модели схемы в целом, но необходима определенная работа по формализации приведенного выше описания ее функционирования, программированию этого описания и внесение в библиотеку как самой макромодели, так и ее параметров.

Типовые блоки I и III макромодели осуществляют требуемое сопряжение внешних переменных программы и внутренних переменных макромодели. Кроме того, они могут отражать инерционные свойства входных и выходных цепей, а также влияние объединения по входу и разветвления по выходу эквивалентными схемами или уравнениями разного уровня точности и соответственно сложности. Основной преобразовательный блок II макромоделей обычно связан с блоками I и III только по управлению (управляется переменными из блока I и выдает управляющие переменные для блока III) и не связан явно с внешними цепями. В блоке II в ряде макромоделей учитывается суммарная инерционность внутренних цепей функционального узла.

Очевидно, что уровень сложности представления каждого из блоков и соответственно точности определяемых каждым блоком характеристик, получаемых при использовании макромоделей цифровых или аналоговых подсистем, могут быть различными. В связи с этим имеется иерархическое подразделение макромоделей по уровню сложности и точности. Как правило, это связанные понятия: увеличение точности достигается повышением сложности описаний управляемых источников, увеличением числа компонентов, введением нелинейностей активных и пассивных компонент макромоделей. Однако в ряде случаев оригинальное построение макромоделей позволяет на очень простой структуре получить требуемую практически точность.

11.2. МЕТОДЫ ПОСТРОЕНИЯ МАКРОМОДЕЛЕЙ

11.2.1. Основные процедуры. В настоящее время считается целесообразным создание макромоделей для всех серийно выпускаемых микросхем и для типовых функциональных узлов БИС и СБИС. Общая методика формирования макромоделей, предназначенных для АСХП, включает следующие процедуры [9].

1. Изучение свойств моделируемого объекта с целью получения информации о тех свойствах, которые должны быть отражены в его макромоделе.

2. Синтез структуры физической или информационной макромоделей, который осуществляется путем либо разработки оригинальной структуры, либо выбора готовой структуры из числа рекомендуемых для данного типа функционального узла.

3. Определение числовых значений параметров макромоделей, исходя из условий минимизации расхождений между характеристиками объекта или точной модели и аналогичными характеристиками, рассчитанными с использованием макромоделей.

4. Оценка точности полученной макромоделей.

5. Представление макромоделей в форме, соответствующей требованиям программы моделирования, для которой предназначена макромоделей.

При синтезе структуры электрической макромоделей в основном используют два принципиальных подхода: упрощение известной развернутой эквивалентной схемы или достижение функционального подобия внешних характеристик развернутой эквивалентной схемы и макромоделей.

Упрощение развернутой модели подсистемы осуществляется заменой группы элементов компонентной модели одним или несколькими элементами, в том числе управляемыми источниками напряжения или тока.

Функциональное подобие внешних характеристик достигается либо методами формальной аппроксимации заданных характеристик, либо неформальным синтезом схемы простой структуры, либо разработкой алгоритмической процедуры, обеспечивающей имитацию заданных функций.

11.2.2. Методы упрощения. Методы упрощения развернутого компонентного представления цифровой или аналоговой подсистемы, для которой строится макромодель, подразделяется на неформальные и формальные. В первом случае на основании неформального анализа производится замена группы элементов подсистемы одним или несколькими нелинейными управляемыми элементами макромодели. Во втором случае с использованием формальных процедур и машинного анализа выделяются наиболее существенные элементы схемы, которые и составляют макромодель, как правило, для некоторого конкретного режима ее работы.

Очевидно, что неформальное упрощение развернутой компонентной модели представляет собой достаточно сложную задачу, решение которой основано на глубоком знании схемотехники.

Пример 11.2. Рассмотрим макромодель ТТЛ-вентилей И — НЕ (см. рис. 11.3), в которой используется нелинейное сопротивление, заменяющее все элементы между входной и выходной цепями. Характеристика этого нелинейного сопротивления R_n , показанная на рис. 11.5,а, косвенно учитывает зависимость тока базы выходного транзистора ТТЛ-вентилей T_4 от входного напряжения. Эта характеристика может быть сформирована многократным расчетом на ЭВМ развернутой модели вентилей. Полученная в результате упрощения эквивалентная схема представлена на рис. 11.5,б. Эта схема вместе с нелинейной характеристикой сопротивления является макромоделью ТТЛ-вентилей.

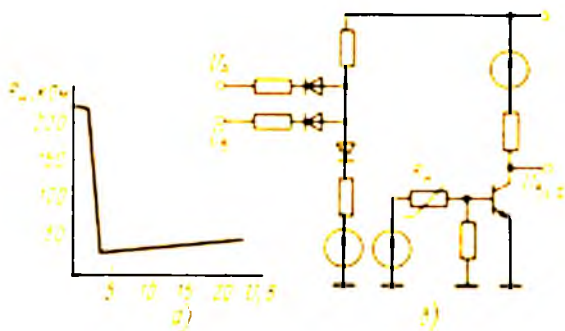


Рис. 11.5. Макромодель ТТЛ-вентилей И — НЕ с нелинейным сопротивлением:

а — зависимость нелинейного сопротивления от входного напряжения; б — эквивалентная схема

Одним из формальных подходов к упрощению схемы является исследование реакции на заданный сигнал при поочередной замене каждого компонента развернутой схемы короткозамкнутой или разомкнутой ветвью. В том случае, когда изменение выходного сигнала после проведенного элементарного преобразования структуры исходной схемы мало (в заданных пределах), соответствующий компонент исключается. Эта процедура осуществляется на ЭВМ применительно ко всем элементам схемы. Оставшаяся часть схемы используется как ее макромодель.

Другой способ формального получения структуры макромоде-
ли основан на выделении в исходной принципиальной схеме доми-
нирующих элементов. Для этого на ЭВМ проводится анализ раз-
вернутой схемы и рассчитывается чувствительность $S(p) =$
 $= p \cdot \partial F / \partial p$ ее выходных характеристик F к параметрам компонен-
тов p . Считается, что параметр p_1 является доминирующим по
отношению к параметру p_2 , если $S(p_1) \geq 10S(p_2)$. Элементы, име-
ющие большие на порядок чувствительности, включаются в мак-
ромодель.

Пример 11.3. Макромодель ТТЛ-вентиля И—НЕ (см. рис. 11.3), полу-
ченная методом анализа чувствительности параметров компонентов, состоит из
четырёх эквивалентных схем (рис. 11.6). Каждая схема соответствует опреде-
ленному участку передаточной характеристики вентиля. Чувствительность рас-
считывалась отдельно для каждого из режимов 1, 2, 3, 4. По результатам расче-
та на основании приведенного выше условия доминирования выделены эле-
менты макромодели для соответствующего участка характеристики.

Применение этой макромодели сократило время анализа примерно в 20
раз при погрешности моделирования передаточной характеристики в пре-
делах 0...2%.

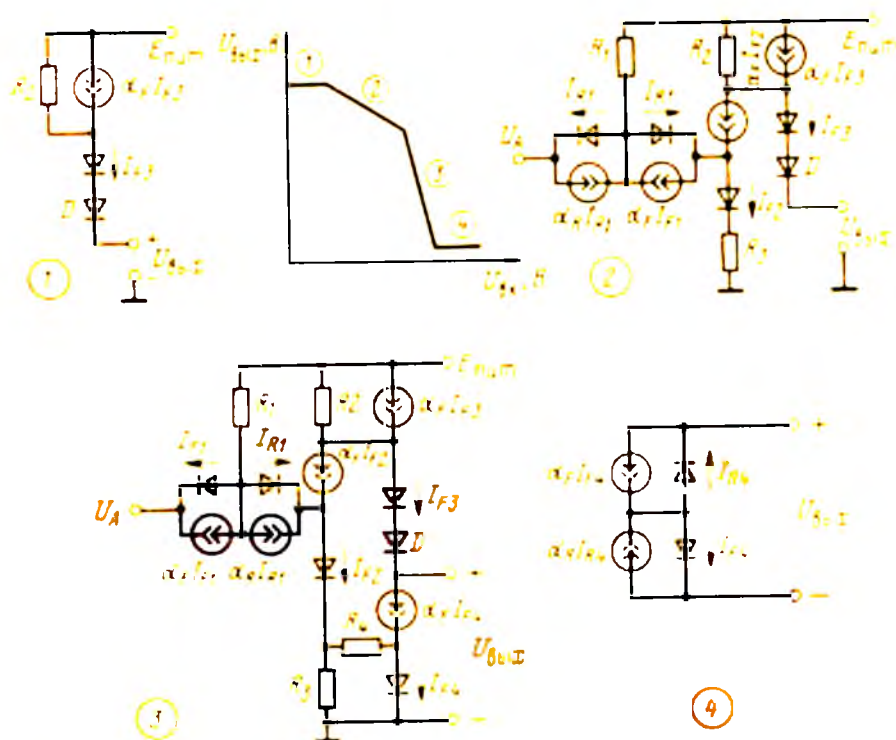


Рис. 11.6. Четыре эквивалентные схемы макромодели ТТЛ-вентиля И—НЕ, соответствующие разным областям его передаточной характеристики

Достоинство рассмотренных формальных методов упрощения развернутых схем для синтеза макромоделей состоит в ясном физическом смысле элементов макромоделей, а также в автоматизации процесса получения структуры макромоделей многокомпонентных функциональных узлов при сохранении заданной точности. Однако построенные в результате рассмотренных формальных процедур макромоделей все же достаточно сложны, а потому не всегда обеспечивают требуемое повышение эффективности расчетов.

11.2.3. Методы функционального подобия. Этими методами могут быть построены как физические, так и информационные макромоделей аналоговых или цифровых подсхем. Однако структура физических макромоделей, состоящих из традиционных схемных элементов, может существенно отличаться от структуры исходной модели, ибо подобие требуется только для внешних рабочих характеристик. В этом случае классификация на физические и информационные модели является условной и отражает лишь форму представления макромоделей.

Макромоделей, построенные методом подобия, как правило, мало связаны с технологическими особенностями изготовления микросхем. Более того, микросхемы одного функционального назначения, но изготовленные по разной технологии (например, МДП- и биполярной) могут имитироваться одинаковыми макромоделями, полученными аппроксимацией характеристик на внешних выводах. Такие макромоделей не подвержены моральному старению, связанному с быстрым прогрессом технологии.

Пример 11.4. Построенная по принципу функционального подобия макромоделей ТТЛ-вентиля, большая часть параметров которой определяется из выходных характеристик вентиля, показана на рис. 11.7 [47].

Входная цепь моделируется резистором $R_{вх} = (E - U_{вых0}) / I_{вх0}$, где E — напряжение источника питания, $U_{вых0}$, $I_{вх0}$ — номинальные для уровня логического 0 выходное напряжение и входной ток.

Сопротивление R эквивалентно сопротивлению R_2 в исходной схеме (см. рис. 11.3). Генератор тока $I = GU$ моделирует зависимость коллекторного тока транзистора T_2 от управляющего напряжения U . Эта зависимость аппроксимируется кусочно-линейной функцией. Емкость C является эквивалентной емкостью транзистора T_2 , определяющей инерционность вентиля. Значение C выбирается так, чтобы было обеспечено соответствие $U_{вых}$ фронту выходного импульса схемы; форма фронта описывается следующим выражением:

$$U_{вых}(t) = (U_1 - U_0) \{1 - \exp[-t/(RC)]\} + U_0.$$

Выходное сопротивление вычисляется по формуле

$$R_0 = (U_{макс0} - U_{мин0}) / \{N[(E - U_0) / R_{вх}]\},$$

где $R_{вх}$ — входное сопротивление, определенное ранее; $U_{макс0}$, $U_{мин0}$ — максимальное и минимальное выходное напряжение низкого уровня; N — число нагрузок вентиля.

Выходной генератор тока $I_0 = G_0 U_K$ задается кусочно-линейной аппроксимацией зависимости выходного напряжения, деленного на R_0 , от напряжения на коллекторе транзистора T_2 исходной схемы.

При определении управляющего напряжения $U = \min\{U_A, U_B\}$ каждое входное напряжение задерживается на соответствующее время t^+_{α} или t^-_{α} посредством специального оператора задержки в макромодели, действие которого иллюстрируется рис. 11.7,з. График I представляет входной импульс, а графики II и III — тот же импульс, но задержанный на время t_{II} и t_{III} соответственно ($t_{III} > t_{II}$). Если $t^+_{\alpha} < t^-_{\alpha}$, то управляющим импульсом является огибающая максимумов задержанных сигналов (график IV). Для соотношения задержек переключения $t^+_{\alpha} > t^-_{\alpha}$ управляющим сигналом является огибающая минимумов задержанных сигналов (график V).

Значения параметров данной макромодели ТТЛ-вентиля, например, следующие: $R_{вх} = 4$ кОм, $R = 1,6$ кОм, $R_0 = 57$ Ом, $C = 2,55$ пФ, $t^+_{\alpha} = 2$ нс, $t^-_{\alpha} = 2,5$ нс.

Рассмотренная макромодель обладает достаточной гибкостью. Она легко может быть преобразована в макромодель элемента ИЛИ—НЕ при замене функции управляющего напряжения на $U = \max\{U_A, U_B\}$ или в макромодель триггера введением в управляемый источник другой управляющей функции, которая является кусочно-линейной аппроксимацией вольт-амперной характеристики бистабильного элемента.

Использование описанной в примере 11.4 макромодели для анализа цифровой схемы, полная модель которой имеет 257 узлов, обеспечило сокращение количества узлов до 40. Временные затраты на моделирование уменьшились в 15 раз по сравнению с моделированием на компонентном уровне.

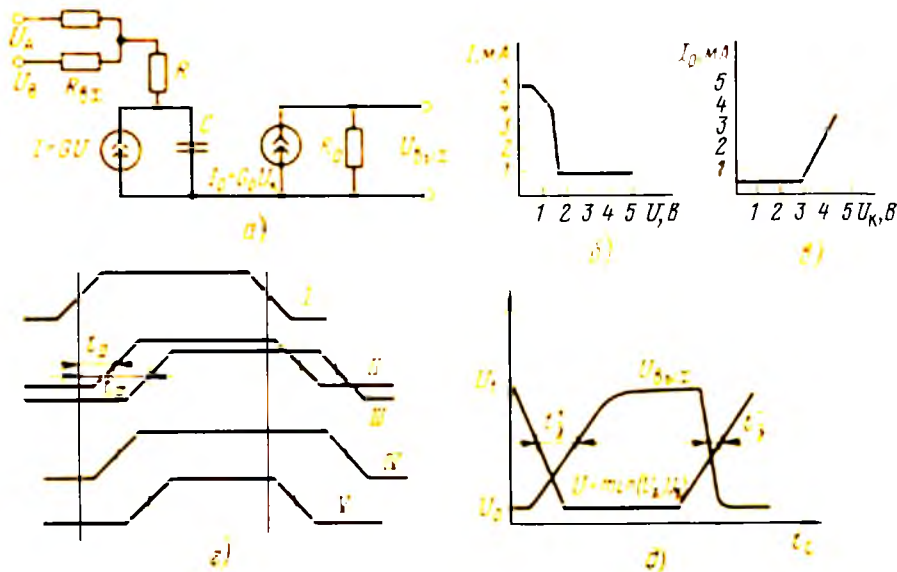


Рис. 11.7. Функциональная макромодель ТТЛ-вентиля И — НЕ:

а — схема макромодели; б, в — вольт-амперные характеристики управляемых источников тока макромодели; г — представление оператора несимметричной задержки; д — входной и выходной импульсы схемы

Следует отметить, что подход к построению физических макромоделей в виде эквивалентных схем методом подобия внешних характеристик, особенно часто применяемый при построении макромоделей сложных аналоговых узлов, под силу только специалисту-схемотехнику. Сложность задач состоит в неформальном выборе характеристик, которые следует отразить в макромоделе управляемыми источниками тока или напряжения, а также в необходимости получения простых аппроксимирующих выражений для определения параметров макромоделей.

11.2.4. Аналитические преобразования. Этот подход к построению макромоделей основан на аналитических преобразованиях уравнений развернутой полной модели, которые направлены на то, чтобы все напряжения на внутренних элементах и все аргументы нелинейных зависимостей оказались выраженными через внешние входные и выходные переменные. Число уравнений макромоделей в этом случае становится равным числу внешних переменных. Полученные при свертывании выражения включаются в описание управляемых источников макромоделей, которые в результате как бы содержат внутри себя исключенные элементы развернутой модели.

Особенностью и главным достоинством такого подхода является полное сохранение точности моделирования при сокращении в несколько раз числа элементов в макромоделе.

Рассмотренный способ получения макромоделей, называемый также редукцией уравнений развернутой модели, применялся для получения макромоделей вентилях на МДП-структурах и для упрощения макромоделей операционных усилителей. При этом использовались неформальные процедуры вывода требуемых аналитических выражений. Автоматизация процесса редукции представляется затруднительной, однако ее осуществление могло бы стать одной из эффективных процедур создания наиболее точных макромоделей для САПР.

Пример 11.5. Рассмотрим макромоделей вентилях на МДП-структурах, полученные методом редукции развернутой модели. Эта модель вентиля ИЛИ-НЕ показана на рис. 11.8,а, где для представления МДП-транзистора используется источник тока и четыре емкости. В соответствии с приведенной схемой справедливы следующие соотношения:

$$\begin{aligned} I_1 &= C_{11} \frac{dU_1}{dt} + C_{21} \frac{d}{dt} (U_1 + U_{\text{вых}}), \\ I_2 &= C_{12} \frac{dU_2}{dt} + C_{22} \frac{d}{dt} (U_2 - U_{\text{вых}}), \\ I_{\text{вых}} &= C_{41} \frac{dU_{\text{вых}}}{dt} + J_1 - C_{21} \frac{d}{dt} (U_1 - U_{\text{вых}}) + C_{42} \frac{dU_{\text{вых}}}{dt} + J_2 - \\ &- C_{22} \frac{d}{dt} (U_2 - U_{\text{вых}}) + C_{33} \frac{dU_{\text{вых}}}{dt} - J_3 + C_{23} \frac{d}{dt} (U_{\text{вых}} - E) + C_n \frac{dU_{\text{вых}}}{dt}, \end{aligned}$$

которые получены на основании того, что управляющие напряжения на затворе, стоке и подложке каждого МДП-транзистора можно выразить через

внешние напряжения $U_1, U_2, U_{\text{ВЫХ}}, E$ ($U_{3и1}=U_1, U_{3и2}=U_2, U_{3и3}=0, U_{сн1}=U_{\text{ВЫХ}}, U_{сн2}=U_{\text{ВЫХ}}, U_{сн3}=E-U_{\text{ВЫХ}}, U_{пн1}=0, U_{пн2}=0, U_{пн3}=U_{\text{ВЫХ}}$).

Полученные выше выражения, описывающие управляемые источники токов, преобразуются к следующему виду:

$$I_1 = (C_{11} + C_{21}) \frac{dU_1}{dt} - C_{21} \frac{dU_{\text{ВЫХ}}}{dt},$$

$$I_2 = (C_{12} + C_{22}) \frac{dU_2}{dt} - C_{22} \frac{dU_{\text{ВЫХ}}}{dt},$$

$$I_{\text{ВЫХ}} = (C_{41} + C_{42} + C_{33} + C_{21} + C_{22} + C_{23} + C_{\Pi}) \frac{dU_{\text{ВЫХ}}}{dt} - C_{21} \frac{dU_1}{dt} - C_{22} \frac{dU_2}{dt} + J_1 + J_2 - J_3.$$

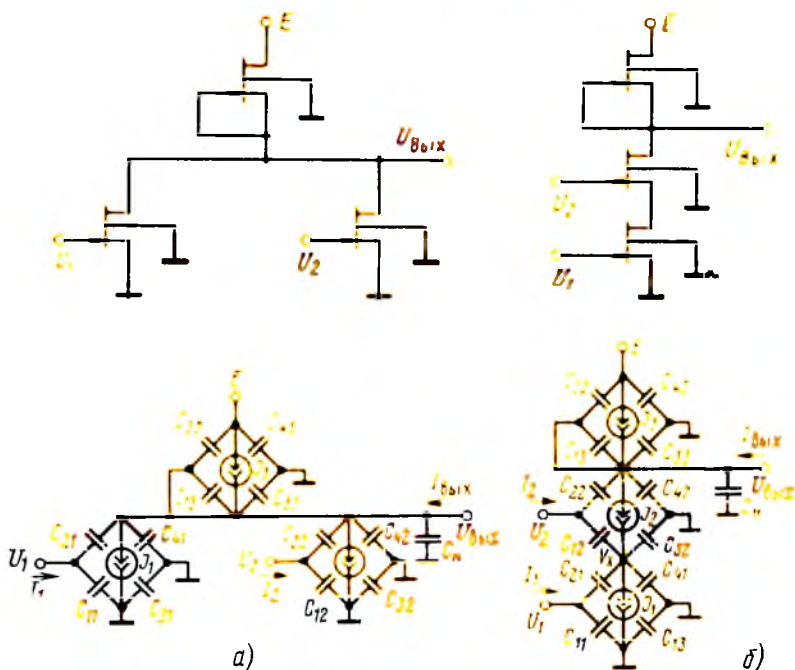


Рис. 11.8. Макромодели вентилях на МДП-транзисторах:
а — вентиль ИЛИ — НЕ; б — вентиль И — НЕ

Таким образом, макромодель вентиля ИЛИ — НЕ можно ввести в программу АСХП как три источника тока $I_1, I_2, I_{\text{ВЫХ}}$, описанных приведенными дифференциальными уравнениями. В данном случае емкости как бы оказались «погруженными» в управляемые источники тока. Число дифференцируемых переменных уменьшилось в четыре раза: от 12 напряжений на емкостях в компонентном описании до 3 напряжений (входные и выходные) в макромодели.

Таким же способом для МДП-вентиля И — НЕ (рис. 11.8, б) получена макромодель из трех источников тока $I_1, I_2, I_{\text{ВЫХ}}$, управляемых тремя внеш-

ними напряжениями U_1 , U_2 , $U_{\text{вых}}$ и одним внутренним узловым потенциалом U_x , который описывается отдельным дифференциальным уравнением согласно первому закону Кирхгофа:

$$\begin{aligned} I_1 &= (C_{11} + C_{21}) \frac{dU_1}{dt} - C_{21} \frac{dU_x}{dt}, \\ I_2 &= (C_{12} + C_{22}) \frac{dU_2}{dt} - C_{22} \frac{dU_x}{dt}, \\ I_{\text{вых}} &= (C_{33} + C_{23} + C_{22} + C_{42} + C_N) \frac{dU_{\text{вых}}}{dt} - C_{22} \frac{dU_x}{dt} + J_2 - J_3, \\ (C_{32} + C_{12} + C_{21} + C_{41}) \frac{dU_x}{dt} - C_{12} \frac{dU_2}{dt} - C_{21} \frac{dU_1}{dt} + J_1 - J_2 &= 0. \end{aligned}$$

Применение такой макромоделей позволяет сократить в три раза число дифференциальных переменных на каждый вентиль И—НЕ по сравнению с компонентной моделью при сохранении той же точности.

11.2.5. Методы факторизации. Для решения специальных задач схемотехнического моделирования могут быть построены макромоделей с учетом внешних факторов Q , к которым относятся эксплуатационные факторы (температура окружающей среды, влажность, воздействия радиации, изменения напряжения питания, параметры входных сигналов); структурные факторы (коэффициенты разветвления по выходу и объединения по входу для логических схем, виды нагрузки); конструктивные факторы (геометрические размеры, конфигурация элементов, параметры линий связи и т. п.).

Такие макромоделей строятся обычно как функциональные модели преобразования входного сигнала в выходной. Форма выходного сигнала (статического или динамического) описывается аппроксимирующей функцией, параметры которой в свою очередь являются функциями внешних факторов. Формирование таких макромоделей осуществляется с использованием либо статистических методов, либо методов формальной аппроксимации, в частности на выбранном разработчиком стандартном наборе функций для обеспечения реализации в конкретных программных разработках. Например, динамическая макромоделей электронной схемы может быть представлена следующей полиномиальной зависимостью

$$dU_{\text{вых}}/dt = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \dots$$

где $\{\alpha_i = f_i(Q, S), i=0, 1, 2, \dots\}$ — параметры макромоделей, зависящие от внешних факторов Q и параметров входного сигнала S . Параметры макромоделей α_i должны быть определены в каждой точке факторного пространства, для чего могут использоваться методы планирования эксперимента.

11.2.6. Библиотеки типовых элементов. Рассмотренные выше методы построения макромоделей приводят к заключению, что в настоящее время не существует автоматической процедуры получения макромоделей многокомпонентных подсхем, и их построение

является, по существу, искусством. В связи с необходимостью введения в САПР радиоэлектронной аппаратуры формальных методов создания макромоделей представляется перспективной разработка библиотеки типовых каскадов макромоделей в виде макроописаний или моделей макроэлементов.

Возможность создания таких типовых наборов обусловлена тем, что практически все макромодели как цифровых, так и аналоговых подсистем имеют общую структуру (см. рис. 11.2), а конкретные макромодели отличаются зачастую в реализации лишь одного или двух блоков.

Имея набор макроописаний типовых блоков макромоделей, можно «собирать» требуемую макромодель в соответствии с конкретным устройством и решаемой задачей схемотехнического проектирования, в частности получать адаптивные макромодели требуемой точности.

Наряду с наличием в САПР таких библиотек типовых решений для отдельных блоков макромоделей, необходимо обеспечение диалогового режима формирования и оценки макромоделей на основе формальных процедур, а также обеспечение методами и процедурами определения и оптимизации их параметров.

Некоторые вопросы создания подобных типовых решений рассмотрены в § 11.4 и 11.5.

11.3. ФОРМЫ ПРЕДСТАВЛЕНИЯ МАКРОМОДЕЛЕЙ

Каждая макромодель в программах АСхП рассматривается как неделимое целое и ей соответствуют два представления — внешнее и внутреннее.

Внешнее представление макромодели — это ее формальное описание на входном языке моделирующей программы. Оно обычно включает имя макромодели, по которому осуществляется обращение к соответствующей подпрограмме расчета макромодели, список узлов, к которым в заданном порядке подключаются внешние выводы, а также перечень параметров макромодели или указатель для поиска параметров в базе данных.

Внутреннее представление макромодели — это набор подпрограмм, специальных библиотечных функций и таблиц, взаимодействия которых определяется главной по отношению к ним программой. Внутреннее представление зависит от организации моделирующей программы. Следует отметить, что наличие в программе макромоделей усложняет передачу и обработку данных, поэтому вопрос рационального внутреннего представления макромоделей является существенным. При неудачной организации (излишнем количестве подпрограмм, неэффективном поиске, частом обмене с внешней памятью) может значительно уменьшиться предполагаемый выигрыш от применения макромоделей.

Практически во всех случаях при использовании макромоделей создается соответствующая библиотека. Такая библиотека может содержать иерархический ряд макромоделей одного типа

функциональных узлов. Макромодели различаются уровнями точности и сложности (требуемых ресурсов памяти, затрат машинного времени).

Макромодели конкретных типов микросхем или функциональных узлов БИС, как правило, содержат параметры в виде числовых констант и представляются таблицами. В этом случае максимально используются преимущества табличного представления. Для внутреннего представления макромоделей, имеющих универсальный характер, применяются функции и подпрограммы, что обычно несколько увеличивает, по сравнению с табличным, временные затраты при меньших требованиях к памяти. В связи с наличием в библиотеках программ проектирования макромоделей различного типа в настоящее время встает задача разработки стратегии адаптации макромоделей, которая позволила бы выбирать оптимальную по точности и сложности макромодель по формальным признакам в автоматическом режиме.

Следует подчеркнуть, что при любой начальной форме макромодели — в виде функций, связывающих выходные переменные с входными, или в виде эквивалентной схемы, внутреннее представление всегда связано с подпрограммами или таблицами, описывающими функции нелинейных или управляемых элементов макромодели.

Во многих программах анализа отдается предпочтение табличному представлению макромоделей. Это значит, что входные и выходные характеристики исходной схемы или некоторых нелинейных элементов макромодели описываются с помощью параметрических таблиц соответствия. Параметрами в таких таблицах являются различные нагрузки и различные входные сигналы. Значения, заносимые в таблицы, получаются в результате либо анализа развернутой схемы, либо измерений.

Преимущество табличных макромоделей состоит в том, что при их использовании в сочетании с эффективными алгоритмами поиска примерно в 4...5 раз уменьшается время решения нелинейных уравнений схем. Однако для получения достаточной точности табличных макромоделей необходимы большие таблицы, которые требуют большого количества предварительных вычислений и значительной емкости памяти ЭВМ. Очевидно, что табличные макромодели становятся эффективными при условии многократного использования — следовательно, это должны быть макромодели типовых стандартных элементов.

Пример 11.6. Рассмотрим способы внутреннего представления макромодели вентиля И—НЕ, эквивалентная схема которой показана на рис. 11.9 и состоит из трех каскадов, соответствующих входным, передаточным и выходным характеристикам этого логического элемента [47].

Управляемые источники тока $I_1(U_1)$ и $I_2(U_2)$ моделируют входные вольт-амперные характеристики вентиля. Характеристики эти либо снимаются экспериментально, либо рассчитываются моделированием на компонентном уровне. Для задания токов $I_1(U_1)$ и $I_2(U_2)$ используется табличное представление

входных характеристик. С целью повышения точности необходимо при описании одного входа учитывать состояние другого входа (нулевое или единичное).

Источник напряжения $E_1(U)$ моделирует передаточную характеристику вентиля (см. рис. 11.6) $U_{\text{вых}} = f(U_{\text{вх}})$. Управляющее напряжение $U = \min(U_1, U_2)$. Передаточная характеристика аппроксимируется кусочно-линейной функцией с использованием табличного описания. Значение выходного напряжения $U_{\text{вых}} = f(U)$ присваивается источнику напряжения $E_1(U)$.

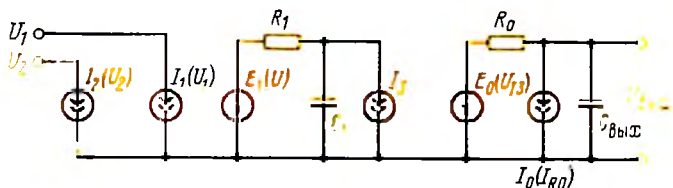


Рис. 11.9. Трехкаскадная макромодель ТТЛ-вентиля И — НЕ

Элементы эквивалентной схемы R_1 и C_1 образуют фильтр нижних частот, который моделирует переходную характеристику вентиля, т. е. фронты выходного импульса напряжения, задаваемого источником E_1 . Значение R_1 постоянно и задается константой. Значение C_1 различно при моделировании длительности фронта и среза выходного импульса, и оно задается двумя константами и условием проверки режима. Выбор C_1 осуществляется с помощью подпрограммы-функции. Значение I_3 не изменяется при анализе по постоянному току и задается константой.

Источник напряжения $E_0(U_{I3})$ в выходном каскаде моделирует среднее время задержки распространения сигнала от входа к выходу t_3 . Для этого значение напряжения $U_{I3}(t_k)$, соответствующее k -му шагу интегрирования или моменту времени t_k , присваивается источнику напряжения E_0 в момент времени $(t_k + t_3)$. Данная операция задержки реализуется подпрограммой-функцией.

Моделирование выходных характеристик осуществляется источником тока $I_0(I_{R0})$, управляемым током через сопротивление R_0 . Выходное сопротивление R_0 определяется из кусочно-линейной аппроксимации семейства выходных характеристик вентиля. Для токов и R_0 используется табличное представление. Емкость $C_{\text{вых}}$ — это выходная емкость вентиля, которая задается константой.

11.4. МАКРОМОДЕЛИ ЦИФРОВЫХ СХЕМ

Макромодели логического элемента И—НЕ рассмотрены в примерах предыдущих разделов. Аналогичным образом формируются макромодели сложных цифровых схем [35]. Все эти макромодели содержат электрологические и логико-электрические преобразования для сопряжения сигналов макромоделей с электрическими сигналами, используемыми программами схемотехнического проектирования.

Пример 11.7. Рассмотрим макро модель схемы И — ИЛИ — НЕ, состоящей из четырех включенных параллельно вентилей И и последовательного вентиля ИЛИ — НЕ (рис. 11.10,а). Логика действия схемы описывается следующим образом:

1) на выходе схемы устанавливается логическая 1, если на выходах всех вентилей И устанавливается логический 0, т. е. имеется логический 0 хотя бы на одном из входов каждого вентиля И;

2) на выходе схемы устанавливается логический 0, если хотя бы на одном из выходов вентилей И установлена логическая 1, т. е. на всех входах хотя бы одного вентиля И имеются логические 1.

Эквивалентная схема макро модели показана на рис. 11.10,б. Входные источники тока могут быть идеальными или иметь диодную характеристику, заданную, как правило, в табличной форме. Они учитывают влияние входных цепей на форму сигнала. Значение напряжения источника E_1 определяется в подпрограмме, составленной с использованием значений уровней логических 0 и 1 и либо условия 1), либо условия 2), которые рассмотрены выше. Цепь R_1C_1 описывает регулируемую задержку нарастания или спада напряжения. Напряжение источника E_2 равно напряжению на емкости C_1 . Резистор R_2 моделирует выходное сопротивление, которое в первом приближении считается постоянным и задается константой, а при более детальном моделировании представляется функцией напряжения и тока нагрузки.

Аналогичным образом составляются модели мультивибраторов, триггеров различных типов, регистров, счетчиков, дешифраторов и других цифровых устройств.

Как видно из рассмотренных примеров, входные и выходные цепи могут быть одинаковыми практически для всех макромоделей цифровых схем, так как они соответствуют входным и выходным цепям логического вентиля. Большая или меньшая сложность представления этих цепей (идеальные или неидеальные источники входного тока, инерционные или безынерционные цепи, постоянные или управляемые параметры пассивных элементов) зависит от требований к точности макромоделей.

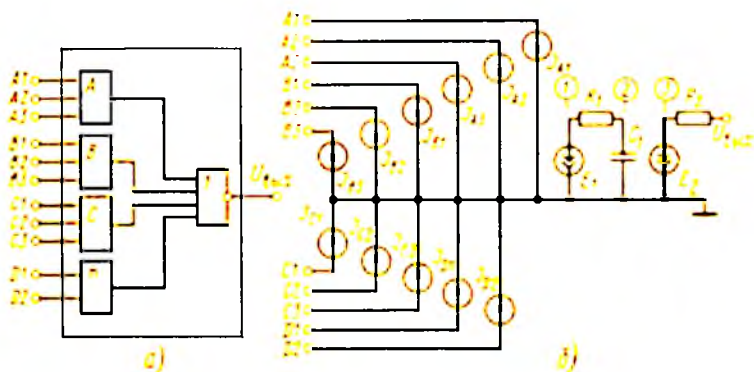


Рис. 11.10. Макро модель схемы И — ИЛИ — НЕ:

а — структура схемы; б — эквивалентная схема макро модели

Преобразовательная часть макромодели индивидуальна для каждой схемы. Эту часть можно записывать для каждой новой макромодели (новой схемы цифрового устройства) в виде подпрограммы на языке программирования и заносить в библиотеку макромоделей. Вмешательство в программную часть требует определенной квалификации пользователя и не всегда желательно, так как может повлечь за собой непредсказуемые последствия.

Поэтому для расширения возможностей макромоделирования следует использовать другой подход к формированию макромоделей — задание описания макромодели на уровне входной информации для программы проектирования в виде способа соединения стандартных макроэлементов. Такое информационное задание макромоделей обеспечивает неограниченные интересы пользователей без вмешательства в программную часть.

Набор типовых макроэлементов, включаемых в программы АСХП для формирования макромоделей цифровых устройств, должен содержать управляемые электрические двухполюсники и логические макроэлементы, подпрограммы которых содержатся в моделирующей системе и для которых разработано стандартное представление во входном языке.

Пример 11.8. Рассмотрим набор типовых макроэлементов, достаточный для формирования макромоделей любых заданных информационным описанием цифровых схем. В этот набор могут быть включены такие элементы, как нелинейные источники тока $I(U)$, сопротивления R , емкости C , времязависимые источники напряжения $E(t)$, которые имеются практически во всех программах АСХП, а потому могут считаться стандартными.

Специальными макроэлементами являются типовые нелинейные двухполюсники и логические макроэлементы в виде источников напряжения:

макроемкость $W(U)$, емкость которой

$$C(W) = \begin{cases} f_1(U) & \text{при } dU/dt \leq 0, \\ f_2(U) & \text{при } dU/dt > 0, \end{cases}$$

служит для моделирования задержки при нарастании или уменьшении напряжения входного сигнала и отражает задержку переключения логической схемы;

управляемые напряжениями источники $E_z = f(U_V)$, $E_{\min} = \min(U_{V1}, U_{V2}, \dots, U_{VM})$, $E_{\max} = \max(U_{V1}, U_{V2}, \dots, U_{VM})$, где $V, V1, V2, \dots, VM$ — напряжения на управляющих элементах;

элемент логики $L(V1, V2, \dots, VM)$, заданный в виде логического уравнения от аргументов $V1, V2, \dots, VM$, причем значение $V_i = 1$ соответствует выполнению отношения $\{U_{Vi}(\cdot) U_{Vi0}\}$, где $(\cdot)$ обозначает один из знаков отношения $>, \geq, <, \leq$, а U_{Vi0} — заданное напряжение; если указанное соотношение не выполняется, то $V_i = 0$; выходное значение соответствует уровням напряжения логических 1 или 0;

индикатор фронта сигнала $F(V)$, который равен 0 во всех случаях, кроме случая пересечения уровня U_{V0} , заданного на элементе напряжения ($F(V)$ принимает значение 1 тогда и только тогда, когда на текущем шаге интегрирования дифференциальных уравнений t^+ напряжение на рассматриваемом элементе больше (меньше) U_{V0} , а на предыдущем шаге интегрирова-

ния t^- — меньше (больше) U_{V0} (рис. 11.11)); индикатор фронта обозначается $F(V) = \uparrow U_{V0}$ или $F(V) = \downarrow U_{V0}$;

элемент памяти $P(V1, V2)$ задается в виде логической зависимости p^+ от значений $V1, V2, p^-$, где p^+ и p^- — значения соответствующих напряжений $V1$ и $V2$ на текущем и предыдущих шагах интегрирования;

элемент сообщения $X(V1, V2)$, заданный аналогично элементу логики, причем значению 1 ставится в соответствие операция выдачи сообщения о недопустимом сочетании сигналов $V1$ и $V2$.



Рис. 11.11. Графическое представление действия макроэлемента «индикатор фронта (а) и среза (б)»

Эквивалентная схема макромоделей ТТЛ-вентиля И — НЕ, сформированная из макроэлементов, представлена на рис. 11.12. Условное описание макромоделей отражает узлы подключения и логико-функциональные цепи (ЛФЦ):

входы: $A(JA, CA),$

$B(JB, CB);$

ЛФЦ: $\left. \begin{array}{l} \min(A, B), R1, W1; \\ Z1(W1), R2, W2; \end{array} \right\} \text{ERC-цепи};$

выход: $Z2(W2), R3, J3, C3.$

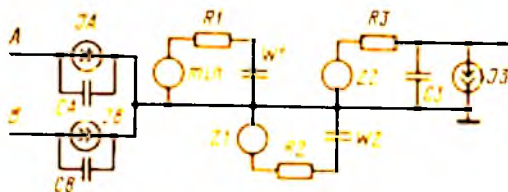


Рис. 11.12. Макромодель ТТЛ-вентиля И — НЕ, сформированная из макроэлементов

Эквивалентная схема макромоделей тактируемого DRS-триггера с асинхронными S- и R- входами, сформированная из макроэлементов, приведена на рис. 11.13.

Рассматриваемый триггер переключается на фронте тактового сигнала, подаваемого на вход C, когда значение сигнала превосходит заданное значение U_{C0} , причем переключение не зависит от длительности фронта. После переключения состояние триггера не меняется до прихода следующего тактового

сигнала. Переключение происходит согласно таблице истинности. Напряжение на входе D считается 0 при $U_D < U_{D0}$ и считается 1 при $U_D \geq U_{D0}$. Кроме того, в данном триггере имеются вход S установки 1 и вход R сброса в 0, воздействие сигналов на которых не зависит от информации на входах C и D .

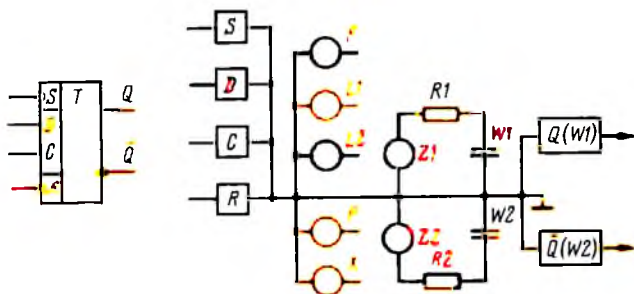


Рис. 11.13. Макромодель DRS-триггера, сформированная из макроэлементов

Макромодели DRS-триггера соответствует следующее условное описание: входы: S, D, C, R ;

ЛФЦ: логические макроэлементы:

$$F(C) = \uparrow U_{C0};$$

$$L1(F, D, S) = F \cdot D + S,$$

$$L2(F, D, R) = F \cdot D + R,$$

$$\rho^+(L1, L2) = \rho^- \cdot \overline{L1} \cdot \overline{L2} + L1 \cdot \overline{L2},$$

$$X(S, R) = S \cdot R;$$

ERC-цепи:

$$Z1(p) = \begin{cases} U_{\text{вых}}^1 & \text{при } p = 1, \\ U_{\text{вых}}^0 & \text{при } p = 0; \end{cases}$$

$$R1, W1;$$

$$Z2(p) = \begin{cases} U_{\text{вых}}^0 & \text{при } p = 1, \\ U_{\text{вых}}^1 & \text{при } p = 0; \end{cases}$$

$$R2, W2;$$

выходы: $Q(W1), \bar{Q}(W2)$

В приведенном описании логических макроэлементов знаки «+», «·», «—», обозначают логическое сложение, умножение и отрицание, $U_{\text{вых}}^0$ и $U_{\text{вых}}^1$ являются выходными уровнями триггера.

Проанализируем действие макромодели триггера, сформированной из макроэлементов. В момент, когда положительный фронт тактового импульса пересекает уровень U_{C0} , значение F становится равным 1. Пусть напряжение D_0 превосходит U_{D0} , тогда оно интерпретируется как 1. Значение $L1$ становится

равным 1. Происходит переключение p в 1, если перед этим $p^- = 0$, и не происходит переключения, если перед этим $p^- = 1$. Если же напряжение на входе $D = 0$, то согласно ЛФЦ происходит переключение триггера в 0, если $p^- = 1$, и переключение не происходит, если $p^- = 0$. Легко убедиться, что поступление сигналов S и R влечет (в зависимости от состояния триггера) соответствующее переключение. Элемент X при значении $S \cdot R = 1$ выдает сообщение о неправильном функционировании схемы (ситуация совпадения во времени поступления импульсов установки и сброса). Элементы $Z1$ и $Z2$ по характеру зависимости от p взаимно дополняют, что необходимо для моделирования взаимодополнительных выходов Q и $\bar{Q}$, причем макроемкости $W1$ и $W2$ моделируют длительность переключений $0 \rightarrow 1$ и $1 \rightarrow 0$.

11.5. МАКРОМОДЕЛИ АНАЛОГОВЫХ СХЕМ

Аналоговые интегральные схемы предназначены для преобразования и обработки сигналов, изменяющихся по закону непрерывной функции, в частном случае линейной. Это — разнообразные усилители, выпрямители, преобразователи, умножители, стабилизаторы и т. д. В РЭА наиболее широко распространены операционные усилители (рис. 11.14, а), которые имеют большой

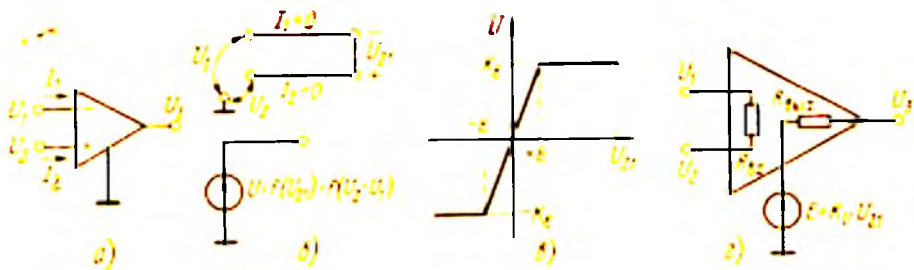


Рис. 11.14. Квазиидеальные макромоделли операционного усилителя: а — условное обозначение ОУ; б — макромоделль в виде управляемого источника напряжения $U = f(U_{21})$; в — характеристика $U = f(U_{21})$; K_c — усиление в разомкнутом контуре, K_d — напряжение, составляющее доли милливольт; г — статическая макромоделль, учитывающая входное, выходное сопротивления и K_U

коэффициент усиления, порядка $10^3 \dots 10^6$, и применяются в качестве активного элемента в схемах с обратными связями. Операционные усилители используются для выполнения математических операций: суммирования, вычитания, интегрирования и дифференцирования, а также при построении всевозможных аналоговых устройств [48].

Простейшая макромоделль операционного усилителя (рис. 11.14, б) соответствует идеальному операционному усилителю, характеристика которого показана на рис. 11.14, в. Указанная макромоделль имеет бесконечно большие коэффициент усиления по напряжению, полосу пропускания, входное сопротивление и скорость нарастания выходного напряжения, а также нулевые входной ток, напряжение смещения нуля и его температурный дрейф, время

нарастания выходного напряжения, выходное сопротивление и потребляемый ток. Для обеспечения таких свойств и стабилизации параметров операционные усилители реализуют по достаточно сложным схемам, содержащим несколько десятков компонентов, из которых, как правило, свыше 50% составляют транзисторы, причем относительное число транзисторов растет с увеличением числа компонентов схемы.

Для приближенной оценки качественных характеристик проектируемых устройств может использоваться рассмотренная выше идеальная макромодель операционного усилителя или его квази-идеальная макромодель (рис. 11.14,з). Последняя учитывает основные показатели «неидеальности» — коэффициент усиления по напряжению $K_U \neq \infty$, входное сопротивление $R_{вх} \neq \infty$, выходное сопротивление $R_{вых} \neq 0$.

При составлении макромоделей аналоговых схем применяют, в основном, эвристические приемы. Первоначально рассматривается идеальная макромодель, отражающая основную функцию, выполняемую аналоговым устройством, а затем производится постепенное повышение точности введением в состав идеальной макромодели дополнительных элементов, характеризующих отклонения или нестабильности выполнения этой функции.

Составление макромоделей, таким образом, осуществляется специалистом-схемотехником, и эти модели отражают знания, а в ряде случаев — искусство разработчика. Контролем корректности макромоделей служит степень совпадения характеристик с данными эксперимента или с результатами машинного расчета развернутых принципиальных схем, в которых используются модели нелинейных компонентов большей универсальности, сложности и точности.

Пример 11.9. На рис. 11.15,а приведена схема прецизионного усилителя. Если при проектировании данной схемы предполагается использовать, например, операционный усилитель 153УД2, то полная развернутая схема этого устройства будет содержать порядка трехсот элементов типа источников тока, резисторов и емкостей. В то же время свойства данного усилителя с достаточной точностью можно проанализировать, используя макромодель, сформированную на основе квазиидеальной макромодели операционного усилителя (ОУ). На рис. 11.15,б приведена эквивалентная схема полученной макромодели, а в табл. 11.1 даны ее параметры. Численные значения параметров получены с учетом параметров усилителя 153УД2 [48, 49].

Полученная аналитическим путем статическая макромодель устройства на операционных усилителях может успешно использоваться как в теоретическом исследовании, так и в программах автоматизированного проектирования, допускающих применение произвольных схем замещения.

Следует отметить, что подобная макромодель пригодна для решения задач параметрической оптимизации, так как она содержит резисторы, параметры которых могут варьироваться.

Пример 11.10. Макромодель операционного усилителя, учитывающая зависимость параметров от частоты и нелинейные искажения сигнала, показана

Таблица 11.1

Параметр	Численное значение	Формула для макроэлемента	
		идеального ОУ	неидеального ОУ
K_{U1}	0,999	1	$K_U/(1+K_U)$
K_{U2}	-49,98	$-R_2/R_1$	$(-R_2/R_1) \times K_U/(1+K_U+R_2/R_1)$
$R_{вх1}$	$12 \times 10 \text{ кОм}$	∞	$R_{вх} K_U$
$R_{вх2}$	$1,0003 \text{ кОм}$	R_1	$R_1 + (R_1+R_2)/K_U$
$R_{вых1}$	$2 \times 10 \text{ кОм}$	0	$R_{вых}/K_U$
$R_{вых2}$	$1 \times 10 \text{ кОм}$	0	$R_{вых}(1 - R_2/R_1)/K_U$

на на рис. 11.16. Эквивалентная схема макромодели состоит из четырех каскадов. Входной каскад моделирует частотные зависимости входного сопротивления, а также ослабление сигнала. Первый промежуточный каскад отражает конечную скорость изменения выходного напряжения и разностный коэффициент усиления. Наиболее сложно описание управляемого источника $I_1(U_{вх})$, аппроксимирующего передаточную характеристику с учетом основной и высших гармоник, появляющихся в выходном сигнале при больших амплитудах входного сигнала. Второй промежуточный каскад, имеющий единичный коэффициент усиления, моделирует высокочастотный полюс. Выходной каскад макромодели представляет выходной импеданс индуктивного характера и учитывает ограничение амплитуды выходного напряжения.

Пример 11.11. Макромодель операционного усилителя, представленная на рис. 11.17, также предназначена для анализа схем в частотной области. Нелинейные искажения, возникающие при больших уровнях входного сигнала,

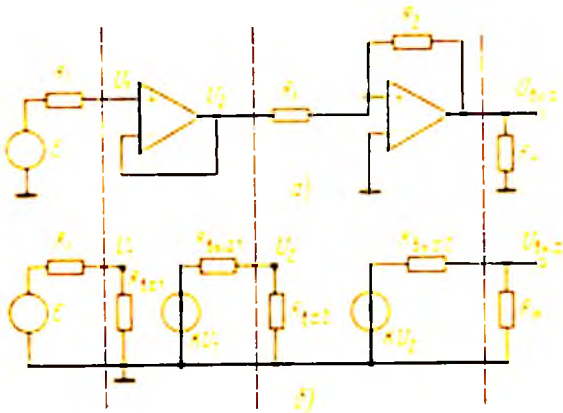


Рис. 11.15. Макромодель двухкаскадного прецизионного усилителя;
а — функциональная схема усилителя; б — эквивалентная схема макромодели

учитываются кусочно-линейной аппроксимацией передаточной характеристики, которая заложена в описании управляемого источника тока I :

$$I = g(U_{вх})U_{вх},$$

где при задании характеристики $(2p-1)$ отрезками

$$g(U_{вх}) = \sum_{k=1}^p (m_k - m_{k+1}) f(\delta_k / |U_{вх}|);$$

здесь f — аппроксимирующая функция.

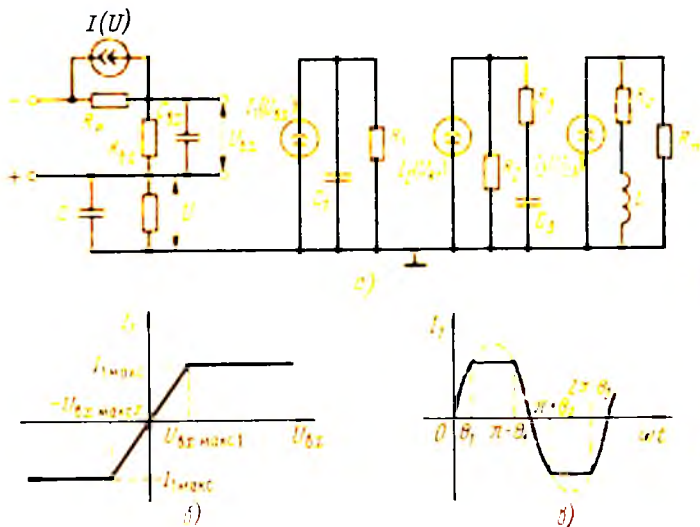


Рис. 11.16. Динамическая макромодель операционного усилителя, учитывающая нелинейные искажения сигнала:

a — четырехкаскадная эквивалентная схема макромодели; b — вольт-амперная характеристика управляемого источника $I_1(U_{вх})$; $в$ — реакция $I_1(U_{вх})$ на гармоническое воздействие

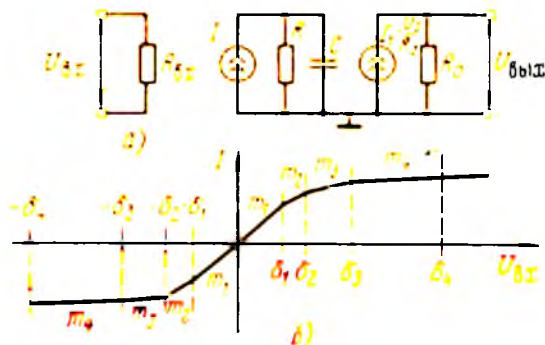


Рис. 11.17. Макромодель операционного усилителя (a), в которой используется кусочно-линейная аппроксимация передаточной характеристики, задаваемой управляемым источником тока $I(U_{вх})$ (b)

Одновременно с нелинейными искажениями учитывается и ограничение скорости изменения выходного напряжения.

Макромодели аналоговых схем, как правило, строятся поблоч-но, что соответствует структуре самих схем. Современные анало-говые ИС отличаются регулярностью структуры. Это позволяет выделить в качестве типовых следующие каскады: дифферен-циальный усилитель, отражатель тока, выходной усилитель, про-межуточный усилительный каскад по схеме с общим эмиттером. Для типовых каскадов разрабатываются модели, при построении которых используются необходимые модели активных полупровод-никовых элементов с учетом режима включения, диапазона сиг-налов и т. п. Модели типовых каскадов совместно с рядом управ-ляемых источников образуют базовый набор макроэлементов, ис-пользуемый для создания макромоделей сложных аналоговых и цифро-аналоговых устройств [50].

При разработке базового набора макроэлементов целесообраз-но создание иерархического ряда моделей типовых каскадов ана-логовых схем, различающихся уровнем сложности и точностью. Это обеспечивает возможность формирования диалоговых про-цедур для получения на ЭВМ макромоделей, обладающих задан-ными характеристиками.

В заключение следует отметить, что использование макромо-делей аналоговых схем позволяет повысить эффективность анали-за в процессе автоматизированного схемотехнического проектиро-вания более, чем на порядок.

ГЛАВА 12

МЕТОДЫ РАСЧЕТА И АНАЛИЗА ВЫХОДНЫХ ПАРАМЕТРОВ СХЕМ

12.1. РАСЧЕТ ВЫХОДНЫХ ПАРАМЕТРОВ СХЕМ

12.1.1. Одновариантный расчет или анализ схем. Задачей ана-лиза является определение всех интересующих разработчика вы-ходных параметров схемы в статическом и динамическом режимах и при расчете частотных характеристик.

В нелинейных переключательных схемах наиболее распростра-ненными выходными параметрами являются потребляемая мощ-ность, помехоустойчивость, длительности фронтов, срезов, задер-жек и импульсов. Алгоритмы вычисления статических выходных параметров представляют собой функциональные зависимости, со-держащие в качестве аргументов токи и напряжения и позволяю-щие найти заданный выходной параметр. Например, мощность, рассеиваемая в какой-либо ветви схемы, определяется выраже-нием $P=IU$, где I и U — значения тока и напряжения в заданной ветви.

В динамическом режиме средняя мощность может быть рассчитана как средняя сумма мгновенных мощностей, рассеиваемых на каждом шаге переходного процесса,

$$P_{\text{дин}} = \frac{1}{n} \sum_{k=1}^n I_k U_k,$$

где n — число шагов на интервале времени T .

Динамические выходные параметры (длительности задержек, фронтов, срезов, крутизна сигнала и период его следования) рассчитываются на основе оценки изменений во времени значений токов и напряжений. Например, длительность задержки фронта t_s некоторого сигнала вычисляется путем сравнения текущего значения Y с заданным уровнем Y_1 (рис. 12.1,а). Особенность алгоритмов расчета характеристик моментов времени состоит в том, что значения токов и напряжений вычисляются дискретно, поэтому возникает необходимость запоминать значения Y и после выполнения соответствующих условий проводить линейную интерполяцию определяемого момента времени (рис. 12.1,б).

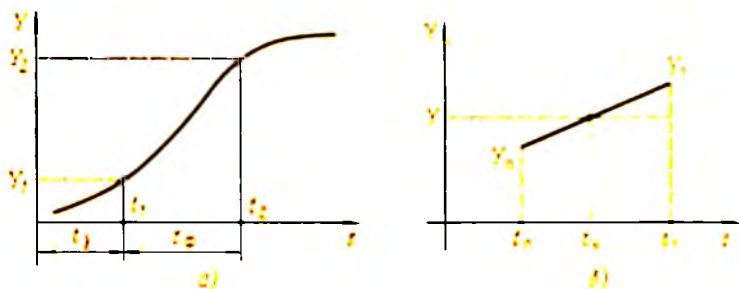


Рис. 12.1. Определение длительности задержки t_s и фронта t_f сигнала

Пример 12.1. Рассмотрим алгоритм определения длительности фронта сигнала между двумя уровнями Y_1 и Y_2 (рис. 12.1,а). В данном примере длительность фронта t_f определяется как разность моментов времени t_2 и t_1 . Каждый из этих моментов вычисляется в точке пересечения заданного уровня Y и прямой, полученной линейной интерполяцией сигнала между точками Y_r и Y_n (рис. 12.1,б), соответствующими сигналу на текущем и предыдущем шаге интегрирования. Искомая точка на оси времени вычисляется по формуле

$$t_Y = t_r - \frac{Y_r - Y}{Y_r - Y_n} \Delta t,$$

где Δt — шаг интегрирования.

повышения точности определения t_f можно также использовать квадратичную интерполяцию.

Длительность импульса t_n рассчитывается как разность $t_1 - t_s$, а длительность задержки t_s фронта сигнала, а t_1 — момент време-

ни пересечения определенного уровня (абсолютного или относительного) срезом сигнала.

Для расчета амплитуды сигнала Y_m необходимо вычислять и запоминать на каждом шаге переходного процесса наибольшее и наименьшее значения Y . При этом могут возникнуть погрешности в определении амплитуды из-за выбросов, образующихся, например, при прохождении сигнала через проходные емкости.

Для радиотехнических схем расчет большинства выходных параметров, характеризующих переходные процессы, базируется на оценке искажений различных сигналов при их прохождении через схему [44]. К таким выходным параметрам относятся (рис. 12.2):

τ_{ϕ} — длительность фронта или время установления импульса, определяемые временем нарастания сигнала от уровня 0,1 до 0,9 стационарного значения; τ_c — длительность среза импульса, определяемая аналогично времени нарастания; δ — выброс на вершине, т. е. максимальное превышение мгновенного значения сигнала над установившимся; Δ — неравномерность вершины импульса; l — выброс в паузе импульса; τ_a — запаздывание импульса, определяемое временем достижения половины установившегося значения высоты импульса или его амплитуды; τ_n — время длительности выброса, возникающего в паузе импульса.

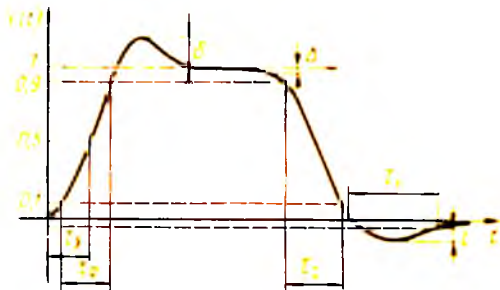


Рис. 12.2. Выходные параметры радиотехнических схем во временной области

Выходные параметры, наиболее часто рассчитываемые и контролируемые для радиотехнических схем в частотной области, сведены в табл. 12.1.

12.1.2. Многовариантный расчет схем. Простейшим видом анализа, необходимым для любого комплекса программ схемотехнического проектирования, является многовариантный расчет, т. е. многократное вычисление выходных параметров Y для заданных наборов (вариантов) внутренних параметров X . Многовариантный расчет позволяет разработчику провести всестороннее изучение схемы, определить зависимость выходных параметров Y от внутренних параметров X , от разброса этих параметров ΔX в широком диапазоне внешних воздействий G (температурных, радиационных и т. д.) и получить семейства характеристик $Y=f(X, G)$. Только при наличии таких семейств разработчик может судить о поведении схемы во всей области работоспособности, о возможностях реализации схемы на данных элементах и о соответствии схемы техническому заданию (ТЗ).

В результате многовариантного расчета уточняется также рельеф функции, описывающей исследуемый выходной параметр,

Таблица 12.1

Выходной параметр	Способ определения
Коэффициент усиления по напряжению, току или мощности K_0	Определяется как максимальный коэффициент усиления в пределах полосы пропускания или на центральной частоте полосы пропускания f_0
Полоса пропускания Δf	Соответствует области частот, где коэффициент усиления не опускается ниже $K_{гр}$, где $K_{гр} = 0,7K_0$ по напряжению или току и $K_{гр} = 0,5K_0$ по мощности
Верхняя f_v и нижняя f_n граничные частоты полосы пропускания	Обычно задаются на уровне $0,7K_0$
Избирательность δ_k	$\delta_k = (f - f_0) / (\Delta f_{0,7}/2)$, где f — частота, соответствующая фиксированному уровню подавления сигнала K_0 (обычно $K_0 = 0,1$ или $0,01$)
Наклон фазочастотной характеристики ψ	Рассчитывается на фиксированной частоте f_0 : $\psi = \left. \frac{\partial \varphi(f)}{\partial f} \right _{f=f_0}$, где $\varphi(f)$ — уравнение фазовой характеристики
Линейность фазочастотной характеристики в диапазоне частот γ	$\gamma = (\varphi(f) - \varphi(f_0)) / (\varphi(f_1) - \varphi(f_0)) - 1$, где f_1 — частота определения γ
Коэффициент нелинейных искажений K_{Γ}	$K_{\Gamma} = \sqrt{\sum_{i=2}^{\infty} P_i / P_1}$, где P_i — мощность i -й гармоники; P_1 — мощность первой гармоники

что дает возможность выбирать наиболее подходящий метод оптимизации, область работоспособности схем и начальные условия для оптимизации схемы. По этой причине, проведение многовариантного расчета должно предшествовать этапу оптимизации при проектировании.

12.2 АНАЛИЗ ЧУВСТВИТЕЛЬНОСТИ

12.2.1. Постановка задачи. В общем виде под чувствительностью понимают реакцию схемы на малое изменение ее внутренних параметров X . Количественная оценка такого изменения выходного параметра схемы y при заданном изменении внутреннего параметра x называется коэффициентом чувствительности. Анализ

чувствительности необходим при определении требований к точности реализации тех или иных внутренних параметров схемы, а следовательно, и к технологии изготовления элементов с целью увеличения процента выхода годных схем, а также при проектировании схем, работающих в сложных условиях окружающей среды, т. е. подверженных ускоренному старению и температурным изменениям.

Цель анализа чувствительности заключается в нахождении тех элементов схемы и параметров X этих элементов, отклонение которых от номинальных значений приводит к наибольшему отклонению выходных параметров схемы Y .

При анализе чувствительности находится матрица чувствительности $A[m \times n]$, элементами которой являются коэффициенты чувствительности выходного параметра y_j к изменениям внутреннего параметра x_i : $A_{ji} = \partial y_j / \partial x_i$.

$$A = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \dots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \quad (12.1)$$

Каждая j -я строка матрицы A является вектором-градиентом j -го выходного параметра y_j в пространстве внутренних параметров X , а каждый i -й столбец характеризует влияние i -го внутреннего параметра x_i на все выходные параметры Y .

По значениям коэффициентов чувствительности разработчик может определить направление изменений внутренних параметров для улучшения выходных параметров, оценить влияние каждого внутреннего параметра на выходные параметры и определить сильно влияющие для того, чтобы отделить их от слабо влияющих.

При проектировании РЭА аналитические зависимости, явно выражающие коэффициенты чувствительности через внутренние параметры X схемы, могут быть получены в весьма редких случаях, только при использовании аналитических моделей схемы. В общем случае модель схемы представляется алгоритмически, аналитические зависимости неизвестны и для анализа чувствительности применяются численные методы.

Исследование чувствительности, как правило, не проводится самостоятельно, так как требует больших вычислительных затрат, а выполняется в сочетании с одним из основных видов анализа (статического, динамического, частотного, статистического), который оказывает определяющее влияние на выбор метода расчета коэффициентов чувствительности.

12.2.2. Метод приращений. Наиболее простым и поэтому распространенным при проектировании методом расчета чувствитель-

ности является метод приращений, основанный на численном дифференцировании. В этом методе коэффициент чувствительности j -го выходного параметра к изменению i -го внутреннего параметра определяется по формуле

$$\frac{\partial y_j}{\partial x_i} = \frac{\Delta y_j}{\Delta x_i} = \frac{y_j(x_1, x_2, \dots, x_i + \Delta x_i, \dots, x_n) - y_j(x_1, x_2, \dots, x_n)}{\Delta x_i} \quad (12.2)$$

Алгоритм расчета методом приращений по (12.2) включает в себя следующие этапы.

1. Расчет выходных параметров в номинальном режиме, т. е. вычисление вектора $Y(x_1, x_2, \dots, x_n)$.

2. Выполнение n вариантов расчета, в каждом из которых дается отклонение Δx_i от номинального значения x_i только одному из внутренних параметров x_i . При этом каждый раз получается вектор выходных параметров $Y(x_1, x_2, \dots, x_i + \Delta x_i, \dots, x_n)$, каждая j -я компонента которого определяет коэффициент чувствительности данного j -го выходного параметра к изменению очередного i -го внутреннего параметра.

Очевидно, что анализ методом приращений требует выполнения $(n+1)$ варианта расчета схемы, где n — число внутренних параметров, по которым исследуется чувствительность схемы. Этот большой объем вычислений требует соответствующих затрат машинного времени, что является одним из недостатков метода приращений.

Более существенным недостатком метода приращений является невысокая точность определения коэффициентов чувствительности, что обусловлено нелинейным характером зависимостей выходных параметров $Y=f(x_1, x_2, \dots, x_n)$ от внутренних параметров, тогда как (12.2) основана на линейном приближении. Поэтому чем сильнее нелинейность зависимости $Y=f(X)$, тем существеннее погрешность. Для увеличения точности существуют два способа, один из которых заключается в выполнении для каждого внутреннего параметра x_i двух вариантов расчета путем отклонения Δx_i в обе стороны от номинального значения. При этом рассчитываются два значения y_j : y_j^+ и y_j^- . Тогда коэффициент чувствительности более точно определяется по формуле

$$\partial y_j / \partial x_i \approx (y_j^+ - y_j^-) / (2 \Delta x_i). \quad (12.3)$$

Однако при этом увеличивается число необходимых вариантов расчета схемы и становится равным $2n$ вместо $(n+1)$ в предыдущем случае.

Второй способ заключается в уменьшении Δx_i , что ведет к более оправданному допущению о линейности выходных параметров. Но в свою очередь использование малых приращений внутренних параметров Δx_i приводит к необходимости уменьшения шагов интегрирования h при решении системы дифференциальных уравнений схемы. Уменьшение шага интегрирования h ведет также к значительному увеличению вычислительных затрат на расчет коэффициентов чувствительности.

Несмотря на рассмотренные недостатки, метод приращений наиболее широко распространен для расчета чувствительности [2, 28], так как применим для расчета коэффициентов чувствительности любых выходных параметров y , к изменению любых внутренних параметров схемы x_i . Кроме того, реализация метода чрезвычайно проста и допускает возможность распараллеливания вычислительного процесса, что может оказаться весьма важным при переходе на ЭВМ с параллельной обработкой информации.

12.2.3. Метод присоединенной схемы. Рассматриваемый ниже метод, в отличие от метода приращений, позволяет существенно снизить вычислительные затраты на расчет коэффициентов чувствительности. Метод присоединенной схемы [28, 51] наиболее эффективен при расчете статических выходных параметров схемы. В основе этого метода лежит теорема Теллегена [32], доказывающая, что алгебраическая сумма произведений токов и напряжений всех ветвей схемы равна нулю:

$$\mathbf{U}^t \mathbf{I} = \mathbf{I}^t \mathbf{U} = 0 \text{ или } \sum_{k=1}^n U_k I_k = 0, \quad (12.4)$$

где U_k, I_k — напряжение и ток k -й ветви схемы; n — число ветвей.

Метод присоединенной схемы основан на справедливости соотношения (12.4) и для случая, когда вектор напряжений $\mathbf{U}$ и вектор токов $\mathbf{I}$ принадлежат различным схемам, имеющим одинаковую топологию:

$$\sum_{k=1}^n U_k \hat{I}_k = 0, \quad \sum_{k=1}^n \hat{U}_k I_k = 0. \quad (12.5)$$

Здесь I_k и U_k — ток и напряжение k -й ветви исходной схемы N , а $\hat{I}_k$ и $\hat{U}_k$ — ток и напряжение соответствующей ветви схемы $\hat{N}$, т. е. ветви, включенной в схеме $\hat{N}$ между теми же узлами, что и k -я ветвь в исходной схеме.

Если в исходной схеме напряжения и токи ветвей изменились на dU_k и dI_k соответственно, то из (12.5) получаем, что

$$\sum_{k=1}^n (U_k + dU_k) \hat{I}_k = 0, \quad \sum_{k=1}^n (I_k + dI_k) \hat{U}_k = 0$$

или

$$\sum_{k=1}^n dU_k \hat{I}_k = 0, \quad \sum_{k=1}^n \hat{U}_k dI_k = 0.$$

Отсюда следует, что

$$\sum_{k=1}^n (dU_k \hat{I}_k - dI_k \hat{U}_k) = 0. \quad (12.6)$$

Соотношение (12.6) лежит в основе метода присоединенной схемы.

Для получения присоединенной схемы на основе (12.6) необходимо при расчете статического режима схемы закоротить все источники напряжения; на выходе схемы подключить единичный

Необходимо отметить, что метод присоединенной схемы позволяет таким образом вычислить чувствительность только одного выхода. При наличии нескольких выходов необходимо производить расчет статического режима присоединенной схемы соответствующее число раз (так как на выходе присоединенной схемы каждый раз должен быть подключен единичный источник), в результате этого объем вычислений может приближаться к объему, требуемому для двукратного расчета исходной схемы.

ния к выходу схемы источника $I = \delta(t - t_k)$ вместо единичного источника в статическом режиме.

Пример 12.2. Рассмотрим применение (12.6) для расчета чувствительности выходного напряжения в статическом режиме к изменениям внутренних параметров схемы, изображенной на рис. 12.3 [28]. В соответствии с принятыми допущениями введем на выходе исходной схемы разомкнутую ветвь, параллельную резистивной

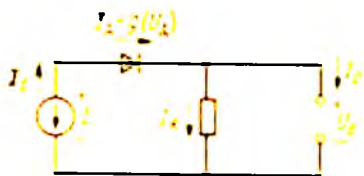


Рис. 12.3. Исходная схема для анализа чувствительности методом присоединенной схемы

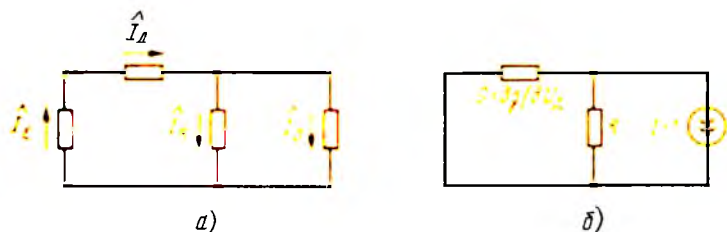


Рис. 12.4. Присоединенная схема:
а — с неизвестными параметрами; б — с определенными параметрами

ветви. Исследуем некоторую схему, изображенную на рис. 12.4,а, которая имеет такую же конфигурацию, как и исходная, но обладает неизвестными характеристиками ветвей. Составим для этих схем (рис. 12.3 и рис. 12.4,а) уравнение по методу присоединенной схемы (12.6)

$$(dE\dot{I}_E - dI_E\dot{E}) + (dU_R\dot{I}_R - dI_R\dot{U}_R) + (dU_D\dot{I}_D - dI_D\dot{U}_D) + (dU_0\dot{I}_0 - dI_0\dot{U}_0) = 0.$$

В полученном соотношении в соответствии с принятыми допущениями для расчета в статическом режиме положим $E=0$, $I=1$ и учтем, что

$$dE=0, \quad dI_0=0, \quad dI_D = \frac{\partial g}{\partial U_D} dU_D + \frac{\partial g}{\partial P} dP, \\ dU_R = \frac{\partial U_R}{\partial R} dR + \frac{\partial U_R}{\partial I_R} dI_R = I_R dR + R dI_R.$$

В результате получим уравнение

$$I_R dR \dot{I}_R + R dI_R \dot{I}_R + dU_0 \dot{I}_0 - dI_R \dot{U}_R + \left(dU_D \dot{I}_D - \frac{\partial g}{\partial I_D} dU_D \dot{U}_D - \frac{\partial g}{\partial P} dP \dot{U}_D \right) = 0.$$

Преобразуем его к виду

$$\{I_R \dot{I}_R dR + (R \dot{I}_R - \dot{U}_R) dI_R\} + \left[\left(\dot{I}_D - \frac{\partial g}{\partial I_D} \dot{U}_D \right) dU_D - \frac{\partial g}{\partial P} dP \dot{U}_D \right] + dU_0 \dot{I}_0 = 0.$$

Из (12.7) видно, что если в присоединенной схеме выполняются условия

$$R \dot{I}_R - \dot{U}_R = 0, \quad \dot{I}_D - \frac{\partial g}{\partial I_D} \dot{U}_D = 0,$$

т. е. если резистор остается без изменений, а на месте диода ставится линейная проводимость $G = \partial g / \partial U_D$, то чувствительность выходного напряжения по параметрам схемы определяется по следующим простым выражениям:

$$\frac{\partial U_0}{\partial R} = -I_R \dot{I}_R, \quad \frac{\partial U_0}{\partial P} = \frac{\partial g}{\partial P} \dot{I}_D.$$

Получившаяся присоединенная схема показана на рис. 12.4,б.

12.3. МЕТОД РАСЧЕТА НА НАИХУДШИЙ СЛУЧАЙ

12.3.1. Постановка задачи. Цель расчета на наихудший случай состоит в определении вектора выходных параметров $Y_{nc} = (y_{1nc}, y_{2nc}, \dots, y_{mnc})$, компоненты которого $y_{i\ nc}$ наименее благоприятны с точки зрения выполнения требований технического задания или наихудшие среди всех возможных значений выходных параметров. Расчет на наихудший случай является простейшим видом определения разброса выходных параметров без оценки плотности распределения этих параметров. Получаемые с помощью этого расчета оценки рассеяния значений выходных параметров сильно завышены, что и определяет название метода. Метод применяется, если известны предельно возможные отклонения Δx_j внутренних параметров от своих номинальных значений $x_{j\ ном}$. Обычно значения Δx_j , являющиеся допусками на значения внутренних параметров

ров, указываются в справочниках или паспортных данных на устройство. Данный метод удобен тем, что для его применения не требуется знания законов распределения внутренних параметров X и числовых характеристик этих законов.

12.3.2. Алгоритм расчета на наихудший случай. Пусть задано условие работоспособности i -го выходного параметра в виде $y_i < y_{i \text{ доп}}$. С точки зрения невыполнения данного неравенства или условия работоспособности интерес представляют наибольшие значения y_i или верхняя граница диапазона рассеяния. Выходной параметр y_i достигает верхней границы своего диапазона рассеяния, когда все аргументы функциональной зависимости $y_i = f(X)$ принимают самые неблагоприятные значения.

Самым неблагоприятным значением аргумента x_j будет его максимально возможное значение $x_{j \text{ макс}} = x_{j \text{ ном}} + \Delta x_j$, если $dy_i/dx_j > 0$, и минимально возможное значение $x_{j \text{ мин}} = x_{j \text{ ном}} - \Delta x_j$, если $dy_i/dx_j < 0$. В том случае, если условие работоспособности задано в виде $y_i > y_{i \text{ доп}}$, самым неблагоприятным значением аргумента x_j будет $x_{j \text{ мин}} = x_{j \text{ ном}} - \Delta x_j$, если $dy_i/dx_j > 0$ и $x_{j \text{ макс}} = x_{j \text{ ном}} + \Delta x_j$, если $dy_i/dx_j < 0$. Здесь Δx_j — допуск на внутренний параметр x_j .

Алгоритм строится исходя из предположения, что знаки производных остаются неизменными во всей рассматриваемой области, и включает в себя следующие процедуры.

1. Анализ чувствительности, в результате которого определяются знаки коэффициентов чувствительности $\text{sign}(dy_i/dx_j)$ для всех выходных параметров.

2. Присвоение внутренним параметрам x_j самых неблагоприятных значений:

$x_{j \text{ нс}} = x_{j \text{ ном}} - \text{sign}(dy_i/dx_j) \Delta x_j$ для условия работоспособности $y_i > y_{i \text{ доп}}$;

$x_{j \text{ нс}} = x_{j \text{ ном}} + \text{sign}(dy_i/dx_j) \Delta x_j$ для условия работоспособности $y_i < y_{i \text{ доп}}$.

3. Выполнение анализа схемы, т. е. расчета выходного параметра y_i для наихудшего случая.

Для схемы, характеризуемой m выходными параметрами, процедуры 2 и 3 выполняются m раз (каждый раз для i -го выходного параметра). Если число внутренних параметров равно при этом n , то проведение анализа чувствительности (т. е. процедуры 1) методом приращений требует $n+1$ обращения к модели схемы. Всего для проведения анализа на наихудший случай потребуется $m + n + 1$ обращение к модели схемы.

12.3.3. Преимущества и недостатки метода расчета на наихудший случай. Преимуществами метода являются его простота, небольшое количество обращений к модели схемы и возможность расчета допусков на выходные параметры без знания законов распределения внутренних параметров. Но так как вероятность сочетания внутренних параметров схемы, близких к самым неблагоприятным, в реальных схемах пренебрежимо мала, то оценки рассеяния параметров по методу наихудшего случая оказываются значительно завышенными по сравнению с более точными методами

статистических испытаний. Вследствие этого многие практически работоспособные варианты схемы представляются неработоспособными по результатам такого расчета.

Поэтому метод расчета на наихудший случай редко используется в САПР, где получили распространение более точные вероятностные методы. Можно рекомендовать применение метода расчета на наихудший случай только для предварительной оценки разброса выходных параметров схемы.

12.4. СТАТИСТИЧЕСКИЙ АНАЛИЗ

12.4.1. Постановка задачи. Целью статистического анализа является определение процента выхода годных схем, соответствующих техническому заданию при данном конкретном разбросе параметров ΔX .

Статистический анализ схемы сводится к расчету вероятности $P(X)$ того, что вектор внутренних параметров X , определяющий состояние схемы в момент ее изготовления, находится в области работоспособности $G(X)$. Исходной информацией для статистического анализа являются характеристики законов распределения внутренних параметров X , а результатом расчета — характеристики законов распределения выходных параметров Y . Необходимо отметить, что с течением времени и изменением условий эксплуатации числовые характеристики не остаются стабильными. Учет их изменений является весьма сложной задачей, решаемой на этапе статистической оптимизации схемы. Статистический анализ схемы ограничивается лишь расчетом начальной надежности схемы, т. е. не учитывается изменения параметров схемы с течением времени.

12.4.2. Метод статистических испытаний (метод Монте-Карло).

Алгоритм статистического анализа схем методом Монте-Карло представлен на рис. 12.5. Здесь N — запланированное число статистических расчетов, а k — номер очередного расчета схемы или испытания. Как видно из рисунка, сущность метода заключается в многократном повторении расчета схемы, при котором внутренние параметры схемы X моделируются по заданному случайному закону.

Моделирование случайных значений внутренних параметров

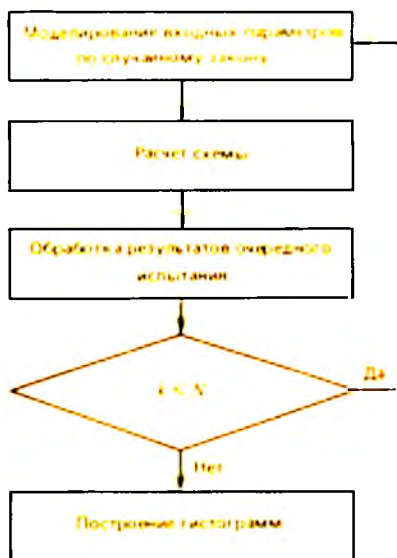


Рис. 12.5. Блок-схема алгоритма статистического анализа методом Монте-Карло

X производится с помощью библиотеки законов распределения, имеющейся в любом комплексе программ САПР. Программа моделирования случайных значений равномерно распределенной в интервале $[0, 1]$ величины, являющейся основой для получения законов распределения произвольного вида, имеется в составе общего программного обеспечения любой ЭВМ так же, как и программы получения законов нормального и экспоненциального видов. Каждый расчет эквивалентен как бы одному испытанию конкретной реальной схемы. Это испытание считается успешным, если все условия работоспособности удовлетворяются, и неудачным, если нарушено хотя бы одно условие работоспособности. Если из общего числа N назначенных испытаний M испытаний явились успешными, то отношение M/N с определенной точностью и характеризует искомую вероятность $P(X)$ получения годных схем или процент выхода годных схем.

Результатом статистического анализа также являются гистограммы выходных параметров Y и оценки числовых характеристик их распределений (например, оценки математических ожиданий M_Y выходных параметров и их среднеквадратических отклонений σ_Y при нормальном законе распределения). Обычно гистограммы строятся после проведения всей серии, содержащей N испытаний. Простейший алгоритм построения гистограмм требует запоминания всех вычисленных значений вектора Y для всех N испытаний в памяти ЭВМ. Тогда после проведения серии испытаний оценки математического ожидания j -го выходного параметра M_j и его среднеквадратического значения σ_j производятся по формулам:

$$M_j = \sum_{k=1}^N y_{jk}/N; \quad \sigma_j^2 = \sum_{k=1}^N (y_{jk} - M_j)^2/(N-1),$$

После расчетов M_j и σ_j осуществляется построение гистограммы распределения j -го выходного параметра y_j .

Если схема имеет m выходных параметров, то в памяти ЭВМ необходимо хранить двумерный массив (m, N) результатов испытаний. Память ЭВМ можно сэкономить, если можно заранее предсказать поле рассеяния каждого параметра, разделить его на необходимое число интервалов гистограммы и фиксировать попадание каждого испытания в соответствующий интервал путем добавления единицы в счетчик числа испытаний данного интервала. Математическое ожидание и среднеквадратическое отклонение можно подсчитывать, не запоминая после проведения каждого k -го испытания значений всех выходных параметров, а лишь подсчитывая статистические суммы $\sum_{i=1}^{k-1} y_{ji}$ и $\sum_{i=1}^{k-1} y_{ji}^2$ путем добавления к ним каждый раз значений y_{jk} и y_{jk}^2 соответственно.

12.4.3. Преимущества и недостатки метода статистических испытаний. Преимущество метода статистических испытаний заключается в возможности расчета с его помощью сложных интегральных схем с большим числом корреляционных связей между па-

параметрами с высокой точностью, что невозможно никакими другими известными методами. Однако для реализации этой возможности необходимо знание статистических сведений о параметрах, что в настоящее время является практически трудновыполнимым требованием.

Кроме этого, при проведении статистического анализа достаточные трудности представляет правильный выбор числа испытаний N : при большом N возможно получение высокой точности и надежности расчетов, но требуются большие затраты машинного времени, а малое число испытаний N может привести к недостаточной точности анализа. Практически для расчета с ошибкой 5...10% достаточно 500...1000 испытаний статистическим методом [2], что вполне приемлемо при существующем быстродействии современных ЭВМ.

12.5. ЛИНГВИСТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММ РАСЧЕТА И АНАЛИЗА ВЫХОДНЫХ ПАРАМЕТРОВ СХЕМ

12.5.1. Требования к языкам описания задания на расчет и анализ выходных параметров схем. Расчет выходных параметров схемы заключается в определении токов и напряжений в схеме в статическом и динамическом режимах и расчете на их основе параметров, характеризующих функциональные свойства схемы.

В задании на расчет выходных параметров указываются тип выходного параметра (фронт, чувствительность, мощность и т. д.) и параметры подпрограммы для его вычисления (имя подпрограммы, данные об элементах, для которых вычисляется параметр, например номер узла или ветви, параметры алгоритма вычислений, например верхний и нижний уровни, между которыми отсчитывается фронт сигнала).

Для уменьшения типов описания различных выходных параметров в некоторых языках применяют иерархическую структуру описания в виде конструкции $Y = Y_{3i}(Y_{2j}(Y_{1k}))$, где Y_{1k} — выходной параметр нижнего, первого уровня, например ток, напряжение или потенциал в статике или на одном шаге динамики, или на одной частоте; Y_{2j} — выходной параметр второго уровня, например некоторое j -е множество или таблица значений параметров первого уровня, а Y_{3i} — выходной параметр третьего уровня, например i -й вид обработки j -го множества.

Так, расчет фронта потенциала в узле 2 между уровнями 2 В и 3 В можно описать следующей конструкцией:

Фронт (верхний уровень = 3 В, нижний уровень = 2 В (переходный процесс на интервале 250 нс (потенциал на шаге динамики в узле 2))).

Здесь «Фронт» — трехуровневая иерархическая процедура, причем на первом уровне рассчитывается потенциал на каждом шаге, на втором — накапливается множество значений потенциала в течение 250 нс, а на третьем — выполняется обработка этого множе-

ства для расчета длительности фронта потенциала между 2 В и 3 В. Заметим, что задание уровней при расчете фронтов — непостоянная задача, так как необходимо, чтобы график переходного процесса пересекал эти уровни, иначе фронт рассчитан не будет.

Описание задания на анализ параметров состоит из двух частей. В первой части описывается информация о варьируемых в процессе анализа элементах или их параметрах. Например, в задании на многовариантный анализ описываются сами варьируемые элементы или их параметры, диапазон и шаг варьирования, в задании на статистический анализ указываются законы распределения элементов (нормальный, равномерный и др.) и параметры законов. Во второй части описываются тип и параметры подпрограммы, выполняющей анализ выходных параметров. Например, в задании на статистический анализ методом Монте-Карло указываются количество испытаний, шаг гистограммы и т. д.

12.5.2. Структура директив на расчет выходных параметров схем. Директивы этого типа состоят из имени директивы, ее параметров и аргументов. Под параметрами будем понимать численные значения, задающиеся в исходных данных на расчет, а под аргументами — величины, вычисляемые в процессе расчета. Аргументами могут служить как токи, напряжения и потенциалы схемы, так и выходные параметры, вычисленные при их использовании. В директиве каждый аргумент задается символьным обозначением, определяющим его тип и место хранения в программе его вычисленного значения.

Если аргументами служат токи, напряжения и потенциалы, то они обозначаются, например, символами I , U и V соответственно, при этом также указывается элемент или номер узла, например $IC5$, $UR11$, $V7$. Вычисляемые выходные параметры могут задаваться символом Y и номером в конструкции объединения директив, например $Y1$, $Y2$ и т. д.

Пример 12.3. Приведем описание многоуровневой директивы расчета [33].
ПРОВЕСТИ РАСЧЕТ PAR Z:

$$Z1 = \text{ИМЯ } 1 (A1, A2, \dots) Y1, Y2, \dots$$

$$Y1 = \text{ИМЯ } 2 (B1, B2, \dots) Y2, Y3, \dots$$

$$YN = \text{ИМЯ } K (D1, D2, \dots) UR1, UC1, \dots$$

Первая директива формирует первый выходной параметр схемы с помощью записанных в скобках параметров $A1$, $A2$, ... и аргументов, записанных после скобок и характеризующихся выходными параметрами $Y1$, $Y2$, ... Каждый из аргументов $Y1$, $Y2$, ... в свою очередь задается соответствующей директивой.

Приведем в качестве примера формирования задания на расчет выходного параметра директиву расчета задержки сигнала, рассматриваемого на интервале времени $T_n - T_k$, от заданного момента времени T до момента пересечения уровня Y_P потенциалом $V1$:

$$Y1 = \text{ТЗАД}(T_n, T_k, T, Y_P) V1.$$

Вычисление максимального значения из набора значений, найденных на интервале времени или частоты $T_n - T_k$, может иметь вид

$$Y1 = \text{MAX}(T_n, T_k) UR1, UR2, UR3.$$

12.5.3. Структура директив на анализ выходных параметров схем. Аргументами директивы анализа служат помимо анализируемых выходных параметров варьируемые при анализе внутренние параметры схемы. Тогда структуру директивы анализа можно представить в следующем виде:

ПРОВЕСТИ АНАЛИЗ:

Имя анализа (A1, A2, ...) Z1, Z2, Z3, ...

X1) ИМЯ элемента 1 (ИМЯ парам) = ИМЯ 1 (B1, B2, ...)

XN) ИМЯ элемента N (ИМЯ парам) = ИМЯ N (C1, C2, ...)

Первая строка определяет собственно директиву анализа, имеющую символьное имя, набор параметров и анализируемые выходные параметры Z1, Z2, Z3, ... Каждая из последующих строк определяет местоположение в исходной информации варьируемого внутреннего параметра X, т. е. указывает имя элемента схемы и имя его изменяемого параметра. Например, выражение $T5(RB) = \dots$ означает, что варьируемым параметром является сопротивление базы RB транзистора T5. После знака равенства символьными обозначениями ИМЯ 1, ..., ИМЯ N задаются имена законов изменения соответствующих внутренних параметров, а в скобках указываются параметры этих законов.

Пример 12.4. Задание нескольких директив анализа выходных параметров Z2 и Z3 при изменении сопротивления базы RB транзистора T5, может иметь вид

ПРОВЕСТИ АНАЛИЗ:

МНОВОВАРИАНТНЫЙ Z2, Z3

X1) T5(RB) = ВАРИАНТЫ (0.1, 0.15, 0.2, 0.25)

ЧУВСТВИТЕЛЬНОСТИ Z2, Z3

X1) T5(RB) = ВАРИАНТЫ (0.15, 0.2)

СТАТИСТИЧЕСКИЙ (500, 20) Z2, Z3

X1) T5(RB) = РАВНОМЕРНЫЙ (0.1, 0.25)

ТЕМПЕРАТУРНЫЙ (-60, 40, 10) Z2, Z3

X1) T5(RB) = ЛИНЕЙНЫЙ (0.12, 0.2)

Первая директива на проведение многовариантного расчета предполагает вычисление выходных параметров Z2 и Z3 при перечисленных вариантах варьируемого внутреннего параметра RB транзистора T5. Вторая директива задает анализ чувствительности параметров Z2 и Z3 к изменению внутреннего параметра при двух вариантах его значения. Третья директива определяет статистический анализ, в том числе построение гистограмм выходных параметров в серии из 500 испытаний на 20 интервалах при изменении внутреннего параметра по равномерному закону в интервале от 0,1 до 0,25. И последней директивой задается температурный анализ при изменении под влиянием температуры внутреннего параметра по линейному закону (параметры закона 0,12 и 0,2) в диапазоне температур -60 ... +40° С с шагом 10° С.

Закон изменения внутренних параметров может быть также задан обращением к процедуре описания аналитического выражения, аргументом которого служит изменяемый при анализе параметр, например температура.

Возможны и другие способы задания информации на расчет и анализ выходных параметров [2].

ГЛАВА 13

КРИТЕРИИ ОПТИМАЛЬНОСТИ

13.1. ОСОБЕННОСТИ ЗАДАЧ ТЕХНИЧЕСКОЙ ОПТИМИЗАЦИИ

13.1.1. Область применения задач оптимизации. Завершающим этапом проектирования РЭА является оптимизация схем. Оптимизация состоит в определении такой совокупности внутренних параметров схемы (емкостей, сопротивлений, параметров активных элементов), при которой заранее выбранные выходные параметры (например, быстродействие, время задержки сигнала, потребляемая схемой мощность) принимают наилучшие возможные значения. Помимо этого оптимизация позволяет решить ряд попутных задач: определить внутренние параметры, обладающие наибольшими коэффициентами влияния на выходные характеристики, оценить влияние различного рода дестабилизирующих факторов, обнаружить невозможность функционирования схемы или получения заданных значений выходных параметров на основе используемой структуры схемы и элементной базы. С помощью оптимизации решаются также задачи статистического характера по увеличению вероятности безотказной работы схем или числа работоспособных схем при их производстве.

Решение любой задачи оптимизации схемы начинается с выбора и построения критерия оптимальности схемы или целевой функции и формулировки системы ограничений, в рамках которых отыскивается оптимальное решение.

13.1.2. Особенности задач оптимизации параметров РЭА. На практике формулировка критерия оптимальности и решение конкретной задачи расчета оптимальных параметров схемы нередко вызывает затруднения. Это определяется, в основном, следующими факторами.

1. Формализация задачи оптимизации схемы существенно сложнее формализации задачи ее расчета и анализа. Многое здесь основано на интуиции разработчика, на применении эвристических подходов и методов, которые трудно свести к стандартным задачам оптимизации и описать с помощью используемого входного языка комплекса программ схемотехнического проектирования.

2. Все параметры схемы в процессе оптимизации можно варьировать лишь в определенных пределах, т. е. следует соблюдать обычно большое число ограничений. Например, все пассивные эле-

менты могут принимать только положительные значения, при этом конструирование и технология их изготовления лимитирует как их значение, так и получаемую при их изготовлении точность, что тоже необходимо учитывать при оптимизации. Многие параметры схем являются зависимыми от других параметров, при этом эти зависимости зачастую характеризуются нелинейными и невыпуклыми функциями. Выходные параметры схемы не всегда могут принимать произвольные значения, а могут быть связаны какими-либо зависимостями с другими выходными параметрами или ограничениями. Все эти ограничения существенно затрудняют выбор и работу методов оптимизации и, главное, осложняют процесс формализации и сведения каждой конкретной задачи к общей задаче, описываемой в терминах математического программирования.

3. Как правило, все задачи оптимизации схем характеризуются большим количеством переменных, т. е. внутренних параметров схем. Даже в том случае, когда далеко не все эти параметры влияют на положение экстремума целевой функции и часть из них можно не варьировать в процессе оптимизации, они определяют большие размеры систем нелинейных дифференциальных уравнений, образующих математическую модель схемы. На решение таких систем уравнений требуются секунды или даже минуты машинного времени, а так как при оптимизации обычно требуется сотни раз рассчитывать математическую модель схемы, то затраты машинного времени на оптимизацию вырастают в реальных задачах до единиц и десятков часов.

4. Все реальные целевые функции сложны для вычислений и поиска экстремума. Это происходит из-за того, что, во-первых, все они не имеют аналитического представления, а формируются алгоритмически по точкам, получаемым при каждом обращении к математической модели схемы и численном решении системы дифференциальных уравнений, а во-вторых, что они, как правило, имеют сложную невыпуклую и недифференцируемую поверхность. То же самое относится и к ограничениям. Это существенно осложняет процесс поиска экстремума, требует применения более сложных и надежных методов оптимизации, которые в свою очередь требуют для своей реализации большего количества расчетов, что тоже резко увеличивает вычислительные затраты, необходимые для оптимизации.

5. Так как выходные параметры, ограничения и целевые функции носят сложный недифференцируемый характер и вычисляются алгоритмически, то не представляется возможным вычислять их производные по аналитическим формулам, а вычисление их численными методами конечных приращений приводит к высоким затратам машинного времени и значительным погрешностям вычислений. Это обстоятельство существенно ограничивает выбор методов оптимизации, затрудняя применение для оптимизации параметров электронных схем многих методов первого и второго порядка, требующих вычисления соответствующих производных.

6. Наличие у схемы большой совокупности выходных парамет-

ров приводит к необходимости считать ее оптимальной в случае, когда наилучшее значение принимает не каждый отдельно выбранный выходной параметр, а какая-либо их совокупность, в которой отдельные выходные параметры не принимают своего оптимального значения, а носят компромиссный характер. Сведение задач оптимизации с векторным критерием (каждый выходной параметр представляет собой компоненту данного вектора) к задаче оптимизации со скалярным критерием, представляющим собой определенным образом выбранную свертку критериев, будет приведено в последующих параграфах.

Пример 13.1. Рассмотрим оптимизацию цифровой схемы по нескольким критериям. Обычно этими критериями являются: потребляемая мощность P , время задержки сигнала t_z и помехоустойчивость $\Delta U_{\text{п}}$. Таким образом, даже в самом простейшем случае задача оптимизации является многокритериальной, изменяя какой-либо внутренний параметр схемы, например сопротивление R , можно одновременно улучшать один критерий — P и ухудшать два других — t_z и $\Delta U_{\text{п}}$.

Необходима свертка всех трех критериев в один, но таким образом, чтобы улучшение каждого из них не приводило бы к неприемлемому ухудшению остальных. Такая свертка может привести к резкому изменению вида целевой функции. Даже в самом простом случае, когда каждый отдельный критерий в области работоспособности имеет одноэкстремальный характер, сложный критерий может повлечь за собой появление многоэкстремальности, т. е. наличия нескольких экстремумов целевой функции в области работоспособности.

13.2. ОБЩИЕ СВЕДЕНИЯ О КРИТЕРИЯХ ОПТИМАЛЬНОСТИ

13.2.1. Частные критерии оптимальности. В простейшем случае в качестве критерия оптимальности $F(\mathbf{X})$, где $\mathbf{X}$ — вектор внутренних параметров схемы, выбирается один из выходных параметров схемы $f_j(\mathbf{X})$ и подвергается оптимизации (максимизации или минимизации) при наложении ограничений на остальные выходные параметры $f_i(\mathbf{X})$ ($i \neq j$). В этом случае он называется частным критерием оптимальности.

Например, схему можно максимизировать по быстродействию, накладывая определенные ограничения на потребляемую мощность, или, наоборот, минимизировать потребляемую схемой мощность, ограничивая минимальное быстродействие. Недостатком данного подхода является то, что не всегда можно однозначно решить вопрос о выборе того или иного выходного параметра в качестве критерия оптимальности.

13.2.2. Детерминированные критерии оптимальности. Частные критерии являются детерминированными, т. е. критериями, при формировании которых не учитывается степень отклонения выходных параметров от номинальных значений вследствие производственных погрешностей внутренних параметров. К детерминированным критериям относятся и критерии, представляющие собой не

отдельные выходные параметры, а характеристики схемы или зависимости тех или иных выходных параметров от времени, частоты или какого-либо иного внутреннего параметра. В этом случае задача проектирования ставится как задача максимального совпадения одной из характеристик схемы $f(X)$ с теоретической или заданной техническими требованиями характеристикой $f_{тг}(X)$.

Примерами такой задачи является оптимизация амплитудно-частотной характеристики схемы, формулируемая как задача максимального ее приближения к теоретическому (заданному техническими требованиями на схему) виду, или оптимизация параметров модели какого-либо устройства, сводящаяся к выполнению требования максимального совпадения рассчитываемых с помощью модели вольт-амперных характеристик с реальными характеристиками, полученными экспериментально. В таких случаях требуется расчет отклонений двух характеристик в конечном числе узловых точек $\Delta_i = f(X_i) - f_{тг}(X_i)$ и последующая их свертка в сложный критерий [2, 53], подлежащий минимизации.

Обычно свертка производится одним из следующих способов:

$$F = \sum_{i=1}^n C_i |\Delta_i|, F = \sqrt{\sum_{i=1}^n \frac{C_i^2 \Delta_i^2}{n}}, F = \max_i C_i |\Delta_i|, \quad (13.1)$$

где n — число точек, в которых вычисляется отклонение; C_i — весовые коэффициенты, значения которых выбираются тем больше, чем меньше отклонение Δ_i желательно получить в i -й точке.

Несмотря на сложную форму, критерий максимального совпадения характеристик является типичным частным детерминированным критерием, так как оптимизация подвергается лишь один из выходных параметров при учете ограничений на остальные выходные параметры.

13.2.3. Статистические критерии оптимальности. Статистические критерии оптимальности в отличие от детерминированных критериев, зависящих только от номинальных значений внутренних параметров, основаны на использовании статистических характеристик схемы. Одним из примеров такого критерия является процент выхода годных схем: отношение числа схем, удовлетворяющих техническим требованиям, к общему числу изготовленных схем.

Все рассмотренные критерии являются частными критериями и в принципе позволяют получить оптимальное решение поставленной задачи: спроектировать схему таким образом, чтобы выбранный критерий достигал своего максимального (или минимального) значения, а все остальные параметры, как внутренние, так и выходные, не выходили за рамки ограничений, установленных техническими требованиями на схему.

13.2.4. Критерии последовательного принятия решений. Очевидно, что в случае оптимизации схемы по нескольким выходным параметрам одновременно любое принимаемое решение всегда представляет собой компромисс, в котором предпочтение отдается то-

му варианту схемы, который, не являясь наверняка оптимальным ни по одному из выходных параметров, оказывается приемлемым по всей совокупности выходных параметров. Решение о критерии оптимальности можно принимать последовательно по ходу оптимизации, используя получаемую информацию и формируя последовательно изменяемый или уточняемый критерий.

Например, в [53] предложен один из способов нахождения такого компромиссного решения, названный методом последовательных уступок. Метод заключается в следующем. Выходные параметры ранжируются в порядке убывающей важности. Для определенности будем полагать, что каждый из них нужно обратить в максимум. Очевидно, что если критерий требует не максимизации, а минимизации, достаточно изменить его знак на противоположный. Сначала одним из методов оптимизации максимизируется первый по важности выходной параметр $f_1(\mathbf{X})$, затем назначается, более или менее произвольно, «уступка» $\Delta f_1(\mathbf{X})$ в этом выходном параметре, которую мы согласны допустить, чтобы максимизировать следующий выходной параметр. Налагаем на первый выходной параметр $f_1(\mathbf{X})$ ограничение, чтобы он был не меньше $f_{1\text{ макс}}(\mathbf{X}) - \Delta f_1(\mathbf{X})$ и при этом граничном условии максимизируем $f_2(\mathbf{X})$. Снова назначаем «уступку» $\Delta f_2(\mathbf{X})$ теперь уже для выходного параметра $f_2(\mathbf{X})$, за счет чего обращаем в максимум следующий выходной параметр $f_3(\mathbf{X})$ и т. д.

Например, сначала минимизируется потребляемая схемой мощность $P = f_1(x_1, x_2, \dots, x_m)$ и находится значение $P_{\text{мин}}$, затем на мощность P накладывается ограничение $P \leq P_{\text{мин}} + \Delta P$, где ΔP — выбранная уступка, и затем минимизируется время задержки сигнала $t_3 = f_2(x_1, x_2, \dots, x_m)$ уже с учетом этого ограничения.

Такой способ последовательного принятия компромиссного решения удобен тем, что здесь мы всегда видим, ценой какой уступки в одном выходном параметре приобретается выигрыш в другом. Правда, невозможно заранее предсказать, какую область для поиска предоставит назначенная уступка, поэтому назначение уступок в каждом конкретном случае весьма затруднительно. Можно только предполагать, что ценой даже незначительных уступок приобретается довольно существенная свобода выбора последующего решения благодаря тому, что обычно в области экстремума выходные параметры изменяются не резко, а монотонно. Существенным недостатком метода последовательного принятия компромиссного решения является также значительный рост времени оптимизации, так как фактически приходится повторять процесс поиска последовательно n раз, где n — число выходных параметров схемы. По этой причине такие методы редко применяются для оптимизации электронных схем.

13.2.5. Обобщенные критерии оптимальности. Другим подходом к построению единого критерия оптимальности для задачи векторной оптимизации является свертка векторного критерия в скалярный. Свертка такого рода может производиться одним из двух следующих способов [2, 54].

Первый способ состоит в нормировании всех выходных параметров схемы с помощью весовых коэффициентов и последующем объединении их в обобщенный формальный скалярный критерий оптимальности, который затем и подвергается максимизации или минимизации. Второй способ сводится к построению минимаксно-го обобщенного критерия оптимальности, заключающегося в использовании на каждом шаге оптимизации максимального отклонения какого-либо из выходных параметров от своего оптимального (или номинального) значения и решении задачи минимизации данного максимального отклонения. И та и другая постановка задачи будут подробно рассмотрены в последующих параграфах.

13.2.6. Диалоговые критерии оптимальности. Принципиально отличающийся подход к решению задачи векторной оптимизации развивается в [55]. Этот подход основан на оперативном взаимодействии разработчика схемы с ЭВМ посредством диалогового режима работы. В данном методе на ЭВМ возложен процесс вычисления математической модели схемы на каждом шаге оптимизации, а человек осуществляет принятие решения и построение процедуры принятия этого решения. Как правило, этот процесс заключается в поиске множества точек области Парето, т. е. области неулучшаемых решений, в которой ни один из выходных параметров не может быть улучшен без одновременного ухудшения хотя бы одного из остальных выходных параметров. После нахождения данного множества проектировщик схемы на основе дополнительных неформализованных или интуитивных соображений принимает окончательное решение о выборе оптимального, с его точки зрения, варианта схемы.

Нужно, однако, иметь в виду, что при большом числе выходных и внутренних параметров схемы такой процесс принятия ре-

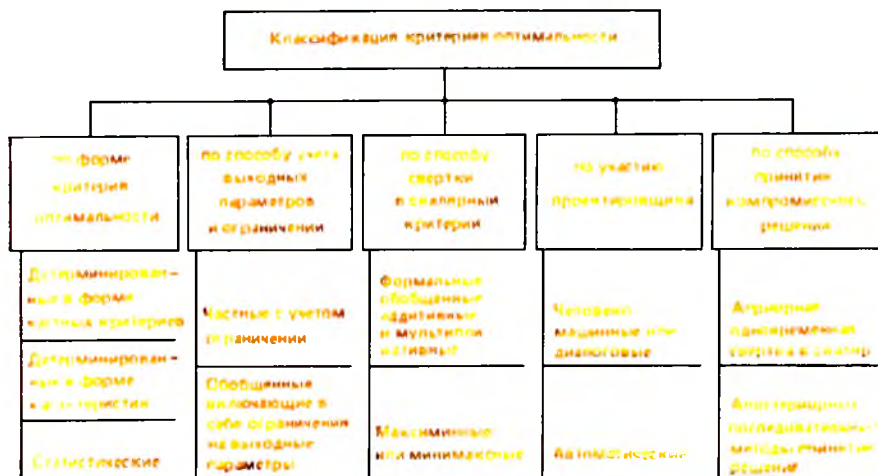


Рис. 13.1. Классификация критериев оптимальности

шения разработчиком существенно затруднен и не гарантирует ему выбора оптимального варианта, так как перебор точек компромисса, даже в какой-то степени целенаправленный, может не привести к перебору всех значений области Парето. Кроме того, так как вычислительные затраты на оптимизацию на один-два порядка превышают затраты на расчет схемы, работа проектировщика в реальном времени может быть трудновыполнимой.

13.2.7. Классификация критериев оптимальности. Классификация рассмотренных критериев оптимальности представлена на рис. 13.1. Классификация может производиться по форме критерия оптимальности, по полноте учета всех выходных параметров в критерии оптимальности, по способу свертки векторного критерия в скалярный обобщенный критерий, по участию проектировщика схемы в процессе формирования критерия, и, наконец, по способу принятия компромиссного решения. Последний способ классификации включает критерии, основанные на одновременной, произведенной до начала оптимизации, свертке вектора выходных параметров в скаляр, и критерии, основанные на последовательных методах принятия решений, использующих информацию о процессе поиска. К таким критериям относятся, например, критерии, формирующиеся в процессе диалога, минимаксные критерии и критерии, основанные на методе последовательных уступок.

13.3. ФОРМАЛЬНЫЕ ОБОБЩЕННЫЕ КРИТЕРИИ ОПТИМАЛЬНОСТИ

13.3.1. Нормирование выходных параметров. При формировании формальных обобщенных критериев возникает задача преобразования всех разноразмерных выходных параметров в безразмерные величины, которые уже можно сравнивать между собой и объединять в скалярные критерии или свертки. Такое преобразование достигается путем масштабирования или нормирования выходных параметров. При этом назначение коэффициентов нормирования или весовых коэффициентов для некоторых критериев оптимальности определяет степень влияния каждого отдельного выходного параметра на целевую функцию. В этом случае процесс нормирования по своей сути представляет собой ранжирование выходных параметров, т. е. их упорядочение по степени важности.

13.3.2. Учет ограничений. Следующей важной задачей, которую необходимо решить при формировании обобщенного критерия, является задача учета технических требований на схему и ограничений, накладываемых при ее проектировании и изготовлении, на внутренние и выходные параметры.

Ограничения, накладываемые на внутренние параметры схемы (параметры транзисторов, сопротивления, емкости и т. п.), называются параметрическими и образуют допустимую область изменения внутренних параметров D_x , которой должен принадлежать

вектор внутренних параметров X . Они, как правило, носят характер либо численных равенств

$$x_i = c, \quad (13.2)$$

либо численных неравенств

$$a \leq x_i \leq b. \quad (13.3)$$

Учет ограничений производится методами условной оптимизации, которые в отличие от методов безусловной оптимизации, предназначены специально для оптимизации при наличии ограничений. К методам условной оптимизации относятся методы возможных направлений, методы проекции градиентов и другие методы, более подробно рассмотренные в следующей главе.

Ограничения на выходные параметры $G(X)$ определяют область допустимых решений или область работоспособности и подразделяются следующим образом.

Ограничения на выходные параметры $g_i(X)$, не входящие в критерий оптимальности, называются *функциональными*. Они обычно учитываются в соответствующих алгоритмах оптимизации, однако их можно также учитывать при формировании критерия оптимальности $F(X)$ путем добавления (если имеется в виду минимизация критерия $F(X)$) дополнительной функции $Q(g(X))$, обозначаемой ниже $Q(X)$. Тогда

$$F(X) = F(X) + Q(X). \quad (13.4)$$

Дополнительная функция $Q(X)$ может строиться различным образом, например, резко возрастать, если в результате нарушения ограничения рабочая точка выходит за пределы области допустимых решений или области работоспособности. Тогда добавление $Q(X)$ к критерию $F(X)$ носит характер штрафа и функция $Q(X)$ называется штрафной. Функция $Q(X)$ может также препятствовать выходу рабочей точки за пределы области работоспособности и увеличивать значение критерия $F(X)$ при приближении к границе области работоспособности, создавая имитацию определенного барьера. В этом случае функция $Q(X)$ называется барьерной.

Ограничения на выходные параметры $f_i(X)$, вошедшие в критерий оптимальности $F(X)$, обычно называются *критериальными* и учитываются уже при формировании самого критерия оптимальности $F(X)$ одним из способов, описанных ниже. Эти ограничения на управляемые выходные параметры могут иметь вид неравенства

$$f_i^+(X) \geq TT_i, \quad (13.5)$$

где $f_i^+(X)$ — i -й выходной параметр, для которого желательно максимальное увеличение, TT_i — значение технического требования, предъявляемого к i -му выходному параметру; или неравенства

$$f_i^-(X) \leq TT_i, \quad (13.6)$$

если выходной параметр $f_i^-(X)$ требуется минимизировать. Ограничения могут также иметь вид двусторонних неравенств.

Как правило, все обобщенные критерии относятся к критериям либо аддитивного, либо мультипликативного типа.

13.3.3. Обобщенные критерии аддитивного типа. В аддитивных критериях целевая функция формируется в результате сложения нормированных выходных параметров с учетом или без учета ограничений на них. Например, критерий максимальной суммы нормированных выходных параметров имеет вид

$$F(X) = \sum_{i=1}^m a_i f_i(X), \quad (13.7)$$

где m — число учитываемых критерием выходных параметров, а весовые коэффициенты a_i имеют положительный знак, если выходной параметр $f_i^+(X)$ необходимо максимизировать, и отрицательный знак, если $f_i^-(X)$ должен подвергаться минимизации. Если в (13.7) должны учитываться ограничения типа (13.5) или (13.6), то данный критерий преобразуется в критерий

$$F(X) = \sum_{i=1}^m a_i (f_i(X) - TT_i). \quad (13.8)$$

Предполагается, что критерии (13.7) и (13.8) представляют собой целевую функцию, подлежащую в дальнейшем максимизации.

Более удобным и поэтому чаще употребляемым является аддитивный критерий, представляющий собой сумму квадратов нормированных отклонений выходных параметров $f_i(X)$ от технических требований на них TT_i :

$$F(X) = \sum_{i=1}^m a_i (f_i(X) - TT_i)^2. \quad (13.9)$$

Удобство данного критерия заключается в том, что для его оптимизации можно применять методы квадратичного или выпуклого программирования, что существенно облегчает поиск оптимума целевой функции.

13.3.4. Обобщенные критерии мультипликативного типа. Мультипликативные критерии обычно представляются в виде отношения произведений нормированных выходных параметров:

$$F(X) = \frac{\prod_{i=1}^m a_i f_i^+(X)}{\prod_{j=1}^l a_j f_j^-(X)}, \quad (13.10)$$

где в числителе перемножаются все выходные параметры, требующие максимизации и имеющие ограничения вида (13.5), а в знаменателе — все выходные параметры, требующие минимизации и имеющие ограничения вида (13.6). Целевая функция (13.10) в дальнейшем подвергается максимизации.

В качестве примера использования мультипликативного критерия вида (13.10) можно привести критерий, формируемый для оптимизации логических переключательных схем:

$$F(X) = a_1 \Delta U_{\Pi} / (a_2 t_3 a_3 P), \quad (13.11)$$

где ΔU_n — запас помехоустойчивости схемы; t_3 — время задержки сигнала на выходе схемы; P — рассеиваемая схемой мощность.

13.3.5. Выбор весовых коэффициентов. Трудность формирования приведенных обобщенных критериев заключается в выборе соответствующих весовых коэффициентов a_i для каждого из выходных параметров $f_i(X)$. Такое нормирование не требуется только в частном случае формирования критерия из выходных параметров, имеющих одинаковую размерность, например, когда под выходными параметрами понимаются ординаты частотной или переходной характеристики в отдельных точках оси частот или времени либо амплитуды сигнала в различных узлах схемы. Во всех остальных случаях можно рекомендовать использовать в описываемых критериях в качестве $f_i(X)$ не абсолютные, а относительные значения выходных параметров, взятые по отношению к своим предельно допустимым значениям.

Весовые коэффициенты a_i характеризуют во всех данных критериях важность соответствующих выходных параметров $f_i(X)$ для конкретного проектировщика схемы. Чтобы уменьшить субъективный характер выбора a_i , удобно определять их в зависимости от разницы между конкретным значением выходного параметра и его предельно допустимым значением (13.7). В этом случае при ужесточении требований на какой-либо выходной параметр, например уменьшении времени t_3 в (13.11), автоматически будет возрастать вес этого параметра по сравнению с остальными.

13.3.6. Преимущества и недостатки формальных обобщенных критериев. Из общности всех рассмотренных в данном параграфе критериев следуют и их общие основные достоинства и недостатки. Преимуществом критериев такого рода является возможность учета в качестве $f_i(X)$ любых выходных параметров схемы, а также надежности схемы, ее стоимости и т. д.

Основным недостатком критериев такого типа является то, что здесь в принципе возможно компенсировать ухудшение целевой функции из-за ухудшения одного выходного параметра за счет улучшения какого-либо другого выходного параметра, так как в процессе оптимизации нельзя контролировать и регулировать вклад каждого выходного параметра в общий критерий оптимальности.

13.4. МАКСИМИННЫЕ (МИНИМАКСНЫЕ) КРИТЕРИИ ОПТИМАЛЬНОСТИ

13.4.1. Общий вид критериев. Критерии данного вида возникли в результате желания проектировщиков на каждом шаге оптимизации контролировать все выходные параметры, выбирать из них наихудший и именно его улучшать до тех пор, пока какой-либо следующий выходной параметр не займет его место и не станет наихудшим. Для этой цели используется критерий вида

$$F(X) = \min_{i \in [1, m]} [a_i f_i(X)], \quad (13.12)$$

требующий дальнейшей максимизации и называющийся максиминным, и критерий вида

$$F(X) = \max_{i \in [1, m]} [a_i f_i(X)], \quad (13.13)$$

требующий дальнейшей минимизации и называющийся, соответственно, минимаксным. Здесь m — число выходных параметров, подвергающихся оптимизации.

13.4.2. Пример использования критериев. С помощью этих критериев могут решаться многие задачи проектирования схем, например задача поддержания максимально возможных напряжений на всех выходах логической переключающей схемы. При использовании максиминного критерия оптимальности вида (13.12) влияние на целевую функцию будет оказывать только напряжение того из выходов, на котором оно будет минимальным, т. е. наихудшим, и оно в результате оптимизации будет повышаться до тех пор, пока минимальным не станет напряжение на другом выходе и т. д. Соответствующим образом и минимаксный критерий вида (13.13) может использоваться при проектировании схемы, имеющей минимально возможные задержки сигнала на всех выходах.

В рассмотренных примерах очевидно преимущество применения максиминных (минимаксных) критериев перед формальными обобщенными критериями аддитивного или мультипликативного типа, при использовании которых можно достигать оптимума для такой совокупности внутренних параметров X , для которой, допустим, задержка сигнала на одном из выходов была бы существенно меньше, чем на другом. Благодаря отсутствию этого недостатка максиминные критерии все более широко применяются в настоящее время [2].

13.4.3. Учет ограничений. При представлении условий работоспособности в виде неравенств (13.5) или (13.6) максиминный критерий приводится к виду

$$F(X) = \min_{i \in [1, m]} [a_i (f_i(X) - TT_i)]. \quad (13.14)$$

В данном случае может использоваться и минимаксный критерий, если желательна минимизация максимального отклонения выходных параметров от заданных технических требований.

13.4.4. Учет статистических распределений выходных параметров. При использовании максиминных (минимаксных) критериев легко производить учет статистических распределений выходных параметров. Например, широко используется максиминный критерий, называемый запасом работоспособности [2] и имеющий вид

$$F(X) = \min_{i \in [1, m]} \left[a_i \left(\frac{f_i(X) - TT_i}{\delta_i} \right) \right], \quad (13.15)$$

где δ_i — величина, характеризующая рассеяние i -го выходного параметра. В этом случае задача расчета вектора внутренних параметров X ставится как экстремальная задача, при решении которой на каждом шаге максимизируется минимальный запас работоспособности по i -му выходному параметру.

Максимизация $F(X)$ при вычислении δ_i путем статистического анализа схемы практически приводит к достижению максималь-

ной вероятности выполнения заданных технических требований. Однако учитывая большие вычислительные затраты для статистического анализа, рекомендуется задавать δ_i в качестве исходных данных на основе априорных представлений о разбросе выходных параметров. Тогда максиминный критерий становится детерминированным, а величины δ_i выступают в роли весовых коэффициентов. После оптимизации и получения решения можно при необходимости с помощью однократного выполнения статистического анализа оценить погрешности предварительного задания δ_i и повторить процедуру оптимизации со скорректированными значениями δ_i .

13.4.5. Выбор весовых коэффициентов. При формировании максиминных (минимаксных) критериев любого вида большое значение приобретает выбор численных значений весовых коэффициентов a_i .

Так как в ряде случаев требуется максимизация степени выполнения заданных условий работоспособности не для всех выходных параметров, то выходные параметры можно подразделять на две группы:

параметры, для которых желательна максимальная степень выполнения условий работоспособности;

параметры, для которых достаточно получения любых значений запасов работоспособности, в том числе близких к нулю.

При таком подразделении удастся добиться большей степени выполнения условий работоспособности для параметров первой группы. Для параметров первой группы принимается $a_i = 1$, а для параметров второй группы $a_i > 1$ (практически достаточно $a_i = 10 \dots 20$).

13.4.6. Преимущества и недостатки максиминных (минимаксных) критериев. К преимуществам максиминных критериев, как уже отмечалось выше, относится то, что в ходе оптимизации контролируются все выходные параметры и улучшается тот из них, который имеет наихудшее значение, что дает возможность применять их в случаях, когда невозможно применение обобщенных формальных критериев. Кроме этого, максиминные критерии в зависимости от конкретных условий оптимизации (наличия ресурсов машинного времени, сведений о статистическом разбросе параметров и т. д.) можно использовать как в качестве детерминированных, так и статистических, что позволяет решить более широкий круг задач проектирования.

К недостаткам максиминных критериев, так же как и аддитивных и мультипликативных критериев, следует отнести произвольность выбора весовых коэффициентов a_i . Но для максиминных критериев этот недостаток более существен по той причине, что значения a_i оказывают значительное влияние не только на результаты оптимизации, но и на ход процесса оптимизации, так как минимальный выходной параметр определяется выбором соответствующих коэффициентов a_i . Существенным недостатком критериев этого рода является также то, что целевая функция, формирующая

яся на каждом шаге оптимизации как минимальная из набора функций, как правило, имеет сложный рельеф со множеством глубоких гребней (или оврагов), что накладывает жесткие требования на используемые методы оптимизации.

13.5. СТАТИСТИЧЕСКИЕ КРИТЕРИИ

13.5.1. Критерий процента выхода годных схем. Использование статистических критериев при оптимизации ставит своей целью получение максимальной вероятности безотказной работы схемы. Наиболее простым и распространенным статистическим критерием оптимальности является процент выхода годных схем $P(X)$ или отношение числа годных схем к общему числу изготовленных схем. Тогда процесс оптимизации сводится к максимизации целевой функции $P(X)$:

$$\max_{X \in D_X} P(X), \quad (13.16)$$

где D_X — область допустимого изменения внутренних параметров.

Существенным недостатком такого подхода является необходимость расчета $P(X)$ на каждом шаге оптимизации, что приводит к чрезмерному увеличению вычислительных затрат на поиск максимума.

13.5.2. Критерий надежности схемы. Проектирование схем с учетом критериев (13.15) и (13.16) позволяет получить максимальный процент выхода годных схем в момент изготовления, однако в данных критериях не учитывается технический ресурс схемы или срок ее службы, т. е. время, в течение которого рабочая точка должна находиться внутри области работоспособности. Очевидно, что чем дальше будет находиться рабочая точка от границ области работоспособности и чем медленнее она будет приближаться к границам области, тем больше технический ресурс и надежнее схема.

Для того чтобы оптимизировать схему по критерию надежности, необходимо знать помимо статистических характеристик внутренних параметров схемы законы их изменения от времени, температуры, влажности и других факторов старения. Критерий надежности работы схемы, таким образом, характеризует вероятность того, что в течение определенного времени выходные параметры схемы не выйдут за границы области работоспособности, и может формулироваться следующим образом [1]:

$$F(X) = \min_{i \in [1, m]} t_i, \quad (13.17)$$

где t_i — технический ресурс i -го параметра, т. е. время, в течение которого выходной параметр $f_i(X)$ находится в области работоспособности:

$$t_i = (f_i(X) - TT_i) / \Delta f_i, \quad (13.18)$$

где Δf_i — изменение i -го выходного параметра $f_i(X)$ за единицу времени.

13.5.3. Недостатки статистических критериев. Основным недостатком, присущим всем статистическим критериям оптимальности, являются большие вычислительные затраты на поиск оптимальной схемы, поскольку, во-первых, сами критерии требуют для своего формирования вычисления статистических характеристик, а во-вторых, оптимизация по статистическим критериям требует применения специальных методов поиска, так как целевые функции, получающиеся в данном случае, имеют сложный негладкий рельеф с наличием многих (или, по крайней мере, нескольких) экстремумов.

Пример 13.2. Формирование критериев оптимальности различного типа.

Исследуемая схема. Используя схему транзисторно-транзисторной логики (ТТЛ), изображенную на рис. 13.2, разберем пример формирования критериев

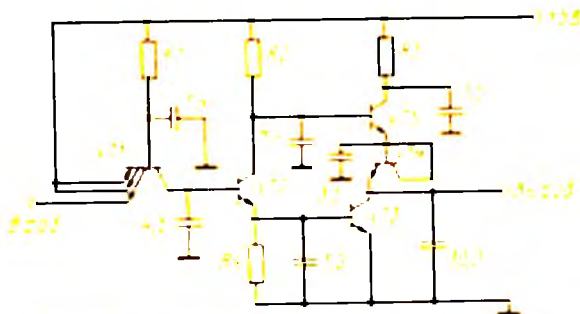


Рис. 13.2 Принципиальная схема ячеек ТТЛ

оптимальности различного типа. Предположим, что в качестве выходных параметров схемы исследуются потребляемая схемой в статике мощность P и время задержки сигнала t_z . Оптимизация этих двух частных критериев проводится по внутреннему параметру — сопротивлению R_1 . Полученные в результате исследования зависимости частных критериев оптимальности — потребляемой мощности и времени задержки сигнала от сопротивления R_1 представлены на рис. 13.3, а и б соответственно.

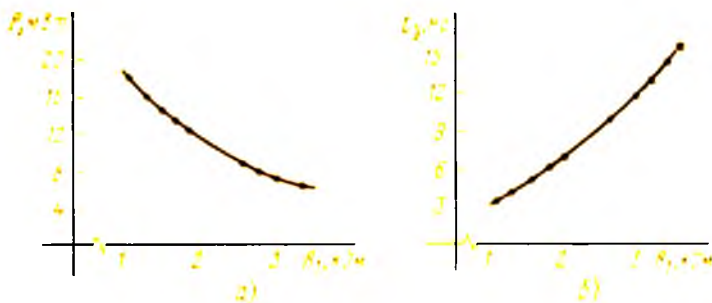


Рис. 13.3. Зависимости потребляемой схемой мощности (а) и времени задержки сигнала (б) от сопротивления R_1

Формирование формальных обобщенных критериев оптимальности. Критерий оптимальности аддитивного типа (13.7), сформированный как

$$F_A(R_1) = \sum_{i=1}^2 f_i(R_1) = a_1 P + a_2 t_3, \quad (13.19)$$

имеет зависимость от R_1 , изображенную на рис. 13.4. Здесь полагается $a_1 = a_2 = -1$, и целевая функция подвергается в дальнейшем максимизации.

Критерий оптимальности мультипликативного типа (13.10), сформированный как

$$F_{\Pi}(R_1) = \left(\prod_{i=1}^2 f_i(R_1) \right)^{-1} = (a_1 P a_2 t_3)^{-1}, \quad (13.20)$$

имеет зависимость от R_1 , изображенную на рис. 13.5. Здесь полагается $a_1 = a_2 = 1/10$.

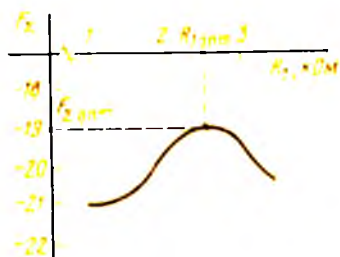


Рис. 13.4. Зависимость аддитивного критерия оптимальности от сопротивления R_1

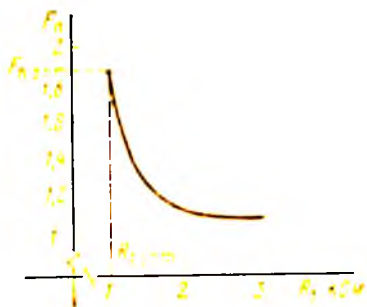


Рис. 13.5. Зависимость мультипликативного критерия оптимальности от сопротивления R_1

Из рис. 13.4 и 13.5 видно, что значения оптимального сопротивления R_1 для аддитивного и мультипликативного критериев значительно отличаются друг от друга.

Формирование минимаксного критерия оптимальности. Согласно (13.15) целевую функцию можно формировать как максимальную из двух частных критериев на каждом шаге оптимизации:

$$F_{\max}(R_1) = \max_{i \in [1, 2]} [(f_i(R_1) - TT_i) / \delta_i]; \quad (13.21)$$

в дальнейшем данный критерий оптимальности должен подвергаться минимизации. Полученная зависимость минимаксного критерия от сопротивления R_1 изображена на рис. 13.6. Здесь полагается $TT_1 = TT_2 = 20$ — верхние ограничения на потребляемую мощность и время задержки сигнала.

Из рис. 13.3—13.6 видно, что оптимальное значение внутреннего парамет-

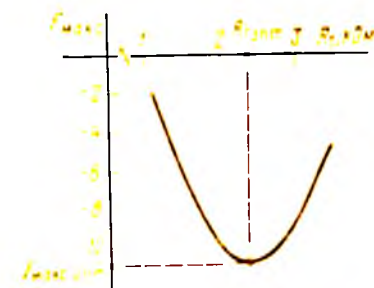


Рис. 13.6. Зависимость минимаксного критерия оптимальности от сопротивления R_1

ра схемы, в данном случае сопротивления R_1 , определяется выбранным критерием оптимальности, т. е. теми целями, которые ставит перед собой проектировщик схемы.

ГЛАВА 14

МЕТОДЫ НЕПРЕРЫВНОЙ ПАРАМЕТРИЧЕСКОЙ ОПТИМИЗАЦИИ

14.1. ПОСТАНОВКА ЗАДАЧИ

14.1.1. Классификация задач оптимизации параметров РЭА. Параметрическая оптимизация РЭА — это процесс определения значений параметров входящих в схему элементов (внутренних параметров X), при которых достигаются заданные или экстремальные значения ее выходных параметров Y или целевой функции. При этом структура схемы, т. е. типы входящих в нее элементов и способ их соединения, предполагаются известными. Также известны диапазоны изменения параметров элементов схемы. При помощи методов оптимизации решаются такие задачи проектирования, которые либо совсем невозможно решить методами расчета схем, либо возможно решить лишь приближенно. Задачи проектирования, требующие применения методов оптимизации, можно разбить на четыре основные группы.

Первая группа — задачи поиска оптимальных параметров схем, обеспечивающих экстремальные значения целевой функции, например минимальный коэффициент нелинейных искажений или максимальное быстродействие схемы. Для задач данной группы, во-первых, заранее неизвестно искомое численное «наилучшее» значение целевой функции, а во-вторых, существуют ограничения на все или большинство внутренних параметров схемы.

Ко второй группе относятся задачи приближения расчетных характеристик к характеристикам, заданным по точкам (обычно полученным экспериментально). Это, в частности, задачи идентификации параметров моделей.

В третью группу входят задачи адаптации известных схем к новым техническим требованиям — другому частотному или температурному диапазону, новой элементной базе.

Четвертую группу составляют задачи уточнения параметров схем, найденных с помощью простого инженерного расчета или в результате использования методов синтеза.

В настоящий момент практически все эти задачи проектирования решаются при помощи многовариантного расчета схемы. Разработчик оценивает результаты каждой серии расчетов и назначает следующую серию расчетов до тех пор, пока не будут получены приемлемые результаты. Если целевая функция схемы $F(X)$ и ее зависимость от внутренних параметров X задана одним из способов, описанных в гл. 13, то представляется возможность

автоматизировать процесс решения задач проектирования такого рода. Тогда каждая из задач перечисленных четырех групп сводится к нахождению такого вектора внутренних параметров $\mathbf{X}^*$, который бы обеспечивал максимальное или минимальное значение целевой функции $F(\mathbf{X}^*)$ и удовлетворял бы всем заданным ограничениям. В самом общем случае задачи оптимизации такого рода формулируются в терминах общей задачи нелинейного программирования.

14.1.2. Общая задача нелинейного программирования. Под общей задачей нелинейного программирования понимается задача нахождения $\max F(\mathbf{X})$ или $\min F(\mathbf{X})$, т. е. максимизации или минимизации скалярной функции $F(\mathbf{X})$ при условиях

$$\mathbf{X} = (x_1, x_2, \dots, x_n) \in D_{\mathbf{X}}, \quad (14.1)$$

и

$$\begin{aligned} g_1(x_1, x_2, \dots, x_n) &\leq 0, \\ g_2(x_1, x_2, \dots, x_n) &\leq 0, \\ &\vdots \\ g_m(x_1, x_2, \dots, x_n) &\leq 0, \end{aligned} \quad (14.2)$$

где $g_i(\mathbf{X})$ — функциональные и критериальные ограничения; $D_{\mathbf{X}}$ — некоторая область n -мерного евклидова пространства R^n , в пределах которой допустимо варьирование вектора внутренних параметров $\mathbf{X}$. Область, определяемая совместно условиями (14.1) и (14.2), обозначается $G(\mathbf{X})$ и называется областью допустимых решений или областью работоспособности; $F(\mathbf{X})$ — целевая функция.

При решении задач оптимизации схем методами нелинейного программирования обычно переменные $x_1, \dots, x_n$ представляют собой варьируемые параметры схемы — сопротивления R , емкости C , параметры активных элементов, а целевой функцией является какой-либо выходной параметр схемы — помехоустойчивость, время задержки сигнала и т. д., либо совокупность выходных параметров, объединенная в обобщенный критерий оптимальности, ограничениями $g_i(\mathbf{X})$ могут быть технические требования на схему или условия ее работоспособности.

Знак неравенства в (14.2) может быть изменен на противоположный умножением (14.2) на -1 , что не изменяет математическую постановку задачи.

В общем случае при проектировании схем целевая функция $F(\mathbf{X})$ и ограничения (14.2) являются нелинейными функциями вектора управляемых параметров $\mathbf{X}$, и задача решается методами нелинейного математического программирования, хотя в отдельных случаях удается линеаризовать целевую функцию и ограничения и свести задачу оптимизации к задаче линейного программирования. В данной главе будут рассмотрены наиболее общие методы нелинейного программирования.

14.1.3. Основные определения. Вектор $\mathbf{X}^* = [x_1^*, \dots, x_n^*]^T$, удовлетворяющий соотношениям (14.1), (14.2) и максимизирующий (минимизирующий) целевую функцию $F(\mathbf{X})$, называется оптимальной или экстремальной точкой, а соответствующее ему значение

$F(\mathbf{X}^*)$ — оптимальным значением (максимумом или минимумом) целевой функции.

Множество точек, для которых целевая функция двух переменных имеет постоянное значение, называется линией уровня функции $F(\mathbf{X})$. Если целевая функция непрерывна и дифференцируема, то существует градиент функции $F(\mathbf{X})$, определяемый как вектор-столбец из первых частных производных $F(\mathbf{X})$ по $\mathbf{X}$ в данной точке $\mathbf{X}^{(k)}$:

$$\nabla F(\mathbf{X}^{(k)}) = \left[\frac{\partial F(\mathbf{X}^{(k)})}{\partial x_1} \dots \frac{\partial F(\mathbf{X}^{(k)})}{\partial x_n} \right]^T. \quad (14.3)$$

Известно, что градиент функции направлен в сторону наискорейшего увеличения функции, т. е. наискорейшего подъема, и что он ортогонален линии уровня $F(x)$, проходящей через данную точку $\mathbf{X}^{(k)}$. Вектор, противоположный градиенту, направлен в сторону наискорейшего спуска.

Методы нелинейного программирования основаны на аппроксимации функции $F(\mathbf{X})$ с помощью ряда Тейлора в окрестности точки $\mathbf{X}^{(k)}$:

$$F(\mathbf{X}) \approx F(\mathbf{X}^{(k)}) + \nabla^T F(\mathbf{X}^{(k)}) (\mathbf{X} - \mathbf{X}^{(k)}) + 0,5 (\mathbf{X} - \mathbf{X}^{(k)})^T \nabla^2 F(\mathbf{X}^{(k)}) (\mathbf{X} - \mathbf{X}^{(k)}), \quad (14.4)$$

где $\nabla^2 F(\mathbf{X}^{(k)}) = \mathbf{H}(\mathbf{X}^{(k)})$ представляет собой квадратичную матрицу вторых частных производных целевой функции $F(\mathbf{X})$, взятых в точке $\mathbf{X}^{(k)}$, так называемую матрицу Гессе

$$\mathbf{H}(\mathbf{X}^{(k)}) = \begin{bmatrix} \frac{\partial^2 F(\mathbf{X}^{(k)})}{\partial x_1^2} & \frac{\partial^2 F(\mathbf{X}^{(k)})}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 F(\mathbf{X}^{(k)})}{\partial x_1 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 F(\mathbf{X}^{(k)})}{\partial x_n \partial x_1} & \frac{\partial^2 F(\mathbf{X}^{(k)})}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 F(\mathbf{X}^{(k)})}{\partial x_n^2} \end{bmatrix}. \quad (14.5)$$

В зависимости от числа используемых членов в разложении (14.4) различают методы оптимизации нулевого, первого и второго порядка.

14.1.4. Необходимые и достаточные условия существования экстремума для задачи без ограничений. Необходимыми и достаточными условиями того, что точка $\mathbf{X}^*$ является точкой минимума функции $F(\mathbf{X})$ для задачи без ограничений являются:

- 1) функция $F(\mathbf{X})$ дифференцируема в точке $\mathbf{X}^*$;
- 2) $\nabla F(\mathbf{X}^*) = 0$, т. е. точка $\mathbf{X}^*$ является стационарной точкой;
- 3) $\mathbf{H}(\mathbf{X}^*) = \nabla^2 F(\mathbf{X}^*) > 0$.

Неравенство следует понимать как положительность всех собственных значений матрицы Гессе, т. е. матрица Гессе положительно определенная.

Для существования максимума в точке $\mathbf{X}^*$ матрица Гессе для $F(\mathbf{X}^*)$ должна быть отрицательно определенной.

Классические методы оптимизации основаны на аналитическом представлении целевой функции $F(\mathbf{X})$ в виде формулы, которую

дифференцируют по варьируемым параметрам X , после чего производные приравнивают нулю и решают получившиеся уравнения. Тем самым проверяют выполнение необходимых условий 1) и 2) существования экстремума и находят стационарную точку. Далее проверяется условие 3) и если оно выполняется, то найденная точка является соответствующим решением. Однако при проектировании схем целевые функции $F(X)$ представляются обычно не аналитическими выражениями, а численными результатами расчетов математических моделей схемы (ММС), поэтому применение классических методов оптимизации становится невозможным и вместо них используются итеративные численные методы поиска экстремума X^* , основанные на движении из выбранной начальной точки X^0 шагами $\Delta X^{(k)}$ в направлении, определяемом алгоритмом оптимизации.

14.1.5. Одноэкстремальные и многоэкстремальные целевые функции. Целевая функция $F(X)$ называется одноэкстремальной, если она имеет одну точку, удовлетворяющую условиям 1) — 3) и многоэкстремальной, если она имеет несколько таких точек.

Глобальным экстремумом называется точка, в которой минимизируемая (максимизируемая) целевая функция имеет наименьшее (наибольшее) значение среди всех остальных экстремумов, носящих название локальных экстремумов.

Методы оптимизации, основанные на предположении одноэкстремальности целевой функции $F(X)$, называются методами локального поиска, а методы, основанные на предположении многоэкстремальности $F(X)$ и ставящие своей целью нахождение глобального экстремума, — методами глобального поиска. Предсказать одноэкстремальность функции $F(X)$ можно только в том случае, если заранее известно, что $F(X)$ — выпукла (при минимизации) или вогнута (при максимизации) во всей области допустимых значений X .

Однако при проектировании РЭА вследствие алгоритмического способа задания и расчета целевых функций установить их выпуклость или вогнутость и предсказать одноэкстремальность невозможно. Это можно выяснить лишь численным исследованием $F(X)$ во всей допустимой области, если же такое исследование слишком сложно, необходимо применять методы глобального поиска.

Кроме того, даже одноэкстремальные целевые функции, получающиеся при оптимизации параметров схем с помощью обобщенных критериев оптимальности, имеют из-за этого, как правило, сложный рельеф с наличием узких изогнутых «оврагов» («гребней»), на дне (вершине) которых расположен минимум (максимум). Определение экстремума функций такого рода с помощью большинства методов оптимизации приводит к зигзагообразному движению вдоль направления поиска, что существенно замедляет сходимость к точке экстремума и увеличивает потери на поиск.

С помощью некоторых методов экстремум такого рода вообще не обнаруживается и поиск прекращается вдали от него. Таким образом, для задач проектирования необходимо использовать

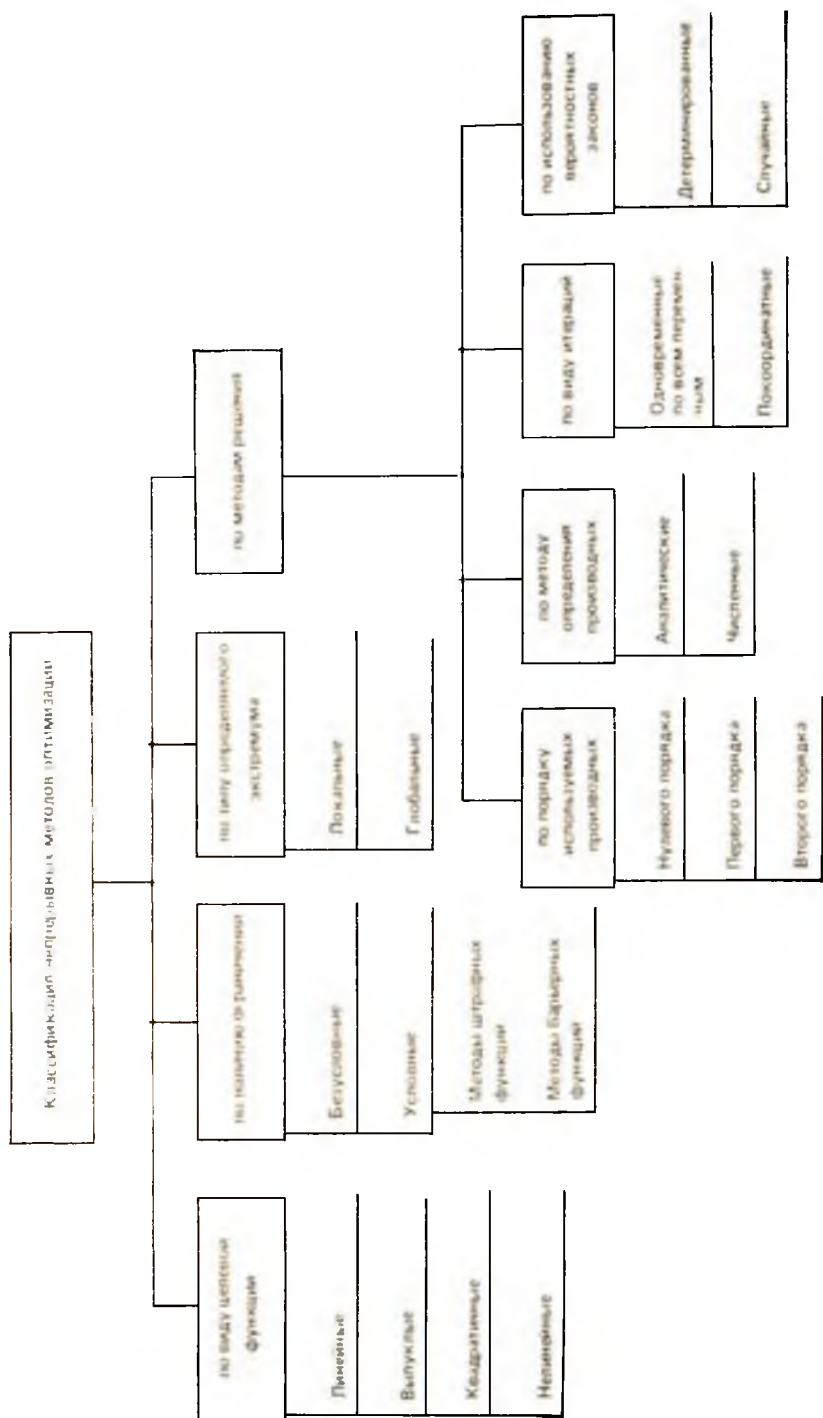


Рис. 14.1. Классификация непрерывных методов оптимизации

те методы оптимизации, которые приспособлены для поиска экстремума в овражных и многоэкстремальных ситуациях.

14.1.6. Безусловные и условные экстремумы. В том случае, если в задаче оптимизации отсутствуют ограничения (14.1), (14.2), т. е. экстремум ищется в пределах неограниченного пространства варьируемых внутренних параметров X , задача носит название безусловной оптимизации, а найденный экстремум — безусловного экстремума. Наличие ограничений приводит к задаче условной оптимизации, а решением ее является условный экстремум.

Задача условной оптимизации существенно более сложна, чем задача оптимизации без ограничений и сводится к ней при помощи методов, основанных на методе множителей Лагранжа [58]. Условия существования экстремума для задач условной оптимизации определяются теоремой Куна — Таккера [57].

14.1.7. Классификация непрерывных методов оптимизации. Ни один из существующих непрерывных методов оптимизации или методов нелинейного программирования не оказывается существенно лучше остальных для целевых функций любого вида. Выбор того или иного метода определяется спецификой каждой конкретной задачи. На рис. 14.1 приведена классификация методов оптимизации, применяемых для задач проектирования РЭУ.

14.2. МЕТОДЫ ПОИСКА ЛОКАЛЬНОГО ЭКСТРЕМУМА

14.2.1. Методы оптимизации, не использующие производных (методы нулевого порядка). Эти методы обычно называются методами поиска или методами нулевого порядка. В типичном методе поиска направление минимизации полностью определяется на основании последовательных вычислений целевой функции $F(X)$.

Как правило, при решении задач безусловной оптимизации для достаточно гладких выпуклых или вогнутых функций $F(X)$ методы, использующие первые и вторые производные, сходятся быстрее, чем методы нулевого порядка. Однако при оптимизации схем в условиях сложного непредсказуемого рельефа целевых функций $F(X)$, их алгоритмического, неявного способа задания, например посредством решения системы дифференциальных уравнений, а также при сложной форме ограничений использование методов нулевого порядка часто предпочтительнее. Объясняется это тем, что в указанных условиях приходится часто менять направление поиска, поэтому затраты времени на поиск направления, т. е. расчет производных, оказываются слишком большими, а движение по направлению малоэффективным. Кроме того, при неявном задании $F(X)$ ее производные приходится определять численно, а возникающие при этом ошибки, особенно в окрестности экстремума, создают значительные трудности для точного определения точки оптимума.

Методы нулевого порядка простейшего типа заключаются в последовательном изменении одной переменной, тогда как другие пере-

меньшие остаются постоянными до тех пор, пока не будет достигнут минимум. Такие методы поиска называются покоординатными. Например, в простейшем из методов покоординатного спуска — методе Гаусса — Зейделя для случая двух переменных — переменная x_1 устанавливается постоянной, а x_2 изменяется до тех пор, пока не будет получен минимум. Затем, сохраняя новое значение x_2 постоянным, изменяют x_1 , пока не будет достигнут оптимум при выбранном значении x_2 и т. д. Однако такой алгоритм работает плохо, если имеет место взаимодействие между x_1 и x_2 , т. е. если в выражение для целевой функции входят члены, содержащие произведение $x_1 x_2$. Таким образом, этот метод можно рекомендовать только для сепарабельных целевых функций, где взаимодействия между переменными незначительны.

Алгоритм покоординатного спуска можно записать в виде

$$x_i^{(n+1)} = x_i^{(n)} - aQ(x_i^{(n)}, x_1^{(n-1)}, \dots), \quad (14.6)$$

где n — номер итерации; i — номер переменной; $aQ(x_i^{(n)}, x_1^{(n-1)}, \dots)$ — рабочий шаг; $Q(x_1^n, x_1^{n-1}, \dots)$ — обычно вычисляется по формуле

$$Q = \text{sign}(F(x_i^{(n)} + \Delta x_i) - F(x_i^{(n)})).$$

Если в этом алгоритме последовательный поиск минимума по каждой переменной обладает глобальными свойствами, то алгоритм в целом отыскивает глобальный экстремум функции $F(\mathbf{X})$.

В [56] описана логически простая стратегия поиска, использующая априорные сведения и в то же время отвергающая устаревшую информацию о характере целевой функции. Этот алгоритм включает два основных этапа: пробные шаги вокруг базисной точки и рабочие шаги в направлении, выбранном для минимизации. Авторы назвали свой метод *методом конфигураций*.

Алгоритм метода конфигураций состоит из следующих операций. Прежде всего задается начальная точка $\mathbf{X}^0$, а также начальное приращение $\Delta \mathbf{X}$. Чтобы начать пробные шаги, следует вычислить значение функции $F(\mathbf{X})$ в начальной точке. Затем в циклическом порядке изменяется каждая переменная (каждый раз только одна) на выбранные значения приращений, пока все параметры не будут таким образом изменены.

В частности, $x_1^{(0)}$ увеличивается на $\Delta x_1^{(0)}$, так что $x_1^{(1)} = x_1^{(0)} + \Delta x_1^{(0)}$. Если приращение не улучшает целевую функцию, $x_1^{(0)}$ уменьшается на $\Delta x_1^{(0)}$ и значение $F(\mathbf{X})$ проверяется, как и ранее. Если значение $F(\mathbf{X})$ не улучшают ни $x_1^{(0)} + \Delta x_1^{(0)}$, ни $x_1^{(0)} - \Delta x_1^{(0)}$, то $x_1^{(0)}$ оставляют без изменений. Затем $x_2^{(0)}$ изменяют на величину $\Delta x_2^{(0)}$ и т. д., пока не будут изменены все независимые переменные. Этим завершается пробный поиск.

После пробного поиска применяется стратегия рабочего поиска. Удачные изменения переменных в пробном поиске (т. е. те изменения переменных, которые уменьшили $F(\mathbf{X})$) определяют вектор, указывающий локальное направление минимизации, которое может быть удачным. Серия увеличивающихся шагов, или рабочий поиск, проводится вдоль этого вектора до тех пор, пока $F(\mathbf{X})$

уменьшается при каждом таком шаге. Длина рабочего шага обычно выбирается пропорциональной числу удачных шагов, имевших место ранее в этом координатном направлении.

Если пробный поиск не дает удачного направления, то последовательно уменьшают шаги ΔX , пока либо будет найдено новое удачное направление, либо ΔX_n не станет меньше некоторой заранее заданной допустимой величины ε . Невозможность уменьшить целевую функцию $F(X)$, когда шаг ΔX достаточно мал, указывает на то, что достигнут локальный оптимум.

Описанный метод конфигураций весьма просто реализуется на практике и обладает рядом ценных преимуществ: хорошо отслеживает овраги как прямолинейные, так и криволинейные и, не требуя вычисления производных, обладает высокой скоростью сходимости. Однако возможна такая ситуация, когда при попадании рабочей точки в малую окрестность узкого дна оврага при определенном направлении оврага поиск прекращается **зادолго до** нахождения минимума.

Метод Розенброка [56] является итерационной процедурой, имеющей сходство с пробным поиском метода конфигураций, которое состоит в том, что также предпринимаются малые шаги во время поиска в ортогональных направлениях. Метод Розенброка избавлен от недостатка метода покоординатного спуска и метода конфигураций благодаря тому, что в процессе поиска производится поворот координатных осей так, чтобы одна из осей была направлена вдоль направления оврага. Тогда в процессе поиска можно отслеживать даже самые узкие овраги, причем для поворота осей используется информация, получаемая во время поиска.

Алгоритм строится следующим образом. Из исходной точки $X^{(0)}$ ищется минимум вдоль всех n координатных осей так же, как и в методе Гаусса — Зейделя. Получается новая точка $X^{(n)}$, лучшая чем предыдущая. Направление вектора $X^{(n)} - X^{(0)}$ принимается за новое направление первой координатной оси, все остальные оси принимаются взаимно ортогональными. Далее повторяется серия спусков из точки $X^{(n)}$, но уже вдоль новых координатных осей.

Из рис. 14.2 видно, что первая из координатных осей будет разворачиваться вдоль направления дна оврага. Здесь точки $X^{(1)}$ и $X^{(2)}$ получены в результате нахождения минимума целевой функции из точки $X^{(0)}$ вдоль исходных координатных осей x_1 и x_2 . Новая система координат (x'_1, x'_2) получена в результате разворота оси x_1 вдоль направления вектора $X^{(2)} - X^{(0)}$. При этом координат-



Рис. 14.2. Поиск экстремума методом Розенброка

ная ось x'_2 принимается ортогональной оси x'_1 . Следующий поиск минимумов вдоль осей x'_1 и x'_2 дает соответственно точки $X^{(3)}$ и $X^{(4)}$ и новое положение координатных осей x''_1 и x''_2 . Поиск прекращается, когда длина шага становится меньше некоторого заданного значения ε . Метод Розенброка обладает большей эффективностью, чем методы Гаусса — Зейделя и конфигураций при минимизации функций с длинными изогнутыми оврагами, направлениями которых первоначально не совпадает с направлениями координатных осей.

Для нахождения локального экстремума часто используют и методы случайного поиска [56]. Эти методы сочетают в себе случайность при выборе направления поиска с прогнозированием поведения функции на основе проведенных ее вычислений. Один из распространенных алгоритмов случайного поиска строится следующим образом.

Из исходной точки (рис. 14.3) равномерно случайно выбирается направление поиска. После получения первой удачной в смысле улучшения $F(X)$ попытки делается предположение о том, что это направление и в дальнейшем окажется удачным. Тогда область случайного выбора направления сужается, например, вдвое, так что выбор будет осуществляться уже лишь из точек, лежащих на половине гиперсферы, расположенной так, что ее середина лежит на направлении первого удачного шага. Затем назначаются два числа m_1 и m_2 и снова выбирается случайное направление, но теперь уже из половины гиперсферы. Если следующий удачный шаг достигается за число попыток, не больше m_1 , то в новой точке выбор будет производиться уже из четверти гиперсферы. Если же потребуется больше m_1 , но не больше m_2 попыток, то и в новой точке будет выполняться выбор из полусферы. Если число неудачных попыток достигло m_2 , то отказываются от сделанного сужения зоны поиска и начинают опять выбирать из полной гиперсферы, пока опять не находят удачного варианта. После этого в новой удачной точке опять производят выбор из полусферы.

Такое движение с перестраиваемым сектором при сужении сектора повышает вероятность выбора правильного направления, если оно существует внутри сектора, а при расширении — повышает возможность существования правильного направления внутри сектора.

Еще один алгоритм случайного поиска работает следующим образом. Так же, как и в предыдущем случае здесь делается шаг заданной длины в случайном направлении, которое варьируется до тех пор, пока не будет достигнуто уменьшение целевой функции. Из новой точки делается шаг в том же самом направлении и



Рис. 14.3. Поиск экстремума случайным методом с перестраиваемым сектором

той же самой длины. Если первая попытка оказалась удачной, то продвигаются в том же самом направлении, но с шагом не исходной длины, а умноженной на заданный коэффициент $k > 0$. Если попытка неудачна, то делается вторая попытка в противоположном направлении и с шагом, длина которого равна исходной, деленной на заданный коэффициент k . Если же и эта попытка неудачна, то вновь переходят к случайному выбору направления, при этом при каждой неудаче уменьшают длину шага. Когда же, наконец, выполняется удачный шаг, то переходят в новую точку и повторяют ту же процедуру поиска, но уже не с исходной длиной шага, а с полученной в результате процесса поиска.

Следующий алгоритм случайного поиска, реализующий полностью случайный поиск на каждом этапе минимизации, получается из соотношения

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \lambda^{(k)} \left[\beta \frac{\mathbf{Z}^{(k)}}{\|\mathbf{Z}^{(k)}\|} + (1 - \beta) \mathbf{r}^{(k)} \right], \quad (14.7)$$

где $\lambda^{(k)}$ — длина шага, которая увеличивается после успешного шага и уменьшается после неудачного шага; $\mathbf{Z}^{(k)}$ — вектор «предыстории», указывающий среднее направление поиска на предыдущих шагах: $\mathbf{Z}^{(k+1)} = \gamma \mathbf{Z}^{(k)} + (1 - \gamma) (\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}) \mathbf{S}^{(k)}$; $\mathbf{r}^{(k)}$ — единичный вектор нормальных отклонений, реализуемый генератором случайных чисел; β — коэффициент, изменяемый в процессе поиска; γ — постоянный весовой коэффициент; $\mathbf{S}^{(k)}$ — вектор масштабных множителей для масштабирования вектора $\mathbf{X}$.

Как видно из (14.7), чтобы получить точку $\mathbf{X}^{(k+1)}$ из точки $\mathbf{X}^{(k)}$ на k -м этапе поиска усредняются случайный вектор $\mathbf{r}^{(k)}$ и вектор

предыстории $\mathbf{Z}^{(k)}$. Точка $\mathbf{X}^{(k+1)}$ принимается, если выполняется неравенство $F(\mathbf{X}^{(k+1)}) < F(\mathbf{X}^{(k)})$. После того как $\mathbf{X}^{(k+1)}$ принята (или отвергнута) длину шага $\lambda^{(k)}$ увеличивают (или уменьшают) с помощью множителя, грубо говоря, зависящего от того, был ли поиск легким или трудным. На рис. 14.4 приведена получающаяся в результате применения



Рис. 14.4. Поиск экстремума случайным методом, учитывающим усредненное направление поиска на предыдущих шагах:

— — — — — неудачные шаги; ————— — удачные шаги

данного алгоритма гипотетическая траектория для случая двух переменных.

14.2.2. Методы оптимизации, использующие первые производные (методы первого порядка). В основе методов этого класса лежит итерационный процесс типа

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} + \alpha^{(k)} \mathbf{P}^{(k)}, \quad (14.8)$$

где $\mathbf{X}$ — вектор внутренних параметров; $\mathbf{P}$ — вектор направления поиска; α — скаляр, определяющий длину шага в направлении $\mathbf{P}$; k — номер итерации.

В зависимости от выбора значений $\alpha^{(h)}$ и вектора $\mathbf{P}^{(h)}$ можно получить различные алгоритмы оптимизации. При этом для обеспечения минимизации оптимизируемой целевой функции $F(\mathbf{X})$ необходимо выполнять условие

$$(F'(\mathbf{X}^{(h)}), \mathbf{P}^{(h)}) < 0, \quad (14.9)$$

вытекающее из разложения функции $F(\mathbf{X})$ в окрестности точки $\mathbf{X} = \mathbf{X}^{(h)} + \alpha^{(h)} \mathbf{P}^{(h)}$ в ряд Тейлора.

Наиболее популярными в классе методов, определяемых выражением (14.8), являются градиентные методы, характеризующиеся алгоритмом

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - \alpha^{(k)} \nabla F(\mathbf{X}^{(k)}), \quad \alpha^{(k)} > 0, \quad k = 0, 1, 2, \dots \quad (14.10)$$

Как уже отмечалось, градиент целевой функции $\nabla F(\mathbf{X})$ в любой точке $\mathbf{X}^{(k)}$ есть вектор в направлении наибольшего локального увеличения $F(\mathbf{X})$. Следовательно, при поиске минимума нужно двигаться в направлении, противоположном градиенту $F(\mathbf{X})$, поскольку антиградиент $F(\mathbf{X})$ в точке $\mathbf{X}^{(k)}$ направлен в сторону наибольшего уменьшения $F(\mathbf{X})$ по всем переменным $\mathbf{X}$ и ортогонален линии уровня $F(\mathbf{X})$ в точке $\mathbf{X}^{(k)}$.

Антиградиент дает только направление минимизации, но не длину шага. Известно очень большое число различных модификаций градиентного метода в зависимости от выбора значения $\alpha^{(h)}$. Одной из самых распространенных является та, в которой $\alpha^{(h)}$ выбирается из условия минимума $F(\mathbf{X})$ в направлении вектора $\mathbf{P}^{(h)}$, т. е. из условия

$$F(\mathbf{X} - \alpha^{(k)} \nabla F(\mathbf{X}^{(k)})) = \min_{\alpha \geq 0} (F(\mathbf{X}^{(k)}) - \alpha \nabla F(\mathbf{X}^{(k)})). \quad (14.11)$$

Этот вариант градиентного метода называется методом наискорейшего спуска. Поскольку один шаг в направлении наискорейшего спуска в общем случае не приводит в точку минимума $F(\mathbf{X})$, формула (14.11) должна применяться несколько раз, до тех пор пока минимум не будет достигнут.

Последовательность значений $\mathbf{X}^{(k)}$ в градиентных методах сходится к точке минимума со скоростью геометрической прогрессии. Знаменатель этой прогрессии зависит от способа выбора значения $\alpha^{(k)}$. Обычно применяются два общих метода, хотя существуют и многие другие возможности. В первом методе при переходе из точки $\mathbf{X}^{(k)}$ в точку $\mathbf{X}^{(k+1)}$ целевая функция минимизируется по α (14.11), в другом методе значение α выбирается фиксированным или меняется от шага к шагу.

В первом методе $\alpha^{(h)}_{\text{опт}}$, минимизирующее значение $F(\mathbf{X})$ в направлении $\nabla F(\mathbf{X})$, вычисляется в простейшем случае методом последовательных приращений $\Delta \alpha^{(h)}$ с вычислением $F(\mathbf{X})$ в каждой точке $\mathbf{X}^{(k)}$ и проверкой условия

$$F(\mathbf{X}^{(k+1)}) - F(\mathbf{X}^{(k)}) > 0.$$

При выполнении этого условия производится вычисление нового направления спуска и поиск нового значения $\alpha^{(h+1)}_{\text{опт}}$. Так как

здесь ищется $F_{\min}$ по α в направлении $\nabla F(\mathbf{X}^{(k)})$, то очевидно, что поиск можно ускорить, используя процедуру одномерной оптимизации по α , в частности, процедуру «золотого сечения», основанную на свойствах ряда чисел Фибоначчи [56].

Если значение α выбирается фиксированным или меняется от шага к шагу, то его необходимо тщательно контролировать во избежание как неожиданного роста $F(\mathbf{X})$, так и чрезмерного числа шагов, необходимых для достижения оптимума. Первое может произойти, если α слишком велико, а второе, если α очень мало или настолько велико, что приводит к колебаниям текущей точки около точки минимума $F(\mathbf{X})$. Таким образом, значение α должно уменьшаться при приближении к точке минимума. Один из возможных способов контроля α заключается в установлении некоторого критерия для α , основанного на использовании угла θ между последовательными векторами шагов в процессе минимизации. Например, если угол θ становится меньше некоторого заданного значения, то значение α должно быть умножено на некоторую заранее определенную константу λ ; если угол становится больше, то α нужно разделить на λ .

Основной трудностью при использовании метода наискорейшего спуска является его зависимость от масштаба внутренних параметров. Если гиперпространство поиска очень вытянуто, так что образует «овраг» или «гребень» (т. е. отношение максимального собственного значения матрицы Гессе к минимальному в каждой точке велико), процедура наискорейшего спуска сходится слишком медленно, чтобы считаться эффективной или вообще может не сойтись за разумное время. Одним из выходов из данной ситуации является изменение масштаба управляемых переменных в целевой функции. Другим подходом является использование информации второго порядка — информации, даваемой вторыми производными целевой функции по внутренним параметрам или их приближениями.

14.2.3. Методы оптимизации, использующие вторые производные (методы второго порядка). Если в градиентных методах используется только линейный член разложения $F(\mathbf{X})$ в ряд Тейлора (линейная аппроксимация целевой функции), то в методах второго порядка, среди которых наиболее известен метод Ньютона, используется также и квадратный член этого разложения (квадратичная аппроксимация целевой функции) (14.4). Минимум функции (14.4) в направлении $\Delta\mathbf{X}^{(k)}$ определяется дифференцированием $F(\mathbf{X})$ по каждой из переменных $\Delta\mathbf{X}$ и приравниванием нулю полученных выражений. Это приводит к выражению

$$\Delta\mathbf{X}^{(k)} = -(\nabla^2 F(\mathbf{X}^{(k)}))^{-1} \nabla F(\mathbf{X}^{(k)}), \quad (14.12)$$

где $(\nabla^2 F(\mathbf{X}^{(k)}))^{-1}$ — матрица, обратная матрице Гессе $\mathbf{H}(\mathbf{X}^{(k)})$. Подстановка (14.12) в (14.4) определяет переход из точки $\mathbf{X}^{(k)}$ в точку $\mathbf{X}^{(k+1)}$ по обобщенному методу Ньютона:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - (\nabla^2 F(\mathbf{X}^{(k)}))^{-1} \nabla F(\mathbf{X}^{(k)}). \quad (14.13)$$

Заметим, что здесь и направление и длина шага точно определены.

Если $F(\mathbf{X})$ — квадратичная функция, то для достижения минимума $F(\mathbf{X})$ достаточно только одного шага. Но в общем случае нелинейной целевой функции $F(\mathbf{X})$ минимума ее нельзя достичь за один шаг, поэтому (14.13) обычно приводят к виду, аналогичному (14.10), и записывают следующим образом:

$$\mathbf{X}^{(k+1)} = \mathbf{X}^{(k)} - \alpha^{*(k)} \mathbf{H}^{-1}(\mathbf{X}^{(k)}) \nabla F(\mathbf{X}^{(k)}). \quad (14.14)$$

Можно считать, что в этом алгоритме реализуется градиентный метод с автоматической регулировкой обобщенного шага $\alpha^{*(k)} \mathbf{H}^{-1}(\mathbf{X}^{(k)})$. Выбор значений $\alpha^{*(k)}$ производится так же, как и в градиентном методе.

Основное достоинство метода Ньютона — квадратичная скорость сходимости. Основной недостаток — необходимость больших вычислительных затрат на расчет, хранение и обращение матрицы Гессе $\mathbf{H}(\mathbf{X}^{(k)})$, а также высокая чувствительность сходимости к начальному приближению. Необходимо также отметить, что этот метод требует вычисления аналитических значений вторых частных производных или их аппроксимаций, что при схемотехническом проектировании представляет существенные трудности. В связи с этим при проектировании обычно применяются различные модификации метода Ньютона, например вычисление $\mathbf{H}(\mathbf{X}^{(k)})$ только в начальной точке или только через определенное число шагов итераций. Скорость сходимости при этом, естественно, уменьшается.

Обычно число итераций в методе Ньютона незначительно, но большие вычислительные затраты на каждую итерацию не всегда дают преимущества данному методу. Поэтому на практике часто применяются модификации метода, имеющие скорость сходимости, близкую к квадратичной, но не требующие непосредственного вычисления производных. Замена матрицы вторых производных информацией о первых производных основывается на том, что если взять две достаточно близкие точки $\mathbf{X}^{(i+1)}$ и $\mathbf{X}^{(i)}$ и записать систему уравнений

$$\nabla F(\mathbf{X}^{(i+1)}) - \nabla F(\mathbf{X}^{(i)}) = \mathbf{A}(\mathbf{X}^{(i+1)} - \mathbf{X}^{(i)}), \quad i = 1, 2, \dots, n, \quad (14.15)$$

то квадратная матрица $\mathbf{A}$ должна быть близка к матрице вторых производных для любых $i \in \{1, n\}$, поскольку (14.15) можно рассматривать как усеченное разложение в ряд Тейлора функции $\nabla F(\mathbf{X})$ в окрестности точки $\mathbf{X}^{(i+1)}$.

Одной из таких модификаций метода Ньютона является метод сопряженных градиентов [58], в котором поиск организуется вдоль векторов, являющихся сопряженными. Понятие $\mathbf{Q}$ -сопряженности векторов $\mathbf{C}$ и $\mathbf{B}$ при заданной матрице $\mathbf{Q}$ основано на соблюдении условия равенства нулю скалярного произведения: $\langle \mathbf{C}, \mathbf{QB} \rangle = 0$, где $\mathbf{Q}$ — квадратная матрица того же порядка, что и размерность векторов $\mathbf{C}$ и $\mathbf{B}$.

В [56] доказывається, що якщо використовуються спряжені напрямлення, то будь-яка квадратична функція n змінних, яка має мінімум, може бути мінімізована за n кроків, причому кожен крок робиться в одному з спряжених напрямлень, а порядок використання цих напрямлень не має значення. Таким чином, збіжність методу спряжених градієнтів гарантована тільки для квадратичних функцій.

Незважаючи на те, що при проектуванні цільові функції можуть мати найрізноманітніший характер, тим не менше метод спряжених градієнтів широко застосовується в задачах схематехнічного проектування [9]. Це пояснюється тим, що в окрестностях точки пошуку цільова функція може бути достатньо хорошою апроксимована квадратичною функцією. Однак чим сильніше відрізняється цільова функція від квадратичної, чим менше вона гладка, тим більше вимагається для її мінімізації ітерацій по n кроків в кожній, т. є. тим менше швидкість збіжності методу і гірше надійність пошуку.

Алгоритм методу спряжених градієнтів будується наступним чином [58]:

1. В точці $\mathbf{X}^{(0)}$ обчислюється градієнт цільової функції і визначається перше напрямлення пошуку $\mathbf{S}^{(0)} = -\nabla F(\mathbf{X}^{(0)})$.

2. На будь-якому k -му кроці з допомогою одномерного пошуку в напрямленні $\mathbf{S}^{(k)}$ знаходиться мінімум цільової функції $F(\mathbf{X})$. Це визначає нову точку пошуку $\mathbf{X}^{(k+1)}$.

3. В новій точці пошуку визначається цільова функція $F(\mathbf{X}^{(k+1)})$ і градієнт цільової функції $\nabla F(\mathbf{X}^{(k+1)})$.

4. Нове напрямлення пошуку визначається із співвідношення $\mathbf{S}^{(k+1)} = -\nabla F(\mathbf{X}^{(k+1)}) + S^{(k)}\omega_k$, де

$$\omega_k = (\nabla^T F(\mathbf{X}^{(k+1)}) \nabla F(\mathbf{X}^{(k+1)})) / (\nabla^T F(\mathbf{X}^{(k)}) \nabla F(\mathbf{X}^{(k)}))$$

визначається із умови спряженості напрямлень пошуку $\mathbf{S}^{(k+1)}$ і $\mathbf{S}^{(k)}$. Після $(n+1)$ -ї ітерації ($k=n$) процедура пошуку циклічно повторюється з заміною $\mathbf{X}^{(0)}$ на $\mathbf{X}^{(n+1)}$.

5. Пошук закінчується, коли $\|\mathbf{S}^{(k)}\| < \epsilon$, де ϵ — задана заздалегідь константа.

Таким чином, метод спряжених градієнтів суттєво відрізняється від методів другого порядку тим, що для обчислення нового напрямлення пошуку в точці $\mathbf{X}^{(k+1)}$ використовуються тільки текущий градієнт і передостанній градієнт.

На питання про те, які методи локального пошуку в якому випадку потрібно застосовувати, на жаль, неможливо дати однозначної відповіді. Все залежить від виду цільової функції, з якою ми маємо справу в кожній конкретній задачі оптимізації. Якщо цільова функція сепарабельна, т. є. така, що її зміни, викликані зміною однієї змінної, просто сумуються з змінами, викликаними іншими змінними, що означає, що $F(x_1, x_2, \dots, x_n)$ мало відрізняється від $F_1(x_1) + F_2(x_2) + \dots + F_n(x_n)$, то найкраще всього використовувати релаксационні методи з групи методів пошуку по координатах. Якщо всі змінні тісно зв'язані

заны и имеют одинаковую чувствительность, т. е. сходны, то лучше всего будут работать градиентные методы. Для функций, имеющих вид, близкий к квадратичному, предпочтительно использовать методы второго порядка. Когда переменные совершенно несходны и лишь выбором масштабов могут быть приведены к внешне однородному безразмерному виду, то лучше всего, по-видимому, использовать методы случайного поиска.

Кроме того, эти методы предпочтительнее использовать и на первых этапах поиска, так как они быстро ликвидируют большие отклонения от экстремума.

Если же вид функции цели совершенно неизвестен, как обычно бывает при проектировании РЭУ, то целесообразно при оптимизации перебирать или комбинировать несколько методов из различных групп, поэтому библиотеки методов оптимизации в комплексах программ САПР содержат, как правило, по одному или по несколько методов каждого класса.

14.3. МЕТОДЫ ПОИСКА ГЛОБАЛЬНОГО ЭКСТРЕМУМА

14.3.1. Постановка задачи. Для многоэкстремальной функции $F(\mathbf{X})$ зоной притяжения S_i любого экстремума $\mathbf{X}^*_i$ называется та часть пространства параметров $\mathbf{X}$, где выполняются следующие условия (рис. 14.5):

экстремум $\mathbf{X}^*_i$ принадлежит этой зоне, т. е. $\mathbf{X}^*_i \in S_i$;

все антиградиентные линии этой зоны сходятся в точку экстремума $\mathbf{X}^*_i$.

Зоной глобальности S^{**} является такая часть зоны притяжения S^* глобального экстремума $\mathbf{X}^{**}$, в которой значения целевой функции меньше, чем в остальных локальных экстремумах (рис. 14.6). Чем меньше S^{**} , тем больше затраты на поиск. Рассмотренные зоны характеризуются объемами:

$$V_i = V(S_i), \quad V^* = V(S^*), \quad V^{**} = V(S^{**}),$$

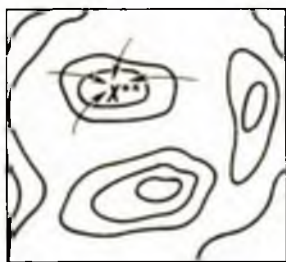


Рис. 14.5. Зоны притяжения локальных экстремумов целевой функции двух переменных

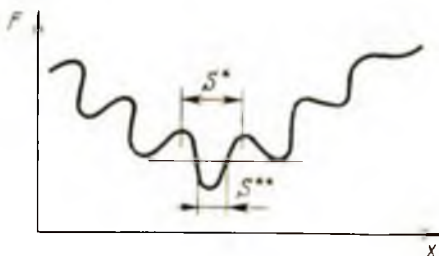


Рис. 14.6. Зона притяжения глобального экстремума S^* и зона глобальности S^{**} многоэкстремальной целевой функции $F(x)$

где V_i — объем зоны притяжения i -го экстремума X^*_i ; V^* — объем зоны притяжения глобального экстремума; V^{**} — объем зоны глобальности.

Величины $V_1, V_2, \dots, V_N, V^*, V^{**}$ определяют степень сложности отыскания глобального решения в данной задаче. В одноэкстремальной задаче выполняется условие $V_1 = V^* = V^{**}$ ($N=1$). Многоэкстремальная задача характеризуется следующими соотношениями объемов: $V^* > V^{**} \ll V_0$, где V_0 — объем зоны поиска.

Задачу поиска глобального экстремума можно сформулировать различным образом:

1) при заданном объеме испытаний K найти оценку положения глобального экстремума X^{**} , такую чтобы она в наименьшей степени отличалась от действительного положения глобального экстремума X^{**} , т. е. $|X^{**} - X^{**}| \rightarrow \min$;

2) при заданном объеме испытаний K найти оценку X^{**} , такую чтобы значение целевой функции в этой точке в наименьшей степени отличалось от глобального, т. е. $F(X^{**}) - F(X^{**}) \rightarrow \min$;

3) определить оценку положения глобального экстремума X^{**} с точностью не меньше заданной $\varepsilon > |X^{**} - X^{**}|$, затратив на это минимальное число испытаний: $K \rightarrow \min$;

4) определить оценку X^{**} , целевая функция в которой отклоняется от глобального значения на значение не более заданного $\varepsilon > F(X^{**}) - F(X^{**})$ при минимальном числе испытаний: $K \rightarrow \min$.

В 3- и 4-й постановках задача с помощью случайных методов может быть решена только с определенной вероятностью. Существует несколько типов методов, обеспечивающих решение задачи в одной из данных постановок [58, 59].

14.3.2. «Независимый» глобальный поиск. Методы этого типа имитируют независимую случайную выборку в соответствии с заданным законом распределения. Простейшим «независимым» алгоритмом «слепого» глобального поиска является совершенно случайный перебор, который сводится к следующему. На каждом шаге в соответствии с заданной плотностью распределения $P(X)$ независимо и случайно определяется вектор X_i , вычисляется целевая функция, соответствующая этому вектору $F(X_i)$, и полученное значение сравнивается с хранимым в памяти значением F^0 . Если $F(X_i) \geq F^0$, то производится очередной случайный эксперимент, если $F(X_i) < F^0$, то новое значение целевой функции $F(X_i)$ запоминается в качестве F^0 , так же как и вектор X_i , который соответствует этому улучшению целевой функции. После этого делается очередной случайный эксперимент.

Как можно заметить, этот алгоритм гарантирует отыскание глобального экстремума при $K \rightarrow \infty$, так как в силу случайного характера поиска когда-нибудь будет выбран вектор, отличающийся от глобального экстремума не более чем на ε , если ε — конечная окрестность экстремума. Однако для решения практических задач

таким способом необходимо значительное число испытаний, которое не всегда удастся реализовать. Поэтому рассмотренный метод «слепого» глобального поиска имеет весьма ограниченное применение.

14.3.3. Случайный поиск с управлением плотностью испытаний. Недостаток предыдущего метода заставил обратиться к изменению плотности распределения случайных испытаний $P(X)$ с тем, чтобы увеличить вероятность отыскания состояния, близкого к глобальному экстремуму. При этом каждый раз возрастает плотность испытаний в той подобласти общей области поиска, которая была признана наилучшей на основании предшествующих испытаний. Такая адаптация распределения испытаний позволяет значительно повысить эффективность поиска.

Этот поиск обеспечивает отыскание глобального экстремума с высокой точностью при одном существенном условии: глобальный экстремум X^{**} не должен быть утерян в процессе сужения зоны поиска, поэтому во всей области поиска не должно существовать такой точки, которая на очередном шаге не могла бы оказаться точкой испытания. Для обеспечения этой цели целесообразно делать распределение испытаний не равномерным, а нормальным с центром в наилучшей точке предыдущего шага поиска и дисперсией, увеличивающейся при неудачных испытаниях и уменьшающейся при удачных.

14.3.4. «Блуждающий» глобальный поиск. «Блуждающий» глобальный поиск является статистическим расширением детерминированного локального градиентного метода и заключается в спуске на каждом этапе поиска градиентным методом из случайно выбранной начальной точки. Найденное в конце каждого этапа значение запоминается в случае, если оно лучше предшествующих. Если целевая функция имеет не слишком большое число экстремумов, то можно предположить, что по истечении достаточно большого числа этапов мы переберем все локальные экстремумы и найдем лучший из них. В принципе, как и для алгоритмов предыдущего класса, никакое конечное число этапов не дает полной гарантии обнаружения глобального экстремума. Поиск заканчивается, если 5... 10 последних попыток не приводят к улучшению результата.

Математик Р. Фриш назвал этот метод сотрудничеством кенгуру и мыши. Данный метод обладает тем же недостатком, что и все градиентные методы — требует больших затрат времени.

Общим недостатком всех случайных методов поиска является снижение эффективности поиска при приближении к экстремуму, что и диктует необходимость использовать сочетания случайных и детерминированных методов для повышения эффективности при непосредственном приближении к экстремуму.

14.4. МЕТОДЫ УЧЕТА ОГРАНИЧЕНИЙ

Существует несколько способов сведения задачи условной оптимизации к задаче безусловной оптимизации, в частности мето-

ды линейной аппроксимации ограничений и целевой функции для сведения исходной задачи к задаче линейного программирования, методы выбора возможных направлений в области поиска и т. д. Большинство из них требуют наличия ограничений дифференцируемого или выпуклого вида, ориентированы на определенные методы безусловной оптимизации и не эффективны при работе с другими методами. Рассмотрим наиболее универсальные методы учета ограничений, применяемые в задачах проектирования.

14.4.1. Метод внешней штрафной функции. Если задача оптимизации ставится в постановке (14.1), (14.2), то идея метода внешних штрафных функций сводится к замене исходной целевой функции $F(\mathbf{X})$ функцией $\Phi(\mathbf{X})$, которая максимально совпадает с функцией $F(\mathbf{X})$ в области работоспособности $G(\mathbf{X})$ при выполнении всех ограничений и существенно превосходит ее вне этой области. Таким образом, нарушение ограничений карается наложением штрафа на целевую функцию, приводящего к ее ухудшению

$$\Phi(\mathbf{X}) = F(\mathbf{X}) + \theta_h(\mathbf{X}),$$

в качестве штрафа обычно выбирается функция [64]

$$\theta_h(\mathbf{X}) = r_h \sum_{i=1}^m [\max \{0, g_i(\mathbf{X})\}]^2,$$

где коэффициент $r_h > 0$ определяет размер штрафа.

Введение штрафной функции $\theta_h(\mathbf{X})$ не изменяет экстремумов, лежащих внутри области работоспособности, но влияет на значение тех условных экстремумов, которые расположены на границе области. Для сближения значений экстремумов функций $\Phi(\mathbf{X})$ и $F(\mathbf{X})$ необходимо увеличивать значение коэффициента r_h , но значительное его увеличение может привести к переполнению разрядной сетки ЭВМ. В связи с этим обычно задача безусловной минимизации функции $\Phi(\mathbf{X})$ сводится к многократному решению задач с последовательно увеличивающимися коэффициентами r_h , до тех пор пока два последующих решения не окажутся достаточно близкими друг к другу. В рассмотренном методе траектория поиска может выходить за область работоспособности, поэтому этот метод носит название метода внешней точки или метода внешней штрафной функции.

14.4.2. Метод внутренней штрафной функции. В другом варианте метода штрафные функции носят характер барьерных функций, не позволяющих траектории поиска покидать область работоспособности. Этот метод носит также название метода барьерных функций или метода внутренней точки. Здесь новая целевая функция формируется следующим образом:

$$\Phi(\mathbf{X}) = F(\mathbf{X}) - \theta_b(\mathbf{X}),$$

где $\theta_b(\mathbf{X}) = -r_b \sum_{i=1}^m (1/g_i(\mathbf{X}))$, $r_b > 0$.

Начальная точка поиска $\mathbf{X}^0$ должна быть выбрана в области работоспособности. Если траектория поиска приближается к гра-

нице области, начинает стремиться к нулю какое-либо из ограничений $g_i(\mathbf{X})$, в результате чего неограниченно возрастает по абсолютному значению барьерная функция $\theta_k(\mathbf{X})$, что не позволяет траектории поиска приближаться к границам области.

Здесь так же, как и в предыдущем случае рекомендуется решать последовательность задач безусловной минимизации $\Phi(\mathbf{X})$ с монотонно уменьшающимися коэффициентами r_k . Эта последовательность минимумов $\Phi(\mathbf{X})$ сходится к минимуму целевой функции $F(\mathbf{X})$. Недостатком данного метода является необходимость выбора начальной точки внутри области работоспособности, что представляет собой весьма трудную задачу при малых размерах области работоспособности по сравнению с областью поиска.

14.5. АЛГОРИТМЫ СТАТИСТИЧЕСКОЙ ОПТИМИЗАЦИИ СХЕМЫ

14.5.1. Алгоритм оптимизации процента выхода годных схем. Рассмотрим случай, когда в качестве статистического критерия при производстве схем используется процент выхода годных схем.

Оптимизацию задач такого рода, обладающих статистическими критериями, очень удобно проводить методами случайного поиска с управлением плотностью испытаний, так как сама природа существования разброса исходных данных предполагает возможность управления плотностью распределения испытаний.

Процесс оптимизации может строиться следующим образом. Выбирается начальное значение вектора математических ожиданий внутренних параметров схемы: $\mathbf{M}\mathbf{X}^0 = (mx^0_1, \dots, mx^0_p)$. В дальнейшем его значение последовательно изменяется таким образом, чтобы соответствующее ему значение процента выхода годных схем $P(\mathbf{X})$ увеличивалось до максимально возможного

$$mx_k^{n+1} = m^* x_k^n (v_{x_k} = \text{const}, k = \overline{1, p}), \quad (14.16)$$

где $v_{x_k} = \sqrt{Dx_k}/(mx_k)$ — коэффициент вариации; Dx_k — дисперсия параметра x_k ; p — число внутренних параметров; $\mathbf{X}^* = \{\mathbf{X} | f(\mathbf{X}) \in G\}$, где G — область работоспособности, $\mathbf{X} = (x_1, \dots, x_k, \dots, x_p)$.

Запись (14.16) означает, что математическими ожиданиями mx_k^{n+1} внутренних параметров схемы на $(n+1)$ -м шаге оптимизации являются математические ожидания $m^* x_k^n$ параметров, вычисленные только для тех схем, которые удовлетворили всем условиям работоспособности на предыдущем n -м шаге оптимизации.

Роль критериев отбора в этом алгоритме играют условия работоспособности и технические требования, задающие область работоспособности. Как отмечалось в гл. 13, задача оптимизации процента выхода является технически оправданной для изготовителя схем, но после того как схема попала к потребителю, ее параметры изменяются в результате старения и могут, в частности, выходить за пределы области работоспособности, вследствие чего схема перестает быть годной для использования. В связи с этим возникает задача оптимизации надежности схем.

14.5.2. Алгоритм оптимизации срока службы схем. Эта задача ставится следующим образом: максимизировать вероятность того, что выходные параметры схемы $f_1, \dots, f_m$ будут удовлетворять заданным техническим требованиям в течение времени t , которое не меньше заданного $t_{\text{доп}}$. Для решения этой задачи можно использовать алгоритм (14.16), но с учетом надежности схем. Для этого в (14.16) под mx_k нужно понимать математические ожидания только тех внутренних параметров x_k , для которых времена движения t_i всех выходных параметров f_i к границам области работоспособности удовлетворяют условию $t_i > t_{\text{доп}}$. Таким образом, алгоритм оптимизации надежности строится следующим образом.

1. Моделируются по соответствующим законам распределения внутренние параметры X и с использованием математической модели схемы рассчитываются все выходные параметры f_i .

2. Для каждого выходного параметра f_i с помощью критерия, рассмотренного в гл. 13, рассчитывается время t_i удовлетворения заданному техническому требованию $f_i > f_{\text{гр}}$ или $f_i < f_{\text{гр}}$, где $f_{\text{гр}}$ — граничное допустимое значение данного выходного параметра f_i .

3. Для каждого выходного параметра f_i проверяется условие $t_i > t_{\text{доп}}$. Схема считается годной лишь при выполнении этого условия для всех выходных параметров. Если хотя бы один из f_i не удовлетворяет этому условию, схема считается негодной, а испытание неудачным.

4. В результате серии испытаний определяются подмножества внутренних параметров

$$x_k^i \in x^i = \{x | t_i > t_{\text{доп}}, i = \overline{1, m}\}$$

годных схем, вычисляются их средние значения и выполняется шаг оптимизации по алгоритму (14.16). После окончания процесса оптимизации определяются значения математических ожиданий внутренних параметров mx_k , обеспечивающих максимальную вероятность удовлетворения схемой заданным техническим требованиям в течение заданного времени $t_{\text{доп}}$.

14.6. СТРУКТУРА ПАКЕТА ПРОГРАММ ОПТИМИЗАЦИИ СХЕМ

14.6.1. Общие требования к пакету программ оптимизации. Опыт разработки комплексов программ анализа и оптимизации [1, 9, 37] позволяет сформулировать основные требования, которые должны предъявляться к пакету программ оптимизации, входящему в данный комплекс.

Пакет должен обладать развитой модульной структурой для облегчения замены отдельных модулей, добавления новых и введения изменения в них без нарушения структуры других программных модулей.

Язык описания задания на оптимизацию должен быть удобным для пользователя, обладать функциональной полнотой, т. е. позволять составлять любое задание на оптимизацию в удобной, информативной и лаконичной форме.

Пакет должен иметь возможность реализовывать различные режимы оптимизации, необходимые при проектировании схем.

Пакет должен содержать обширные библиотеки методов оптимизации и критериев оптимальности. Несколько методов одного и того же класса позволяють разработчику выбрать тот из них, который является оптимальным для каждой конкретной задачи. Библиотеки должны быть открытыми для пользования, т. е. иметь возможность легко пополняться новыми методами, критериями и т. д.

Пакет должен иметь возможность работы в диалоговом режиме и выдавать результаты оптимизации в виде, удобном для пользователя: графики, таблицы и т. д.

Пакет должен быть написан на универсальном алгоритмическом языке высокого уровня, позволяющем использовать его на ЭВМ различного класса и легко стыковать с пакетами программ анализа различного вида, как специального, так и универсального.

В настоящее время для реализации пакетов программ оптимизации в большинстве случаев используются универсальные алгоритмические языки Фортран, Паскаль и ПЛ/1.

14.6.2. Состав пакета программ оптимизации. В состав пакета программ оптимизации должна входить обширная библиотека методов оптимизации, рассчитанная, как правило, на решение следующих задач:

- оптимизация одномерных одноэкстремальных целевых функций оптимальными одномерными методами;

- оптимизация одноэкстремальных квадратичных целевых функций нескольких переменных с помощью методов первого или второго порядка;

- оптимизация одноэкстремальных целевых функций нескольких переменных овражного типа детерминированными методами нулевого порядка;

- оптимизация многоэкстремальных целевых функций одной или нескольких переменных или многопараметрических целевых функций детерминированными или чаще случайными и комбинированными методами глобального поиска.

Кроме того, в пакете программ необходима библиотека случайных законов распределения, включающая в себя равномерный закон распределения, нормальный закон распределения и другие законы, необходимые проектировщику схем.

Для правильного выбора метода оптимизации необходимо знать рельеф целевой функции (одноэкстремальный, овражный и т. д.), однако разработчик схемы, приступая к оптимизации ее параметров, как правило, такой информации не имеет. Поэтому желательно в пакете программ оптимизации иметь один из методов предварительного исследования вида целевой функции и ее идентификации для более рационального выбора метода оптимизации и накопления начальной информации для организации процесса поиска (начальной точки поиска, начального шага и т. д.). При реализации такого исследования можно использовать графический

дисплей для визуального определения рельефа целевой функции или проводить исследование сечений функции и построение на его основе критерия для автоматизированного принятия решения о выборе метода оптимизации.

Помимо этого, пакет программ оптимизации должен содержать библиотеку критериев оптимальности, включающую все наиболее распространенные критерии и позволяющую проводить оптимизацию параметров схемы по частным критериям или обобщенным критериям любого вида. Все библиотеки должны иметь возможность пополнения новыми методами и критериями, замены одного метода или критерия другим и автоматического перехода в процессе оптимизации с одного метода или критерия на другой.

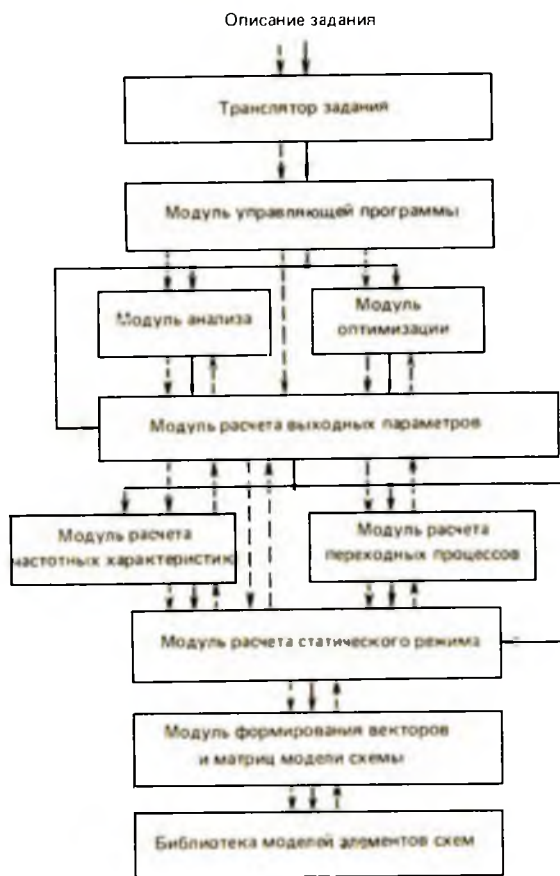


Рис. 147. Структурная схема взаимодействия основных программ схемотехнического проектирования:

— — — — — информационные связи; — — — — — управляющие связи

14.6.3. Взаимодействие программ расчета, анализа и оптимизации. В современных САПР практикуется нерархический способ построения программного обеспечения. Иерархический способ построения программ подразумевает не равноправное взаимодействие программ одного уровня, а взаимодействие программ разного уровня таким образом, что обращение к программам нижнего уровня иерархии осуществляется по командам из программ верхнего уровня иерархии. Программы разного уровня представляются как бы «вложенными» одна в другую по принципу «матрешки». Взаимодействие программ расчета токов и напряжений схемы, анализа выходных параметров схемы и оптимизации представлено на рис. 14.7.

В комплексе программ каждая программа играет роль подпрограммы и оформляется в виде программного модуля, имеющего информационные и управляющие связи с другими модулями. Модули оптимизации и анализа имеют одинаковый уровень старшинства. В модуле оптимизации или анализа происходит вызов модуля расчета выходных параметров и передача в него того значения вектора внутренних параметров X , с которым надо выполнить расчет. Структура дальнейших вызовов ясна из рис. 14.7. После завершения работы модуля более низкого уровня происходит возврат в модуль более высокого уровня с необходимыми для продолжения его работы данными, рассчитанными в подчиненном модуле.

В частности, когда рассчитаны все выходные параметры, происходит возврат в модуль оптимизации, где из полученного вектора выходных параметров формируется обобщенный критерий оптимальности, подсчитывается его численное значение и в соответствии с алгоритмом оптимизации определяется следующий вектор внутренних параметров, т. е. делается шаг оптимизации. Модуль оптимизации обращается к модулю расчета столько раз, сколько нужно для определения экстремума.

Упрощенная структурная схема модуля оптимизации изображена на рис. 14.8. В управляющей подпрограмме организуется

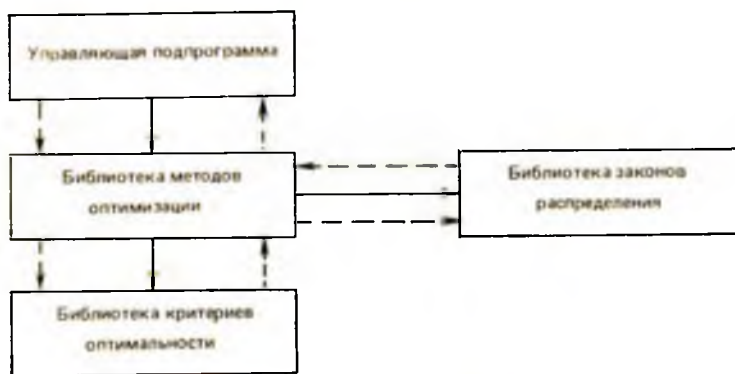


Рис. 14.8. Структурная схема пакета программ оптимизации:
 — — — — — информационные связи; — — — — — управляющие связи

обработка задания на оптимизацию, формируется вектор варьируемых внутренних параметров схемы X , производится выбор метода оптимизации, учет ограничений на внутренние параметры схемы.

14.7. ЛИНГВИСТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПАКЕТА ПРОГРАММ ОПТИМИЗАЦИИ

В задании на оптимизацию должны указываться те же данные, что и в задании на расчет выходных параметров и на анализ, так как при вычислении целевой функции происходит многократное обращение к программам анализа и расчета выходных параметров. Следовательно, в задании должны присутствовать директивы на расчет выходных параметров и информация об изменяемых внутренних параметрах. Кроме того, в задании на оптимизацию должен быть указан тип критерия оптимальности, характер ограничений на входные и внутренние параметры, используемые метод оптимизации и его параметры и законы распределения со своими параметрами.

Структура директив на оптимизацию параметров схем должна включать несколько функциональных групп [33]. В первую группу должна входить директива, задающая метод оптимизации и его параметры. Например, эта директива может записываться в виде

ОПТИМИЗИРОВАТЬ:

ИМЯ метода ($A_1, A_2, \dots$)

Каждый из методов имеет свой набор параметров $A_1, A_2, \dots$. Обязательным среди параметров является EPS — точность нахождения экстремума. Так, например, для детерминированного метода конфигураций набор параметров состоит из параметра EPS и параметра DLT , определяющего начальный шаг поиска.

Во второй группе параметров описывается целевая функция, т. е. задается обобщенный критерий оптимальности, параметры критерия $B_1, B_2, \dots$ и его аргументы. Аргументами критерия служат заданные директивой расчета выходные параметры $Z_1, Z_2, \dots$, являющиеся частными критериями. Задание целевой функции имеет вид

КРИТЕРИЙ Q :

$Q = \text{ИМЯ критерия } (B_1, B_2, \dots) Z_1, Z_2, \dots$

Имя обобщенного критерия может задаваться, например, так: АДДИТИВНЫЙ, МУЛЬТИПЛИКАТИВНЫЙ и т. д., а параметры соответствуют весовым коэффициентам или коэффициентам нормирования. Должна быть также предусмотрена возможность задания обобщенного критерия обращением к процедуре описания произвольного аналитического выражения, аргументами которого служат выходные параметры $Z_1, Z_2, \dots$.

В третьей группе должны быть описаны ограничения на выходные параметры Z и признак, по которому производится максими-

зация или минимизация данного выходного параметра. Директива имеет вид

ПАРАМЕТРЫ Z:

$Z1 = \text{MAX} (ГН, ГВ)$

$Z2 = \text{MIN} (ГН, ГВ)$

$Z3 = (ГН, ГВ)$

Здесь $ГН$ и $ГВ$ — нижняя и верхняя границы допустимой области изменения соответствующего выходного параметра. По данной директиве выходной параметр $Z3$ не является частным критерием, а должен при оптимизации находиться в допустимой области изменения.

В последней группе описания задания на оптимизацию заключена информация о варьируемых внутренних параметрах схемы. Структура задания этой информации аналогична конструкциям описания варьируемых параметров в исходной информации на анализ. Возможны три типа конструкций описания внутренних параметров. Первая используется для описания независимых внутренних параметров для детерминированных методов оптимизации и задает символьным обозначением «ИНТ» интервал изменения внутреннего параметра от нижней границы $ГН$ до верхней границы $ГВ$ и начальную точку поиска $НТ$.

Вторая конструкция используется для случайных методов и задает обозначением «ИМЯ ЗАКОНА» закон изменения внутреннего параметра, например РАВНОМЕРНЫЙ, НОРМАЛЬНЫЙ и т. д. Набор параметров этой конструкции содержит ограничения и параметры случайного закона распределения $D1, D2, \dots$

Третья конструкция описывает зависимые внутренние параметры, т. е. параметры, функционально связанные с независимыми внутренними параметрами. Необходимость в описании такого рода параметров часто возникает при формулировке оптимизационных задач в интегральной схемотехнике, например, суммарная ширина каналов двух МДП-транзисторов может быть зафиксирована: $W1 + W2 = A$, но при этом $W1$ — независимый параметр, а $W2$ — зависимый параметр. Конструкция этого типа содержит ИМЯ закона, задающего функциональную зависимость между параметрами (например, ЛИНЕЙНЫЙ, КВАДРАТИЧНЫЙ и т. д.), параметры закона $C1, C2, \dots$ и символьные обозначения аргументов $X1, X2, \dots$, являющихся независимыми внутренними параметрами.

Ниже приведено описание трех рассмотренных типов конструкций:

ПАРАМЕТРЫ X:

$X1$) ИМЯ элемента (ИМЯ параметра) = ИНТ ($ГН, ГВ, НТ$)

$X2$) ИМЯ элемента (ИМЯ параметра) = ИМЯ закона ($ГН, ГВ, D1, D2, \dots$)

ХЗ) ИМЯ элемента (ИМЯ параметра) = ИМЯ закона (С1, С2, ...) Х1, Х2.

Возможны и другие варианты записи информации для оптимизации [9, 37].

14.8. МЕТОДОЛОГИЯ ИСПОЛЬЗОВАНИЯ ПАКЕТА ПРОГРАММ ОПТИМИЗАЦИИ

Процесс оптимизации — наиболее трудоемкая часть задачи проектирования, поэтому при его использовании желательно обеспечить всемерное снижение вычислительных затрат. Этого можно достигнуть следующими способами.

14.8.1. Формулировка ограничений. Правильная формулировка ограничений позволяет сразу задать область работоспособности, в которой возможен поиск экстремума целевой функции и, таким образом, свести задачу к стандартной задаче нелинейного программирования. Если технические требования на схему заданы, то параметрические ограничения на внутренние параметры схемы X и функциональные ограничения на выходные параметры схемы Y можно считать заданными. После этого можно приступить к решению задачи параметрической оптимизации.

В случае, когда проектируется новая сложная схема и ограничения заранее не известны, можно рекомендовать предварительное однократное или многократное проведение многовариантного анализа, в результате которого пользователь должен сначала назначить приемлемые границы изменения внутренних параметров, а затем на основании полученных значений выходных параметров — уже функциональные ограничения. При этом необходимо помнить, что назначенные ограничения должны быть совместимы, т. е. давать в результате совместную область допустимых решений, в которой в дальнейшем проводится оптимизация выбранных критериев.

14.8.2. Выбор метода оптимизации. Выбор правильной стратегии оптимизации позволяет существенно снизить число обращений к математической модели схемы и увеличить надежность и точность поиска экстремума.

Как уже отмечалось, если о виде целевой функции не существует никакой достоверной априорной информации, то можно рекомендовать первоначально провести оптимизацию во всей допустимой области с использованием случайных глобальных методов поиска с тем, чтобы далее вблизи одной (или нескольких) найденных лучших точек уточнить результат с помощью одного из имеющихся в библиотеке методов локального поиска.

14.8.3. Выбор начальной точки поиска. При выборе начальной точки поиска необходимо помнить, что для экономии числа испытаний она должна находиться как можно ближе к экстремуму и в допустимой области поиска. Поэтому ее тоже желательно назначить, исходя из результатов многовариантного анализа и таким

образом, чтобы она соответствовала координатам лучшего из всех проведенных с помощью многовариантного анализа испытаний.

С помощью многовариантного анализа пользователь может также определить наиболее существенные внутренние параметры схемы, т. е. те параметры, которые оказывают наибольшее влияние на выходные параметры, а следовательно, и на критерий оптимальности. Исключение из числа варьируемых при оптимизации параметров, слабо влияющих на критерий оптимальности, позволяет существенно уменьшить затраты на поиск экстремума.

14.8.4. Общая методология решения задачи оптимизации. Помимо рассмотренных выше вопросов при использовании пакета программ оптимизации пользователю необходимо сформулировать задачу в терминах, допускаемых данной программой.

После того как пользователем рационально выбраны варьируемые внутренние параметры, область их изменения, начальная точка поиска, выходные параметры и ограничения на них и метод поиска, ему необходимо:

- задать критерии точности и другие параметры выбранного алгоритма;

- выбрать нужный критерий оптимальности из числа имеющихся в библиотеке программы, задать весовые коэффициенты в критерии;

- выбрать модели элементов (обычно для проведения оптимизации выбираются наиболее простые модели или макромодели для снижения вычислительных затрат); после того как экстремум найден, его можно уточнить, используя более точные модели;

- при использовании случайных методов поиска выбрать соответствующий закон распределения внутренних параметров из библиотеки законов и задать его параметры;

- записать все директивы на оптимизацию на входном языке данного комплекса.

Разработчик схем должен помнить, что для оптимизации необходимо не менее 50 ... 100 обращений к математической модели схемы, поэтому на основании результатов однократного анализа схемы пользователь должен оценить реальность решения задачи оптимизации и, в случае необходимости, либо пользоваться макромоделями вместо точных моделей, либо проводить пофрагментную оптимизацию отдельных функциональных узлов схемы.

ГЛАВА 15

ОСНОВЫ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

15.1. ПОСТАНОВКА ЗАДАЧИ

При проектировании цифровых РЭУ зачастую возникает необходимость моделирования и анализа объектов с дискретным описанием [2, 37]. У таких устройств варьируемые внутренние параметры (например, параметры микросхем) или оптимизируемая

целевая функция (например, площадь кристалла БИС) в результате технологических причин могут изменяться только дискретным образом — либо с определенным шагом, либо из определенного конечного набора возможных значений.

Математическими методами оптимального проектирования задач подобного рода являются методы дискретной оптимизации.

Постановка задачи дискретной оптимизации вытекает из общей задачи математического программирования (14.1), (14.2) для случая, когда

$$\mathbf{X} = (x_1, x_2, \dots, x_n) \in D_{\mathbf{X}}, \quad (15.1)$$

где множество $D_{\mathbf{X}}$ не является связным, например может быть конечным либо счетным множеством.

Наиболее распространенными задачами дискретной оптимизации являются целочисленные задачи линейного программирования, т. е. задачи, в которых критерий оптимальности $F(\mathbf{X})$ — линейная функция переменных $x_1, x_2, \dots, x_n$, а на сами переменные $x_1, x_2, \dots, x_n$ либо на часть их наложено дополнительное требование целочисленности. Тогда (15.1) называется условием дискретности либо частичной дискретности [61, 62].

Очевидно, что в задачах дискретной оптимизации область допустимых решений (или область работоспособности) $D_{\mathbf{X}}$ является невыпуклой и несвязной. Обычно при проектировании технических объектов условия дискретности подразделяются по отдельным переменным таким образом, что

$$x_j \in D_j, \quad j = 1, 2, \dots, k, \quad (15.2)$$

где $k \leq n$, а все D_j — конечные (либо счетные) множества, содержащие не менее двух элементов. Если $k = n$, это соответствует полностью дискретной задаче, а если $k < n$ — частично дискретной задаче. Простейшим случаем определения условного экстремума на конечном множестве является случай, когда $k = n$, а все D_j — конечные множества. Решение задач целочисленного программирования, как это ни парадоксально, значительно сложнее решения регулярных (линейных, либо выпуклых) задач обычного математического программирования, так как в силу невыпуклости и несвязности области $D_{\mathbf{X}}$ здесь невозможно применение стандартных приемов математического программирования, т. е. движение по градиенту, перемещение по вершинам многогранника и т. п.

Первыми методами решения задач такого рода являлись методы прямого перебора, однако даже в случае конечности всех множеств D_j при $k = n$ количество точек, в которых надо рассчитать функцию $F(x_1, x_2, \dots, x_n)$, определяется условием $l \geq 2^n$, что при увеличении числа переменных n быстро приводит к нереальным вычислительным затратам. Другой распространенный подход к решению дискретной задачи до разработки специальных методов целочисленного программирования заключался в приведении ее к обычной задаче выпуклого программирования, результаты решения которой в последующем округлялись до ближайших целых численных значений.

Таким образом решается задача оптимизации параметров пассивных элементов (резисторов, конденсаторов) при проектировании схем, несмотря на то, что по техническим причинам эти параметры могут принимать лишь определенные дискретные значения. Такое округление до определенных дискретных значений не приводит к существенным ошибкам лишь при больших значениях x_i , а в области малых значений x_i либо в области узкого крутого оврага (гребня) целевой функции может привести даже к недопустимому решению. Поэтому для решения дискретных задач были разработаны специальные методы оптимизации, рассмотренные в § 15.3.

15.2. КЛАССИФИКАЦИЯ ОСНОВНЫХ ЗАДАЧ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

Несмотря на то, что ситуации, порождающие задачи дискретной оптимизации, могут быть самыми разнообразными, большинство из них сводится к решению следующих основных задач [62].

15.2.1. Транспортные задачи. К этому классу задач относятся задачи, на которых формально требование целочисленности не накладывается, однако на практике они всегда имеют целочисленное решение при любых целочисленных исходных данных. Постановка *транспортной задачи* заключается в следующем. Имеется $i=1, 2, \dots, m$ поставщиков некоторого определенного продукта и $j=1, 2, \dots, n$ пунктов его потребления. Для каждого i -го пункта производства задано a_i — объем производства, а для каждого j -го пункта потребления задано b_j — объем потребления. Также задано c_{ij} — затраты на перевозку единицы продукта от i -го пункта производства до j -го пункта потребления.

Обычно производство и потребление сбалансированы, т. е. соблюдается условие

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (15.3)$$

Требуется составить план перевозок таким образом, чтобы он: а) не превышал предел производительности поставщиков, б) полностью обеспечивал всех потребителей, в) давал минимум суммарных затрат на перевозку.

Если обозначить через x_{ij} объем перевозок от i -го поставщика к j -му потребителю, где

$$x_{ij} \geq 0, \quad (15.4)$$

то условия а) — в) запишутся соответственно как

$$\sum_{j=1}^n x_{ij} = a_i, \quad i = 1, 2, \dots, m, \quad (15.5)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, 2, \dots, n, \quad (15.6)$$

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (15.7)$$

и решение задачи сведется к минимизации (15.7) при учете ограничений (15.4)—(15.6), причем доказано, что это решение всегда имеет целочисленное значение независимо от коэффициентов c_i [61].

Важнейшими частными случаями транспортной задачи являются задачи о назначениях и задачи о потоках в сетях, широко использующиеся на этапе конструкторского проектирования схем в случае, если математические модели представлены в виде графов [3]. Например, задача проектирования печатного монтажа сводится к задаче поиска совокупностей путей графа, не имеющих пересечений, которая в свою очередь решается с помощью транспортной задачи.

15.2.2. Задачи с неделимостями. Как уже отмечалось, одной из наиболее распространенных дискретных задач является задача целочисленного линейного программирования, т. е. такая задача линейного программирования, в которой на все переменные (или часть их) наложено дополнительное требование целочисленности. Наиболее характерной задачей данного типа является широко известная задача о ранце. Задача ставится следующим образом: имеется n предметов ($j=1, 2, \dots, n$) и заданы величины a_j — веса каждого j -го предмета, c_j — ценности каждого j -го предмета. Необходимо ранец грузоподъемностью A загрузить набором предметов, имеющих максимальную суммарную ценность.

Задача математически сводится к задаче максимизации

$$\sum_{j=1}^n c_j x_j \rightarrow \max, \quad (15.8)$$

если переменные x_j введены следующим образом:

$$x_j = \begin{cases} 1, & \text{если } j\text{-й предмет подлежит загрузке,} \\ 0 & \text{в обратном случае, } j=1, 2, \dots, n. \end{cases} \quad (15.9)$$

При этом должно выполняться условие

$$\sum_{j=1}^n a_j x_j \leq A. \quad (15.10)$$

Эта задача может всевозможным образом усложняться (каждый предмет может загружаться в нескольких экземплярах, может быть ограничен не только суммарный вес, но и суммарный объем), но тем не менее она всегда представляет собой целочисленную задачу линейного программирования. К данной задаче может быть приведена задача оптимального выбора и размещения фрагментов интегральных схем с учетом их стоимости, объема или веса, т. е. одна из частных задач проектирования интегральных схем, причем роль ранца может играть, например, заданная площадь подложки или монтажной плоскости.

К этой же задаче приводится задача проектирования топологии БИС на основе библиотечного набора элементов. Задача заключается в таком размещении элементов библиотеки, покрыва-

ющих функциональную схему проектируемой БИС на рабочем поле кристалла, чтобы площадь, занимаемая схемой на кристалле, была бы минимальной. Задача отличается большим числом размещаемых элементов, существенной неоднородностью их размеров и наличием жестких ограничений на размер кристалла.

15.2.3. Задачи комбинаторного типа. В задачах этого типа ищется экстремум некоторой целочисленной функции, заданной на конечном множестве, или сами переменные, принадлежащие этому конечному множеству и приводящие к экстремуму данной целевой функции.

Наиболее общей и распространенной задачей комбинаторного типа является *задача коммивояжера*, которая заключается в следующем. Имеется $n+1$ город, причем номером 0 обозначен исходный город, выехав из которого, коммивояжер должен побывать по одному разу в каждом из n городов и вернуться в исходный город с номером 0. Расстояние между i -м и j -м городом обозначается C_{ij} ; тогда $C = [c_{ij}]$ представляет собой матрицу расстояний между городами. Задача заключается в том, чтобы установить порядок объезда городов, при котором суммарный путь, пройденный коммивояжером, будет минимальным.

Эта задача несколько напоминает транспортную задачу, описанную в п. 15.2.1, здесь речь тоже идет о минимизации суммарного расстояния, но уже не по всем матрицам перестановок, а лишь по матрицам циклических перестановок, что резко усложняет задачу. Наиболее употребимая формулировка данной задачи сводится к следующему виду [62].

Вводятся переменные

$$x_{ij} = \begin{cases} 1, & \text{если коммивояжер из } i\text{-го города} \\ & \text{переезжает в } j\text{-ый город;} \\ 0 & \text{в противном случае;} \end{cases} \quad (15.11)$$

здесь $i, j = 0, 1, 2, \dots, n$. Рассматривается задача минимизации

$$\sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \rightarrow \min \quad (15.12)$$

при условиях

$$\sum_{i=0}^n x_{ij} = 1, \quad j = 1, 2, \dots, n, \quad (15.13)$$

$$\sum_{j=0}^n x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (15.14)$$

$$u_i - u_j + n x_{ij} \leq n - 1, \quad i, j = 1, 2, \dots, n; \quad i \neq j. \quad (15.15)$$

Условия (15.13) свидетельствуют о том, что коммивояжер выезжает из каждого города (кроме конечного) ровно один раз; аналогично (15.14) показывают, что коммивояжер въезжает в каждый город (кроме начального) ровно один раз.

Переменные u_i, u_j — целые неотрицательные переменные, $u_i = p$, если i -й город посещается коммивояжером на p -шаге объезда, $p = 1, 2, \dots, n$. Тогда для всех $x_{ij} = 0$ следует, что $u_i - u_j \leq n - 1$ и условия (15.15) выполняются. При $x_{ij} = 1$ условия (15.15) превращаются в равенства $u_i - u_j + nx_{ij} = p - (p + 1) + n = n - 1$. Условие (15.15) вводится для устранения возможности распада пути коммивояжера на несколько не связанных между собой подциклов.

К данной дискретной задаче приводится, например, задача компоновки какого-либо РЭУ из имеющегося набора интегральных схем или печатных плат с фиксированным составом межсоединений, играющим роль маршрута коммивояжера. В зависимости от того, как расположены отдельные схемы на общей подложке, получится различное расположение межсоединений, причем варианты будут отличаться друг от друга общей длиной межсоединений, которую необходимо минимизировать для получения минимума стоимости и веса устройства, а также уменьшения вероятностей возникновения случайных контактов или паразитных емкостей между межсоединениями.

К комбинаторным задачам относится и задача о покрытиях, общая постановка которой сводится к следующему. Имеется конечное множество $S = \{\sigma_1, \sigma_2, \dots, \sigma_m\}$ и некоторое конечное семейство его подмножеств $S_j, j = 1, 2, \dots, n$. Требуется найти минимальное покрытие множества S , т. е. минимальный набор подмножеств S_j такой, что любой элемент множества S принадлежит хотя бы одному из выделяемых подмножеств. Задача формализуется следующим образом. Вводится матрица инцидентий $A = [a_{ij}]$, где

$$a_{ij} = \begin{cases} 1, & \text{если } \sigma_i \in S_j; \\ 0 & \text{в противном случае.} \end{cases} \quad (15.16)$$

Вводятся булевы переменные $x_j, j = 1, 2, \dots, n$, имеющие смысл

$$x_j = \begin{cases} 1, & \text{если } S_j \text{ войдет в покрытие;} \\ 0 & \text{в противном случае.} \end{cases} \quad (15.17)$$

Нахождение минимального покрытия в данном случае сводится к задаче минимизации

$$\sum_{j=1}^n x_j \rightarrow \min \quad \text{либо} \quad \sum_{j=1}^n c_j x_j \rightarrow \min \quad (15.18)$$

(второе выражение справедливо для взвешенной задачи о покрытии, где c_j — заданные действительные числа) при условии

$$\sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, 2, \dots, m. \quad (15.19)$$

К данной задаче приводится одна из задач конструкторского проектирования, а именно, задача размещения S компонентов на множестве S_j слоев многослойной платы; при этом необходимо минимизировать число слоев.

Одним из примеров комбинаторной задачи является упрощенная задача о наилучшем распределении памяти ЭВМ. Пусть A_{ij} — j -я стандартная подпрограмма для вычисления функции i в библиотеке подпрограмм ($i=1, 2, \dots, m; j=1, 2, \dots, n$). Подпрограмма A_{ij} занимает a_{ij} ячеек памяти и требует для счета t_{ij} секунд. Требуется составить программу A , которая определяется заданием некоторого набора I индексов i , т. е. функций, подлежащих вычислению. При этом следует для составления программы A указать такой набор подпрограмм A_{ij} , чтобы длина всей программы не превосходила M ячеек, а время счета по ней было минимальным.

Как обычно, в таких задачах вводятся переменные

$$x_{ij} = \begin{cases} 1, & \text{если } A_{ij} \text{ включается в } A; \\ 0 & \text{в противном случае.} \end{cases} \quad (15.20)$$

Тогда наша задача сводится к задаче минимизации

$$\sum_{i \in I} \sum_{j=1}^n t_{ij} x_{ij} \rightarrow \min \quad (15.21)$$

при условиях

$$\sum_{j=1}^n x_{ij} = 1, \quad i \in I, \quad \sum_{i \in I} \sum_{j=1}^n a_{ij} x_{ij} \leq M. \quad (15.22)$$

15.2.4. Задачи на невыпуклых и несвязных областях. Эти задачи включают в себя модели, в которых к обычным условиям задачи линейного программирования присоединены некоторые дополнительные условия, превращающие область допустимых решений D_X в невыпуклую или несвязную. В теории дискретной оптимизации разработаны специальные приемы, позволяющие свести эти задачи к частично целочисленным задачам линейного программирования.

Простейшим примером такого рода является введение дополнительных ограничений на переменные x_1 и x_2 , например $x_1 x_2 = 0$, $x_1 + x_2 \geq 1$, которые, будучи присоединены к условиям неотрицательности этих переменных в задаче линейного программирования, дают несвязную область, изображенную на рис. 15.1. Кажется, что добавление к задаче линейного программирования подобных условий превращает ее в задачу нелинейного программирования. Однако даже в простейшем случае (рис. 15.1) видно, что введение этих условий настолько изменяет характер задачи, что стандартные методы нелинейного программирования могут не дать ее решения, тогда как методы дискретного программирования дают эффективные способы ее решения.



Рис. 15.1. Область допустимых решений при введении на переменные x_1 и x_2 ограничений вида $x_1 + x_2 \geq 1$, $x_1 x_2 = 0$

Задачи такого характера возникают при определении оптимальной топологии отдельных ячеек ИС, когда вводятся дополнительные линейные и нелинейные ограничения. При этом линейными ограничениями описываются ограничения на геометрию ячейки и ограничения, обуславливаемые необходимостью соединения отдельных ячеек, а нелинейными ограничениями — заданные соотношения вертикальных и горизонтальных размеров, необходимость размещения выводов на одном уровне и т. д.

15.2.5. Задачи с разрывной целевой функцией. Из задач этого типа наиболее важной и изученной является так называемая транспортная задача с фиксированными доплатами. Она отличается от обычной транспортной задачи, описанной в п. 15.2.1, лишь тем, что минимизируемая функция (15.7) здесь имеет вид

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij}(x_{ij}) \rightarrow \min, \quad (15.23)$$

где

$$c_{ij}(x_{ij}) = \begin{cases} 0, & \text{если } x_{ij} = 0; \\ c_{ij}x_{ij} + d_{ij}, & \text{если } x_{ij} > 0. \end{cases} \quad (15.24)$$

Здесь c_{ij} также интерпретируется как затраты на перевозку груза из пункта i в пункт j , а d_{ij} , например, как плата за аренду транспортных средств, не зависящая от их загрузки. Из-за введения этих дополнительных слагаемых в (15.24) имеем разрывность каждого слагаемого в нуле. В [61] доказывается, что данная задача путем введения дополнительных целочисленных переменных также может быть сведена к частично целочисленной задаче линейного программирования.

15.2.6. Некоторые многоэкстремальные задачи. Впервые возможность сведения нелинейной невыпуклой задачи (т. е. задачи, в которой возможна многоэкстремальность — наличие нескольких локальных экстремумов, не совпадающих с глобальным) к частично целочисленной задаче была показана в [61]. Как и всегда, подобное сведение к частично целочисленной задаче приводит к существенному увеличению числа переменных и ограничений, однако, так как численные методы решения задач целочисленного программирования уже достаточно хорошо разработаны, такой способ открывает один из реальных путей фактического решения многоэкстремальных задач, которые часто возникают при проектировании электронных схем.

15.3. МЕТОДЫ ДИСКРЕТНОЙ ОПТИМИЗАЦИИ

В настоящее время существует множество разнообразных численных методов решения дискретных задач. Выделим основные группы методов, принципиально отличающихся по подходу к решению задачи, и дадим их общую характеристику.

15.3.1. Методы отсеечения. Эта группа методов основана на «регуляризации» задачи, достигаемой за счет временного отбрасыва-

ния условий дискретности, т. е. сведения задачи целочисленного линейного программирования к задаче линейного программирования. К получившейся регулярной задаче применяются широко изученные стандартные методы линейного программирования, например симплекс-метод. В дальнейшем требуется переход от полученного решения к целочисленному решению, который, как указывалось, не может быть достигнут простым округлением компонентов вектора полученного нецелочисленного решения.

Для общей задачи целочисленного линейного программирования этот переход заключается в следующем. Если в результате первого шага (получения оптимума без учета требований целочисленности) получено нецелочисленное решение, то к ограничениям исходной задачи добавляется новое линейное ограничение, обладающее двумя свойствами:

- 1) полученное нецелочисленное решение ему не удовлетворяет;
- 2) любое целочисленное решение ему заведомо удовлетворяет.

Затем решается полученная расширенная задача (второй шаг), в случае получения опять нецелочисленного решения добавляется еще одно ограничение и т. д. Процесс повторяется до получения решения, обладающего нужными свойствами целочисленности. Таким образом, решение целочисленной задачи линейного программирования сводится к решению некоторой последовательности обычных задач линейного программирования.

Геометрически добавление каждого такого линейного ограничения соответствует проведению гиперплоскости, отсекающей от многогранника решений «регуляризованной» задачи старую оптимальную точку с дробными координатами, но не затрагивающей ни одной из целочисленных точек этого многогранника. Поэтому методы, использующие эту идею, получили название «методов отсекающих плоскостей» или «методов отсечения».

С помощью методов отсечения решаются задачи средней размерности (не более сотни переменных и десятка ограничений). При этом далеко не все задачи целочисленного линейного программирования поддаются решению с помощью указанных методов.

15.3.2. Методы ветвей и границ. Методы ветвей и границ относятся к группе комбинаторных методов, основную идею которых составляет замена полного перебора всех решений их частичным перебором, что осуществляется путем отбрасывания некоторых подмножеств вариантов, т. е. допустимых решений, заведомо не дающих оптимума; перебор при этом ведется лишь среди остающихся вариантов, являющихся в определенном смысле «перспективными».

Таким образом, основная идея всех методов ветвей и границ при всем их разнообразии базируется на использовании конечности множества вариантов и переходе от полного перебора к сокращенному (направленному) перебору. Важную роль при этом играет оценка и отбрасывание неперспективных множеств вариантов,

Рассмотрим задачу дискретного программирования в следующей общей форме. Минимизировать

$$Z = f(\mathbf{X}) \quad (15.25)$$

при условии

$$\mathbf{X} \in G, \quad (15.26)$$

где G — некоторое конечное множество.

В основе методов ветвей и границ лежат следующие основные алгоритмы, позволяющие для достаточно широкого класса целевых функций и ограничений существенно уменьшить объем перебора.

1. Вычисление нижней границы (оценки) значений целевой функции $f(\mathbf{X})$ на исходном множестве G (или на некотором его подмножестве), т. е. такого $\varphi(G)$, что для $\mathbf{X} \in G$ имеет место $f(\mathbf{X}) \geq \varphi(G)$.

2. Последовательное разбиение (или ветвление) множества решений G на постепенно уменьшающиеся подмножества, т. е. образование дерева решений. Правила ветвления для каждого метода специфичны и часто основаны на эвристических методах.

3. Пересчет оценок. Разбивая в процессе решения исходное множество G на подмножества $G_1, G_2, \dots, G_s$, всегда будем считать, что оценка (граница) для любого подмножества G_i не меньше оценки для множества G : $\varphi(G_i) \geq \varphi(G)$, $i = 1, 2, \dots, s$.

В конкретных ситуациях часто оказывается возможным получить хотя бы для некоторых i строгое неравенство $\varphi(G_i) > \varphi(G)$ благодаря этому отсесть бесперспективные варианты, для которых $f(\mathbf{X}) > \varphi(G_i)$.

4. Вычисление допустимых решений. Исходя из специфики конкретной задачи указываются различные способы нахождения допустимых решений в последовательно разветвляемых подмножествах и отбрасывания бесперспективных вариантов. В частности, допустим, что выбор минимальной оценки $\varphi(G_v) = \min_{1 \leq i \leq s} \{\varphi(G_i)\}$ может однозначно определять подмножество G_v , предназначенное для дальнейших исследований.

5. Формирование признака оптимальности (основанного на использовании оценок). Если допустимое решение $\mathbf{X}$ принадлежит некоторому подмножеству G_v и при этом

$$f(\mathbf{X}) = \varphi(G_v) \leq \varphi(G_i), \quad i = 1, 2, \dots, s,$$

то $\mathbf{X}$ — оптимальное решение задачи на данном этапе. Как отмечалось, именно этот признак применяется обычно на некотором этапе ветвления.

В результате применения рассмотренных алгоритмов происходит постепенное сужение области поиска. Метод ветвей и границ хорошо зарекомендовал себя в решениях целочисленных линейных задач, ряда задач распределения ресурсов, однако в общем случае его применение может привести к отрицательным результатам в смысле экономии вычислительных затрат.

15.3.3. Аддитивные методы. Аддитивные методы основаны на близких к идеям методов «ветвей и границ» приемах и базируются на алгоритмах Балаша [62] и их модификациях для решения задач целочисленного линейного программирования с булевыми переменными. Задача в канонической матричной форме записывается в виде: минимизировать

$$f = CX \quad (15.27)$$

при условиях

$$AX + Y = B, \quad (15.28)$$

$$x_j = \begin{cases} 0, \\ 1, \end{cases} j \in N, \quad (15.29)$$

$$Y \geq 0, \quad (15.30)$$

где Y — m -мерный вектор свободных переменных.

В данном алгоритме пробным планом называется любой $(n+m)$ -мерный вектор $U(X, Y)$, удовлетворяющий (15.28), (15.29). Пробный план называется планом, если он удовлетворяет также и (15.30) и план называется оптимальным, если он удовлетворяет (15.27), т. е. минимизирует $f = CX$. Сформулированная выше задача имеет 2^n пробных планов. Аддитивный алгоритм относится к методам частичного перебора, и основная его идея заключается в получении оптимального плана в результате рассмотрения не всех 2^n пробных планов, а лишь некоторого их подмножества. Это осуществляется также при помощи схемы ветвления.

Для пояснения сущности метода вводятся следующие понятия. Частичным q -планом называется набор, состоящий из q фиксированных допустимых переменных (0 или 1). Под дополнением частичного q -плана понимается набор допустимых значений для остальных $n-q$ переменных. Если оказывается, что некоторый конкретный частичный q -план не имеет дополнения, приводящего к меньшему значению целевой функции f , то из рассмотрения можно исключить 2^{n-q} допустимых планов, продолжив поиск среди остальных.

Таким образом, аддитивный алгоритм заключается в построении последовательности частичных планов ($q=0, q=1, q=2$ и т. д.) и в исключении некоторых подмножеств их дополнений. Такое исключение производится в тех случаях, когда данное подмножество не имеет ни одного дополнения, которое давало бы значение f , не лучшее ранее достигнутого.

В [61] имеется указание на то, что алгоритм работает эффективно для задач среднего размера (до 30 переменных), но допускает значительный рост числа ограничений.

15.3.4. Метод динамического программирования. Возможности приложения идей метода динамического программирования к некоторым дискретным задачам легко просматриваются. В частности, они имеют приложение для решения задач о ранце, о коммивояжере, а также для задач теории расписаний [61].

В качестве примера рассмотрим один из частных случаев теории расписаний и применение к нему алгоритма Джонсона, основанного на методах динамического программирования. Допустим, имеются два сборщика и n схем, каждая из которых должна собираться сначала первым, а затем вторым сборщиком. Время работы над i -й схемой ($i=1, 2, \dots, n$) для первого сборщика составляет a_i , а для второго — b_i . Требуется найти такое расписание, т. е. так составить последовательность сборки схем, чтобы минимизировать общее время сборки всех схем. При этом полагается, что последовательности сборки схем первым и вторым сборщиком совпадают. Искомое расписание тогда представляет собой перестановку $(i_1, i_2, \dots, i_n)$ чисел $1, 2, \dots, n$, минимизирующую время от начала сборки первой схемы i_1 первым сборщиком до окончания сборки последней схемы i_n вторым сборщиком.

Идея получения оптимального расписания согласно алгоритму Джонсона состоит в стремлении максимально сократить простои второго сборщика. На первом шаге алгоритма среди чисел $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n)$ находится наименьшее $t_{\min 1}$ (1 — номер первого шага). Если таких чисел несколько, берется произвольное из них. Если $t_{\min 1} = a_i$, то полагаем $i = i_1$, т. е. помещаем i -ю схему на первое место в расписании, если же $t_{\min 1} = b_i$, то полагаем $i = i_n$, т. е. помещаем i -ю схему на последнее место в расписании. Затем вычеркиваем i -ю схему (т. е. числа a_i и b_i) из списка. Таким образом, перебираем все числа в списке $(a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n)$ и каждый раз ставим соответствующую i -ю схему на первое из мест, еще не занятых на предыдущих $k-1$ шагах, если на k -м шаге $t_{\min k} = a_i$, или на последнее из незанятых мест в расписании, если $t_{\min k} = b_i$. Последнюю оставшуюся схему помещаем на единственное оставшееся в расписании место.

Таблица 15.1 иллюстрирует рассмотренный алгоритм для случая $n=7$ схем. В каждой строке таблицы соответствующее число $t_{\min k}$ отмечено звездочкой.

Таким образом, оптимальное расписание есть $(i_1, i_2, i_3, i_4, i_5, i_6, i_7) = (2, 3, 1, 4, 7, 6, 5)$. Минимальное время сборки составляет 109 мин, из которых второй сборщик тратит на ожидание 12 мин. Напомним, что прямой перебор потребовал бы сравнения $7! = 5040$ перестановок.

Таблица 15.1

Номер схемы i	1	2	3	4	5	6	7
a_i	16*	6*	8*	20	14	14	20
b_i	18	10	12	18*	8*	10*	21
Номер шага	V	I	III	VI	II	IV	VII
Место схемы в расписании	3	1	2	4	7	6	5

15.3.5. Методы случайного поиска. Эти методы относятся к группе приближенных методов. Они получили свое развитие вследствие сложности описанных выше точных методов и трудности их реализации для задач большой размерности. Для задач дискретного программирования нередко также применяется сочетание случайного поиска с локальной оптимизацией.

Чаще всего случайные методы, как и все приближенные методы, включают в себя эвристические приемы, максимально использующие специфику задачи.

15.4. ЗАДАЧИ И МЕТОДЫ СИНТЕЗА В САПР

В существующих в настоящее время САПР алгоритмы и программы синтеза занимают незначительную часть по сравнению с алгоритмами и программами расчета, анализа и оптимизации схем. Объясняется это в основном двумя причинами: трудностями формализации методов синтеза, в большинстве случаев основанных на эвристических приемах, применимых лишь для конкретных частных случаев синтеза, и огромными вычислительными затратами, необходимыми для успешного решения задач синтеза. Однако в последнее время в связи с успешной разработкой эффективных алгоритмов оптимизации, существенно сокращающих затраты на поиск, и расширением взаимодействия разработчиков с ЭВМ, появилась возможность автоматизации некоторого класса задач синтеза, что привело к разработке методов и алгоритмов автоматизированного синтеза.

15.4.1. Классификация задач синтеза. Задачи синтеза решаются обычно в два этапа, первый из которых направлен на создание структуры проектируемого объекта и называется структурным синтезом, а второй — на поиск параметров спроектированной структуры, при которых объект будет функционировать при действии всех дестабилизирующих факторов, и называется параметрическим синтезом.

Структурный синтез является неизмеримо более сложной задачей, чем параметрический синтез, он занимает в общем процессе проектирования объектов различного рода совершенно особое место и выполняется в первую очередь.

Структурный синтез включает в себя три основных этапа: генерацию или поиск варианта структуры, оценку сгенерированного или найденного варианта и принятие решения о его пригодности или непригодности и о продолжении или прекращении дальнейшего поиска вариантов.

Деятельность человека на этапе структурного синтеза носит творческий характер, получаемые решения во многих случаях рассматриваются как изобретения. Объясняется это чрезвычайным разнообразием структур как таковых, достаточно неопределенным объемом учитываемых исходных данных и факторов, большим объемом эвристики.

Отличительной чертой структурного синтеза является его дискретный характер, поэтому дискретная оптимизация является содержанием любого структурного синтеза. Значительное развитие методов дискретной оптимизации позволяет в настоящее время надеяться на успешное применение методов автоматизации для задач структурного синтеза.

Параметрический синтез имеет своей целью определение области работоспособности в пространстве внутренних и выходных параметров, в которой объект будет правильно функционировать в течение требуемого срока при наличии различного рода дестабилизирующих факторов. Параметрический синтез, таким образом, дает возможность уточнить и окончательно сформулировать техническое задание на объект, которое до этого носило эскизный характер. Поиск же оптимальных параметров найденной структуры в данной области работоспособности с формализованными критериями оптимальности и ограничениями является уже задачей следующего этапа проектирования — параметрической оптимизации.

15.4.2. Методы параметрического синтеза. Параметрический синтез объектов с непрерывно изменяющимися параметрами базируется на методах, являющихся основой также для параметрической оптимизации и подробно рассмотренных в гл. 14. Основные различия этих двух процессов проектирования заключаются в том, что в процессе параметрического синтеза объект является еще плохо изученным, нет достоверной информации о виде целевой функции, ее дифференцируемости, наличии и виде ограничений на внутренние и выходные параметры, характере взаимосвязей между параметрами. Все эти причины чрезвычайно затрудняют выбор эффективного метода оптимизации и подбор правильных параметров этого метода, например начальной точки и начального шага поиска. В силу этого для параметрического непрерывного синтеза наиболее пригодны методы случайного поиска, а среди детерминированных — методы нулевого порядка, не требующие вычисления производных целевой функции, являющиеся наименее критичными к выбору параметров и позволяющие достаточно эффективно находить если не оптимальное, то вполне приемлемое решение с реальными вычислительными затратами.

В том случае, если не все элементы проектируемого объекта характеризуются непрерывными параметрами, применение непрерывных методов оптимизации для параметрического синтеза становится невозможным. Как уже отмечалось, при проектировании РЭУ часто надо учитывать дискретные шкалы разрешенных номиналов, например для резисторов и конденсаторов, разрешенный набор геометрических размеров активных элементов интегральных схем и т. д. Задачи параметрического синтеза такого рода решаются методами дискретного математического программирования, наиболее распространенные из которых рассмотрены в § 15.3.

Если часть параметров объекта является непрерывной, а часть носит дискретный характер, задача параметрического синтеза решается методами частично дискретного программирования. Зада-

чи дискретного, а тем более частично дискретного программирования неизмеримо сложнее задач непрерывного математического программирования, вследствие чего задачи параметрического дискретного синтеза требуют для своего решения более сложных программ, основанных на эвристических методах, больших вычислительных затрат и труднее поддаются формализации. Методы решения таких задач из-за дискретности внутренних или выходных параметров по своей сложности и математическому аппарату приближаются к методам решения задач структурного синтеза.

15.4.3. Задачи структурного дискретного синтеза, их классификация по уровню сложности. Для решения задач структурного синтеза необходимо:

1. Формализовать структурный базис, т. е. определить множество типовых структурных элементов, на конкретном наборе которых может быть определена проектируемая структура. Например, структурным базисом для простейших РЭУ типа преобразователей, переключателей, ограничителей и т. д. могут быть резисторы, конденсаторы, катушки индуктивности, транзисторы, диоды и т. д. Описание элементов структурного базиса представляет собой модели элементов, используемые для формирования математической модели синтезируемой структуры.

2. Формализовать структурные связи, т. е. определить ограничения на возможности соединений элементов структурного базиса в синтезируемой структуре. Результат формализации представляется в виде или ориентированного графа, ребра которого представляют разрешенные соединения, или матрицы допустимых связей, или списков.

3. Формализовать структурную или целевую функцию. Структурные функции в качестве своих переменных помимо структурных элементов могут иметь целую систему функций-ограничений, связанных со стоимостью структуры или затратами на ее функционирование, которые могут находиться в сложной нелинейной зависимости от структуры, что сильно затрудняет формализацию данного этапа. Если обычные целевые функции при схемотехническом проектировании формализуются с помощью обобщенных критериев оптимальности, рассмотренных в гл. 13, то структурные функции можно описать с помощью некоторого исчисления (алгебры), основанного на определенном алфавите. При использовании операций над словами данного алфавита получим в явном виде зависимость от вида структуры.

4. Формализовать исходное задание на структуру. Этот процесс существенно зависит от проведения трех предыдущих этапов формализации, а также используемого метода синтеза. Чем более совершенен метод синтеза, тем требуется меньший объем предварительной разработки исходного задания.

По уровню сложности все задачи структурного синтеза можно подразделить на несколько групп [2].

К первой группе относятся наиболее простые задачи, в которых выбор структуры либо однозначен, либо производится из ко-

нечного множества вариантов, поддающихся полному просмотру и сравнению за приемлемое время. Полный перечень вариантов структуры в данном случае задается в информационном обеспечении САПР в виде каталогов типовых вариантов, например перечень типовых ИС, используемых в подсистеме проектирования для синтеза какого-либо РЭУ.

Ко второй группе относятся задачи, для которых невозможен прямой перебор вариантов. Синтез таких задач требует либо понижения уровня сложности еще на стадии формирования исходных данных, что влечет за собой исключение возможности получения принципиально нового варианта структуры, либо применения методов направленного перебора. Примером задач второй группы сложности служит задача компоновки m элементов по n конструктивным блокам РЭУ. Здесь типичное значение m — сотни и тысячи, а n — десятки, причем элементы могут иметь различные типы и размеры, а распределение может носить неравномерный характер.

II, наконец, к третьей группе сложности относятся задачи, основанные на применении теории эвристических решений, используемой в случаях, когда исходной информации недостаточно для строгого решения. Такие методы плохо поддаются формализации и поэтому пока не нашли широкого практического распространения в САПР. Поэтому для решения задач третьей группы необходимо активное участие разработчика в процессе проектирования, которое реализуется в диалоговом режиме обмена информацией между человеком и ЭВМ. Примером задачи третьей группы сложности может являться проектирование функциональной схемы РЭУ, перечень используемых элементов которого может быть весьма обширен, а также возможно бесчисленное множество вариантов устройства, различающихся между собой количеством элементов и их структурными связями друг с другом.

Для облегчения синтеза задач третьей группы сложности пытаются формализовать используемые в процессе творчества эвристические приемы [2, 3], которые могут быть положены в основу каталога творческих приемов для дальнейшего их применения в САПР. Это — приемы разделения объекта на некоторые части, совмещения в одно целое отдельных элементов объекта, переноса решения из другой области техники, упрощения элементов или их соединения друг с другом, изменения условий работы и окружающей среды, приемы количественных изменений и т. д.

Помимо понижения уровня сложности задачи при структурном синтезе и уменьшения числа рассматриваемых вариантов за счет целенаправленного перебора еще одним из подходов к решению задач синтеза является использование упрощенных математических моделей объектов, позволяющих ускорить анализ объекта на этапе рассматриваемого варианта структурного синтеза.

15.4.4. Методы структурного дискретного синтеза. Для строгого решения задачи структурного дискретного синтеза необходим полный перебор всех вариантов проектируемой структуры, однако это

возможно лишь для задач, относящихся к первой группе сложности. Для более сложных задач следует сократить число проб при переборе, что осуществляется благодаря иерархическому подходу к дискретному синтезу. Иерархический подход заключается в разбиении всего сложного процесса синтеза на несколько уровней, на каждом из которых синтезируется определенный ранг сложной системы. Например, при синтезе РЭУ на I уровне синтезируется структурная схема, на II уровне — функциональные связи между структурными блоками, на III уровне — функциональная схема и конструкции блоков, на IV уровне — наиболее мелкие функциональные и конструктивные единицы — элементы.

На рис. 15.2 иерархическая структура процесса синтеза представлена в виде дерева решений, в котором на каждом из уровней

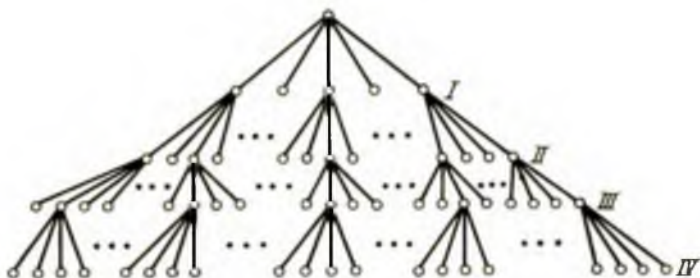


Рис. 15.2. Иерархическая структура процесса синтеза в виде дерева решений (цифрами обозначены уровни проектирования)

проектирования, обозначенных римскими цифрами, выбирается один (или несколько) наилучших из всех возможных вариантов. Далее на каждом из последующих уровней просматриваются лишь выбранные на предыдущем уровне варианты, а все остальные отсекаются как бесперспективные. Для отбрасывания бесперспективных вариантов и выбора наилучших на каждом уровне проектирования используются прямые или косвенные критерии отбора: либо длины проводников в каждом из вариантов, либо стоимости вариантов, либо более сложные целевые функции, формирующиеся на основе решения системы дифференциальных уравнений математической модели устройства. В каждом конкретном случае проектирования большую роль при формировании критерия играет опыт и интуиция проектировщика.

Наиболее эффективным методом, применяющимся при решении задач дискретного структурного синтеза, является рассмотренный выше метод ветвей и границ, который органически приспособлен для решения задач иерархического вида. Другим распространенным подходом к решению задач структурного дискретного синтеза являются последовательные эволюционные алгоритмы, основанные на постепенном последовательном наращивании структуры от первого основного элемента добавлением новых эле-

ментов. Последовательные методы относятся к классу эвристических, требуют активного участия человека (обычно в форме диалогового режима общения с ЭВМ), как правило, не приводят к оптимальной структуре, но зато более экономичны, так как не требуют многократного просчета многих вариантов структуры.

Примерами задач, решаемых в процессе проектирования с помощью последовательных алгоритмов, являются задачи трассировки, размещения и компоновки РЭУ.

15.4.5. Пример структурного дискретного синтеза [61]. Под схемой в данном случае будем понимать некоторое устройство с p булевыми входами и одним булевым выходом. Требования к этой схеме предьявляются в виде списка, сопоставляющего каждой из 2^p возможных комбинаций 0 или 1 на входе соответствующее значение (0 или 1) на выходе. Тем самым указанный список задает некоторую булеву функцию. Такой список называется таблицей истинности логической функции.

Задачей синтеза является: 1) построение схем, реализующих данную булеву функцию; 2) нахождение среди всех схем наиболее экономичной. Таким образом, задача сводится к определению допустимого и оптимального решения в терминах математического программирования. Способы решения первой из этих проблем известны достаточно давно, подход к решению второй может быть дан в терминах дискретного линейного программирования.

Набор конструктивных элементов, из которых синтезируется данная схема, не будет представлять для нас интереса в данном случае — это могут быть интегральные схемы, полупроводниковые приборы, печатные платы и т. п. Важно лишь то, что каждый элемент может находиться в одном из двух состояний: 0 или 1. Элементы можно соединять в блоки двух типов: а) последовательные (или блоки типа И), дающие на выходе 1 только в том случае, если на все входы поданы 1; б) параллельные (или блоки типа ИЛИ), дающие на выходе 1, если хотя бы на один из входов подается 1. Ограничимся рассмотрением схем, состоящих из нескольких блоков типа И, соединенных в один блок типа ИЛИ.

Задача синтеза схемы с данными свойствами эквивалентна задаче аналитического представления булевой функции, заданной таблицей, в виде суперпозиции булевых операций над ее переменными. Поскольку мы ограничиваемся схемами описанного выше вида, это аналитическое выражение должно представлять собой булеву сумму членов, каждый из которых является произведением булевых переменных и их дополнений.

Пусть задана таблица истинности булевой функции $F = F(\xi_1, \xi_2, \dots, \xi_p)$, в которой каждому из 2^p векторов размерности p , состоящих из 0 и 1, сопоставляется 0 или 1. Известный способ построения аналитического выражения заключается в следующем. В таблице булевой функции F отмечаются все строки i , в которых $F=1$. Пусть $(\xi_1^0, \xi_2^0, \dots, \xi_p^0)$ — некоторый (i -й) набор, для которого $F(\xi_1^0, \xi_2^0, \dots, \xi_p^0)=1$. Для этого набора образуется произведение, называемое фундаментальным:

$$P_i = \prod_{k=1}^p \hat{\xi}_k, \text{ где}$$

$$\hat{\xi}_k = \begin{cases} \xi_k, & \text{если } \xi_k^0 = 1; \\ 1 - \xi_k, & \text{если } \xi_k^0 = 0. \end{cases} \quad (15.31)$$

Аналитическое выражение представляет собой булеву сумму из всех таких произведений:

$$F = \sum_{i=1}^m P_i. \quad (15.32)$$

Для решения второй проблемы синтеза, т. е. нахождения наилучшего в определенном смысле аналитического выражения, вводится еще одно важное понятие. Импликантом функции F называется произведение вида

$$I = \prod_{k \in s} \xi_k, \quad (15.33)$$

где s — некоторое подмножество индексов $\{1, 2, \dots, p\}$, обладающее тем свойством, что из $I=1$ следует $F=1$ и такое, что при любом $s' \subset s$ произведение

$\prod_{k \in s'} \xi_k$ (т. е. любой собственный подмножитель (15.33) этим свойством уже не обладает), т. е. импликант — произведение переменных и их дополнений, составленное из минимального числа сомножителей и равное 1 только в случае $F=1$. Наилучшее представление будем понимать как булеву сумму импликантов, содержащую минимальное число слагаемых.

Данная задача оптимального синтеза схемы формулируется в терминах дискретного линейного программирования в виде рассмотренной выше задачи о покрытии. Пусть $P_1, P_2, \dots, P_m$ — фундаментальные произведения функции F , а $I_1, I_2, \dots, I_n$ — ее импликанты. Положим

$$a_{ij} = \begin{cases} 1, & \text{если } I_j \text{ является подмножителем } P_i; \\ 0 & \text{в противном случае.} \end{cases} \quad (15.34)$$

Иначе говоря, $a_{ij}=1$ означает, что из $P_i=1$ следует $I_j=1$. Как обычно, введем переменные

$$x_j = \begin{cases} 1, & \text{если } j\text{-й импликант войдет в представление;} \\ 0 & \text{в противном случае.} \end{cases} \quad (15.35)$$

Требуется найти набор с минимальным числом импликантов. Это значит, что следует минимизировать функцию вида (15.18) при условии, что все единичные значения функции F будут при этом обеспечены. Это требование выражается неравенствами, вошедшими в (15.17) и (15.19). Таким образом, мы действительно пришли к постановке задачи о покрытии.

Можно также считать, что каждому импликанту I_j приписан определенный «вес» c_j (который может отражать, например, затраты на техническую реализацию соответствующего этому импликанту блока). В этом случае можно говорить о нахождении схемы с минимальным суммарным «весом». Очевидно, что это требование описывается взвешенной задачей о покрытии (15.17)—(15.19).

Проиллюстрируем данную задачу следующим примером. Пусть функция F от четырех булевых переменных ($p=4$) описывается табл. 15.2. Здесь $m=4$. Таким образом представление (15.32) будет иметь вид:

$$F = (1 - \xi_1) \xi_2 (1 - \xi_3) \xi_4 + (1 - \xi_1) \xi_2 \xi_3 \xi_4 + \xi_1 \xi_2 (1 - \xi_3) (1 - \xi_4) + \xi_1 \xi_2 (1 - \xi_3) \xi_4. \quad (15.32a)$$

При этом импликанты

$$I_1 = (1 - \xi_1) \xi_2 \xi_4, I_2 = \xi_2 (1 - \xi_3) \xi_4, I_3 = \xi_1 \xi_2 (1 - \xi_3), n = 3.$$

Тогда по правилу (15.34) можно составить матрицу размером 4×3

$$[a_{ij}] = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{vmatrix} \quad (15.34a)$$

и задача о покрытии приобретает вид: минимизировать $x_1 + x_2 + x_3$ при условиях

$$\begin{aligned} x_1 + x_2 &\geq 1, \\ x_1 &\geq 1, \\ x_3 &\geq 1, \\ x_2 + x_3 &\geq 1, \end{aligned} \quad \text{где } x_1, x_2, x_3 = \begin{cases} 0, \\ 1. \end{cases}$$

Оптимальное решение данной задачи сразу же очевидно, так как из ограничений следует, что $x_1 = 1$, $x_3 = 1$, после чего необходимо положить $x_2 = 0$. Таким образом, в минимальное представление функции F войдут первый и третий импликанты, т. е.

$$F_{\text{онт}} = (1 - \xi_1) \xi_2 \xi_4 + \xi_1 \xi_2 (1 - \xi_3).$$

Таблица 15.2

ξ_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
ξ_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
ξ_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
ξ_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
F	0	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0

ГЛАВА 16

КОНСТРУКТОРСКОЕ ПРОЕКТИРОВАНИЕ

16.1. ПОСТАНОВКА ЗАДАЧИ

На этапе конструкторского проектирования РЭА функционально-логическая или принципиальная электрическая схема преобразуется в совокупность конструктивных узлов, которые осуществляют ее физическую реализацию. При этом необходимо рассматривать размеры ИС, транзисторов, резисторов, конденсаторов

диокомпонентов, размеры проводников и расстояния между ними, размеры сменных блоков, число контактов разъема блока, вид монтажа для проведения соединений. Следует обеспечить заданные требования к надежности устройства в заданных физических и климатических условиях, возможность ремонта, приемлемую стоимость изготовления, уложиться в заданные габариты и весовые характеристики.

Основным принципом конструирования РЭУ является модульность построения устройств, причем модульная структура имеет, как правило, иерархический характер. Наиболее ярко это проявляется при конструировании ЭВА, имеющей относительно высокую степень унификации схемных решений. Модули различного уровня сложности объединяются в устройства в соответствии с их функциональной иерархией (рис. 16.1).

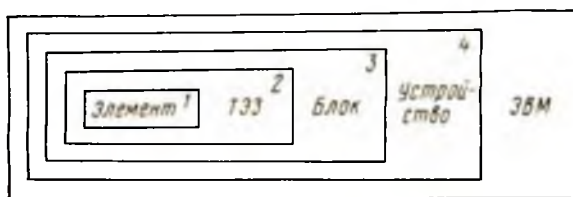


Рис. 16.1. Иерархия конструктивных модулей

Модуль уровня 1 представляет собой или ИС, содержащую несколько логических элементов типа И, ИЛИ, НЕ, или фрагмент БИС — триггер, регистр. Модуль уровня 2 объединяет на одной печатной плате (обычно двусторонней) несколько десятков ИС или несколько БИС и является типовым элементом замены (ТЭЗ), который используется для оперативной замены при возникновении неисправностей в устройстве. Модуль уровня 3 объединяет ТЭЗы в блоки, которые конструктивно могут быть оформлены в виде кассеты с панелью, имеющей проводный или печатный монтаж. Модуль уровня 4 представляет отдельное устройство и содержит ряд кассет, конструктивно объединяемых в стойку или шкаф, в котором межпанельные соединения осуществляются жгутовым монтажом.

Соединения в РЭУ в значительной степени определяют их основные технические параметры, такие как вес, объем, надежность, быстродействие. Например, при переходе от узлов низшего конструктивного уровня к высшему (ИС — печатные платы — панели) происходят значительные потери плотности компоновки элементов, что увеличивает время распространения электрических сигналов.

Системы автоматизированного конструкторского (технического) проектирования в первую очередь были реализованы для про-

ектирования двухслойных и многослойных печатных плат для выпуска технической документации, в состав которой входят электрические схемы и сборочные чертежи, таблицы цепей идентификации, таблицы, технологическая документация на монтажных носителях. Технологическая документация в виде, например, перфокарт представляет собой управляющие программы автоматов для изготовления фотошаблонов, сборки, пайки, сверления монтажных отверстий, контроля монтажа, а затем и тестового контроля логического функционирования ячеек.

При техническом проектировании предпочтение отдается автоматическим системам, когда по исходному описанию схемы обеспечивается проектирование и выпуск документации без участия человека. Это значительно снижает вероятность внесения ошибок и повышает производительность труда проектировщиков, так как ЭВМ передаются утомительные рутинные операции.

Типовые в содержательном плане задачи конструирования РЭУ, которые требуется решать для устройств как различного уровня сложности, так и различной степени интеграции, имеют общую природу с точки зрения автоматизации проектирования. Они решаются с использованием общего семейства алгоритмов, при конкретной реализации которых на ЭВМ в САПР учитывается применяемая технология изготовления устройств. Такими типовыми задачами являются компоновка блоков, размещение компонентов и трассировка монтажных соединений. Эти задачи являются оптимизационными [23, 63—65].

Среди алгоритмов конструкторского проектирования выделяют два основных класса: конструктивные и итерационные.

Конструктивные алгоритмы формируют проектное решение за ряд последовательных шагов: выбирается один элемент схемы рассматриваемого уровня, к нему по определенным правилам присоединяется второй, к полученному комплексу элементов — третий и т. д. Алгоритмы, использующие подобную методологию, называются последовательными. Алгоритмы, в которых формируется несколько групп элементов в пределах одного шага, называются параллельными. Разработаны также комбинированные последовательно-параллельные конструктивные алгоритмы.

Итерационные алгоритмы, в отличие от конструктивных, требуют задания начального приближенного решения задачи конструкторского проектирования, которое затем улучшается. Начальное решение задается конструктором-проектировщиком или получается на ЭВМ как результат работы конструктивного алгоритма. Примером итерационного алгоритма является процедура парных перестановок, состоящая в том, что делается попытка попарно поменять местами друг с другом все элементы конструкции с целью улучшения проектировочного решения на основании максимизации или минимизации заданной функции критерия качества. Аналогична процедура групповых перестановок, в которой производится обмен группами элементов. Основанные на этой процедуре алгоритмы являются параллельно-итерационными.

Задачи конструкторского проектирования имеют по своему существу комбинаторный характер. Проектируемая РЭА при детализации ее до уровня радиокомпонентов имеет число элементов $N=10^2 \dots 10^7$. Поэтому решение комбинаторных задач оптимального проектирования методами, использующими прямой перебор $N!$ вариантов, практически неприемлемо.

С целью сокращения временных затрат, которые в реальных задачах конструкторского проектирования очень велики, используются различные способы сокращения перебора вариантов. Вследствие этого в автоматизированном конструкторском проектировании наибольшее распространение получили эвристические алгоритмы, предназначенные для специализированных методов, учитывающих, как правило, особенности технологии изготовления схем. В эвристических алгоритмах также используются конструктивные и итерационные процедуры.

Очевидно, что само выделение трех этапов конструкторского проектирования — *компоновки, размещения и трассировки*, — направлено на снижение размерности общей задачи. Декомпозиция задачи производится таким образом, что сначала осуществляется компоновка путем оптимальной группировки функциональных узлов, затем размещение компонентов выделенных узлов в их монтажном пространстве с учетом критериев, отражающих оптимальность последующего этапа — трассировки межсоединений. Последняя задача конструкторского проектирования — трассировка — является практически наиболее важной, но до настоящего времени она не имеет полностью автоматического решения. После работы программ автоматической трассировки остается порядка 5 ... 10% неразведенных соединений, которые затем дорабатываются конструкторами.

16.2. МОДЕЛИ КОНСТРУКЦИИ И СХЕМ

Наиболее общей моделью конструкции является монтажное пространство, размеры которого соответствуют габаритным размерам схемы k -го уровня иерархии.

Математическая модель монтажного пространства, как правило, дискретна. В этом случае она представляет собой описание координатной сетки с равномерным или неравномерным шагом (рис. 16.2). Возможные позиции элементов характеризуются целочисленными индексами, а не действительными значениями координат, что удобно для алгорит-

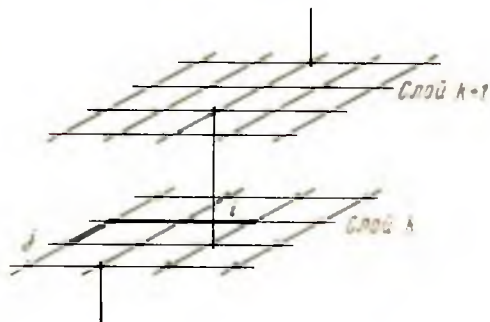


Рис. 16.2. Регулярное монтажное пространство с ортогональной метрикой

мизации задач конструирования. Обычно проектируются и реализуются двухслойные и многослойные конструктивные элементы, причем каждый слой имеет свой номер. Поэтому для описания местоположения элемента в слое достаточно двух индексов. При этом расстояние определяется в относительных единицах. Примерами физического представления плоского монтажного пространства служат печатная плата, подложка микросборки, слой кристалла БИС.

Поскольку задачи конструкторского проектирования практически всегда связаны с оценкой длины соединений, то в монтажном пространстве задаются различные метрики. Например, расстояние d_{ij} между i -м и j -м компонентами схемы определяют следующими способами:

$$\left. \begin{aligned} d_{ij}^{(1)} &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ d_{ij}^{(2)} &= |x_j - x_i| + |y_j - y_i| \\ d_{ij}^{(3)} &= |x_j - x_i|^s + |y_j - y_i|^s \end{aligned} \right\} \quad (16.1)$$

Первый способ дает оценку длины соединения между двумя точками с координатами (x_i, y_i) , (x_j, y_j) по кратчайшему расстоянию (рис. 16.3, а, б). Второй способ предназначен для оценки длины ортогональных соединений, прокладываемых параллельно осям

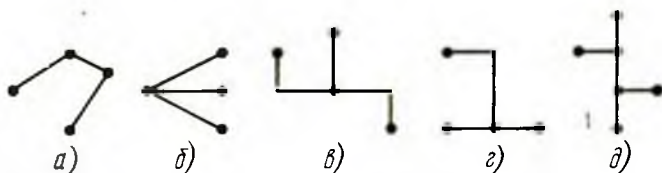


Рис. 16.3. Виды монтажных соединений:
а — цепь; б — звезда; в — д — ортогональные соединения

координат (рис. 16.3, в — д), когда соединения проводятся по магистралям или каналам. Третий способ полезен, когда в задачах оптимизации требуется уменьшение не только суммарной, но и максимальной длины проводников. При использовании последней метрики при $s=2, 3, \dots$ «длинные» соединения сильнее влияют на функцию оптимизации.

Таким образом, введение метрики монтажного пространства позволяет дать количественную оценку качества размещения элементов и монтажа.

Рассмотрим математические модели радиоэлектронных схем, используемые в автоматизированном конструкторском проектировании [63]. Любую исходную схему (функционально-логическую или принципиальную электрическую) представляют как коммутацион-

ную. Коммутационная схема — это набор элементов, связанных между собой соединениями, по которым подаются сигналы на полюсы элементов (рис. 16.4,а).

Элементы схемы и внешние выводы каждого элемента нумеруются, а выводы схемы, имеющие одинаковый потенциал, группируются в комплексы соединений, также нумерованные (рис. 16.4,б).

Перенумерованные n элементов схемы составляют множество $E = \{e_i, i = 0, 1, 2, \dots, n\}$. Внешние выводы элементов образуют семейство множеств $\{C_i = \{c_{ij}, j = 1, 2, \dots, k_i\}, i = 0, 1, 2, \dots, n\}$, где i — номер элемента; j — номер вывода в данном элементе; k_i — число выводов i -го элемента. Отметим, что элементы множества C_0 относятся к внешним выводам схемы, образуя специальную группу. Комплексы соединений, выделенные на рис. 16.4,б пунктирными линиями, образуют семейство множеств $V = \{v_i, i = 1, 2, \dots, M\}$, где v_i — множество эквипотенциальных выводов схемы; M — число комплексов. Поскольку выводы, принадлежащие различным комплексам соединений, непосредственно не связаны (только через элемент), то

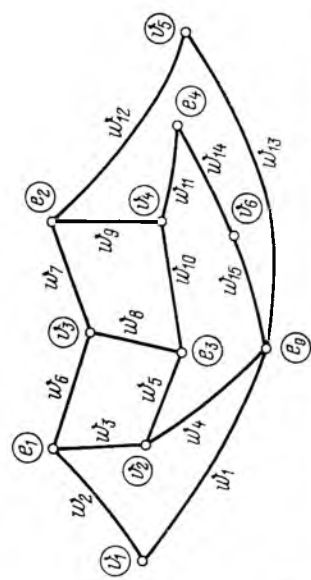
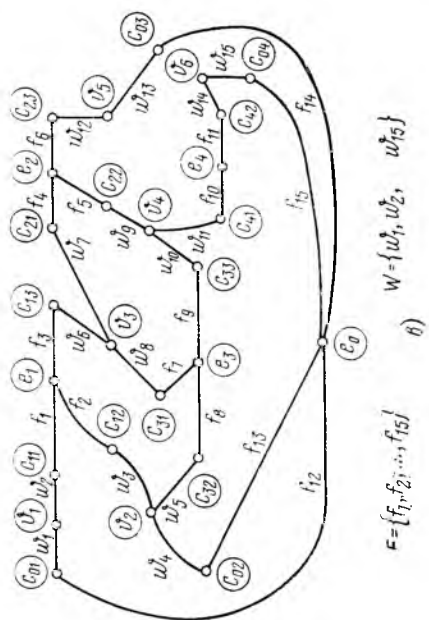
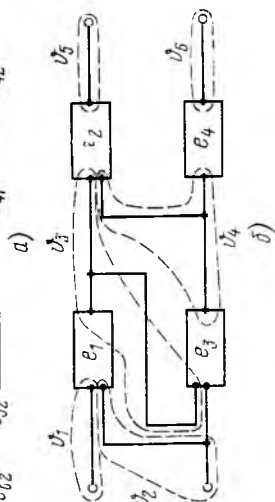
$$C = \bigcup_{i=0}^n c_{ij} = \bigcup_{j=1}^M v_j, \quad v_j \neq \emptyset, \quad v_i \cap v_j = \emptyset \quad (i \neq j). \quad (16.2)$$

Приведенное формальное описание может быть представлено в виде графа (рис. 16.4,в), если элементы множеств E, C, V рассматривать как вершины, а также ввести ребра, определяющие принадлежность выводов элементам (элементные ребра F) и принадлежность выводов комплексам соединений (сигнальные ребра W). Подобная математическая модель в виде графа широко используется в задачах дискретной оптимизации.

Граф с вершинами трех типов (E, C, V) и ребрами двух типов (F, W) дает наиболее детальное представление схемы. Этот граф используется при задании полной информации о схеме в процессе автоматизированного проектирования. Однако в ряде задач и алгоритмов можно обойтись меньшим объемом информации о схеме, например, используя граф, полученный после втягивания вершин C_i в вершины E и удаления элементных ребер F (рис. 16.4,г).

Для отображения коммутационной схемы в ЭВМ применяется как матричное, так и списочное представления рассмотренного графа. В матричном представлении используются матрицы инцидентий «номер комплекса — номер вывода элемента» или «номер элемента — номер вывода элемента» размерности $\left(M \times \sum_{i=0}^n k_i\right)$ и

$\left(n \times \sum_{i=0}^n k_i\right)$ соответственно. Содержащее ту же информацию списочное представление позволяет сократить объем исходных данных, а также повысить эффективность алгоритмов как по памяти, так и по времени (здесь следует провести аналогию с применением разре-


$$A = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} c_{01} & c_{02} & c_{03} & c_{04} & c_{11} & c_{12} & c_{13} & c_{21} & c_{22} & c_{23} & c_{31} & c_{32} & c_{33} & c_{41} & c_{42} \end{matrix}$$
$$\begin{array}{c} \text{Номер} \\ \text{элементарной} \\ \text{выборки} \end{array} \quad \begin{array}{c} \{ (0,1), (1,1), \\ (2,2), (3,3) \} \\ \downarrow v_4 \end{array} \quad \begin{array}{c} \{ (0,2), (1,2), (3,2); \\ (4,1), (0,3), (2,3) \} \\ \downarrow v_5 \end{array} \quad \begin{array}{c} \{ (1,3), (2,1), (3,1); \\ (0,4), (4,2) \} \\ \downarrow v_6 \end{array}$$

Индольные элементы $\{1, 3, 5, 9, 12, 14\}$
 комплексов $\{v_1, \dots, v_6\}$

l_0	l_1	l_2	l_3	l_4
l_1	1	1	3	3
l_2	2	2	4	2
l_3	5	5	3	5
l_4	6	-	-	-

$$T = \|t_{ij}\|_{n \times k} =$$

Рис. 16.4. Представление модели коммутационной схемы в задачах автоматизированного конструкторского проектирования:

a — коммутационная схема; *б* — комплексы соединений на схеме; *в* — граф схемы; *г* — граф элементарных комплексов соединений; *д* — матричное и списочное описания комплексов соединений; *е* — сокращенное табличное описание графа схемы ($n \times k$ — число элементов, $k = \max k_i$; $i = 0, 1, \dots, n$ — максимальное число выводов элементов)

женных матриц при задании топологической информации о схеме при электрическом расчете).

Пример матричного и списочного описания полной модели (см. рис. 16.4,в) коммутационной схемы представлен на рис. 16.4,д. Наиболее компактный способ записи информации о коммутационной схеме дает описание графа с помощью одной свернутой матрицы, строки которой соответствуют элементам схемы, столбцы — номерам выводов элементов, а элементами матрицы являются номера комплексов соединений, которым принадлежат выводы (рис. 16.4,е). Как и в предыдущем примере, вместо матрицы может использоваться списочная структура. Последнее представление удобно для подготовки исходной информации, так как, являясь наиболее кратким, оно в то же время обеспечивает получение любого другого представления графа схемы путем программного преобразования.

16.3. АЛГОРИТМЫ КОМПОНОВКИ

Задача компоновки элементов устройства в блоки обычно ставится как задача оптимального разбиения (разрезания) устройства, для которого задана функциональная, логическая или принципиальная схема. Это значит, что известен набор элементов и их соединения в проектируемой аппаратуре. Необходимо разбить заданное множество элементов A на множество R групп элементов A_i , таких, что

$$R = \{A_i \in A, i = 1, 2, \dots, N\}, \bigcup_{i=1}^N A_i = A, A_i \cap A_j = \emptyset \ (i \neq j), \quad (16.3)$$

N — число групп элементов (блоков, панелей или ТЭЗов в зависимости от уровня, на котором осуществляется разбиение).

Требуется, чтобы разбиение R было в заданном смысле оптимальным и удовлетворяло ограничениям, отражающим реализуемость, технологичность, надежность, ремонтпригодность, стоимость производства и т. д. Ограничения могут трансформироваться в критерии оптимизации и наоборот, в результате чего существует целый ряд различных формулировок задачи компоновки [23, 63, 64].

При формулировке задач компоновки используются следующие условия (критерии оптимизации или ограничения):

1. Каждый блок содержит не более P компонентов и не более Q выводов.

2. Каждый блок размещается в заданном объеме V_i и имеет длину внешних связей не более E_i (это модификация первого требования).

3. Общее число соединений между N блоками минимально.

4. Число блоков N минимально.

5. Блоки выбираются из числа стандартных, составляющих библиотеку эталонных элементов (приведение задачи компоновки к задаче покрытия схемы совокупностью элементов стандартного набора).

6. Число типов используемых блоков минимально (это задача типизации, решение которой направлено на ускорение проектирования, удешевление производства, удобство обслуживания в эксплуатации).

7. Число выводов Q_i в каждом блоке минимально (это требование направлено на минимизацию наибольшей задержки прохождения сигнала через систему в предположении, что каждое внешнее соединение вносит существенную задержку, а внутренние соединения вносят пренебрежимо малую задержку).

8. Выполняются заданные ограничения на задержки распространения сигналов.

9. Обеспечивается удобство тестирования, доступность элементов в блоках и низкая стоимость ремонта.

Задача компоновки (оптимального разбиения) принципиально не может иметь общего алгоритма автоматизированного проектирования из-за большого размера и разнообразия условий компоновки. Сложность состоит также в трудности формализации совокупности критериев, часть которых является немонотонными, как и сами конструктивные параметры.

При решении задач компоновки используются математические формулировки задач целочисленного (дискретного) линейного или нелинейного программирования. Комбинаторные алгоритмы, связанные с перебором всего множества элементов A , как правило, неприемлемы из-за большого размера задачи. Поэтому используются приближенные алгоритмы компоновки последовательного, последовательно-параллельного или итерационного типа.

В конструктивных последовательных алгоритмах оптимальное разбиение формируется путем последовательного удовлетворения m независимых функций Φ_i (см. п. 1—8) — критериев оптимальности — при включении одного элемента или группы элементов. Например, «следующий $(j+1)$ -й» компокуемый элемент выбирается из всех оставшихся элементов, если из элементов, оптимальных по Φ_i , он оптимален также по Φ_{i+1} ($i=1, 2, \dots, m$). Формирование следующего k -го блока начинается после завершения формирования предыдущих блоков с номерами $1, 2, \dots, k-1$. В кон-

структивных последовательно-параллельных алгоритмах формируется сразу несколько блоков.

В итерационных алгоритмах критерий оптимальности

$$\Phi = a_1\Phi_1 + a_2\Phi_2 + \dots + a_m\Phi_m, \quad (16.4)$$

где $a_1, a_2, \dots, a_m$ — весовые коэффициенты, которые выбираются эвристически с целью учета неравнозначности используемых функций оптимизации.

Набор функций $\{\Phi_1, \Phi_2, \dots, \Phi_m\}$ определяется количественными представлениями требований компоновки, рассмотренных выше. Если часть требований представлена в виде ограничений, то на каждом шаге алгоритма компоновки проверяются условия, диктуемые этими ограничениями, и в соответствии с результатами проверки изменяется ход процесса компоновки.

Рассмотрим пример итерационного алгоритма распределения элементов по неограниченному числу допустимых блоков, в каждом из которых суммарный «вес» P элементов не должен превышать заданной величины P^* при условии, что минимизируется степень связи (число соединений) между блоками.

Суммарная степень связи, используемая в качестве критерия оптимальности, выражается соотношением

$$\Phi = \sum_{i \in R} \sum_{j \in R} \left(\sum_{T_R \in A_i} \sum_{T_S \in A_j} m_{RS} \right), \quad i \neq j, \quad (16.5)$$

где m_{RS} — коэффициент связи элементов T_R и T_S , принадлежащих разным блокам.

Модель компоновки схемы представим в виде графа, вершинам которого соответствуют элементы устройства. Примем начальное разбиение (нулевой шаг) на $N=n$ блоков $R = \{A_1, A_2, \dots, A_n\}$, когда в каждый блок входит одна вершина. Очевидно, что в этом случае значение функции Φ из (16.5) на нулевом шаге $\Phi_0 = \max$. При объединении двух вершин T_R и T_S в блок A_k число блоков уменьшается ($N = N - 1$) и уменьшается суммарная степень связи ($\Phi_1 = \Phi_0 - m_{RS}$). Очевидно, что на каждом i -м шаге максимальное снижение Φ_i получится при объединении блоков, имеющих максимальную степень связи. Однако при объединении блоков увеличивается вес k -го блока: $P_k = P(T_R) + P(T_S)$. Поэтому процесс объединения допустим до тех пор, пока $P_k < P^*$. Полученные блоки фиксируются, а включенные в них элементы на следующих шагах компоновки не используются. Результат разбиения схемы приведен на рис. 16.5, где цифры соответствуют весам элементов при $P \leq 10$.

В ряде случаев основу итерационных алгоритмов составляет классическая задача о назначениях. После задания начального разбиения на блоки осуществляется такое перераспределение эле-

ментов в блоках, которое минимизирует суммарную цену назначения. Процедура компоновки организуется, например, по алгоритму

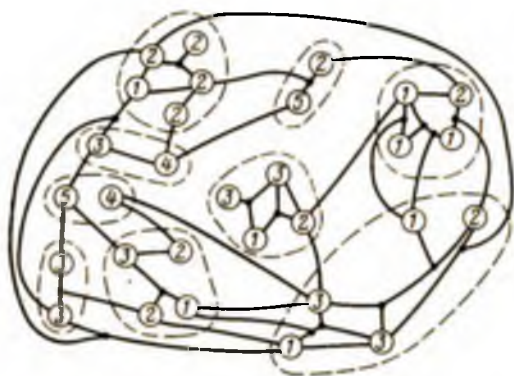


Рис. 16.5. Пример применения алгоритма компоновки: результат разбиения схемы при $P \leq 10$

парных перестановок следующим образом. Из каждого блока выносят по одному компоненту, а затем вносят каждый из них на все свободные места, определяя при этом суммарный качественный показатель. Перестановка фиксируется и закрепляется, если качественный показатель уменьшается. Использование упорядочения в переборе позволяет сократить число рассматриваемых вариантов. Пример применения подобного алгоритма показан на рис. 16.6.

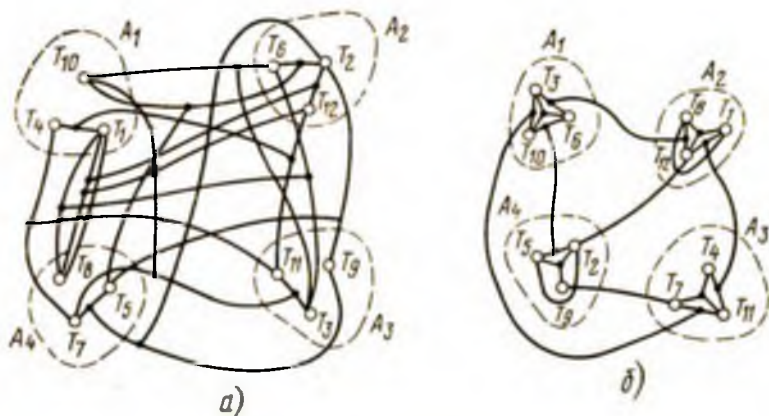


Рис. 16.6. Пример применения итерационного алгоритма назначения для компоновки:

а — начальное разбиение; б — окончательное разбиение

В случае, когда при компоновке блоков необходимо минимизировать задержку, вносимую внешними соединениями (т. е. требуется сократить число внешних соединений), одним из способов является дублирование элементов. Это характерно для технологии микроэлектроники, когда цена внешнего соединения много выше цены внутренних, в том числе дополнительных элементов схемы. Дублирование элементов используется в сочетании с другими рассмотренными выше алгоритмами компоновки.

В качестве примера на рис. 16.7, а для простой логической схемы показано начальное разбиение. Пунктирная линия I разделяет

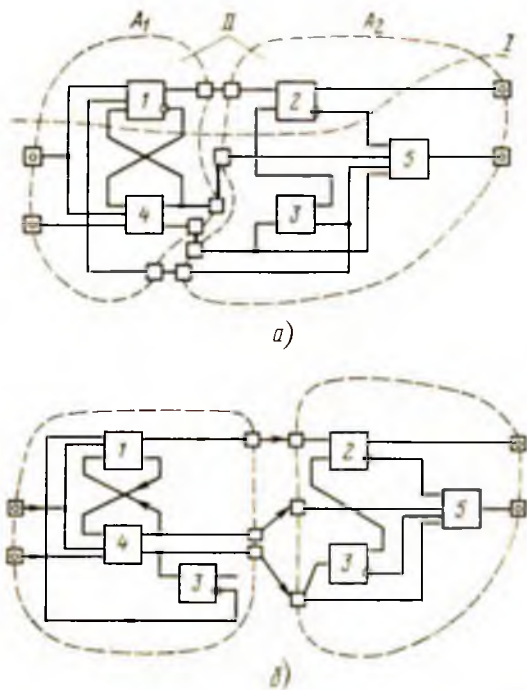


Рис. 16.7. Пример использования нескольких алгоритмов компоновки: а — результат применения итерационного алгоритма по методу парных перестановок (I — начальное разбиение $\{1, 2\}, \{3, 4, 5\}$; II — улучшенное разбиение $\{1, 4\}, \{2, 3, 5\}$); б — оптимизация разбиения дублированием элемента 3

элементы схемы на два блока, при образовании которых потребуется введение $2 \times 7 = 14$ дополнительных контактных площадок. Там же показано улучшенное разбиение, полученное после применения итерационного алгоритма парных перестановок: пунктирные линии II разделяют элементы схемы на два блока, в результате чего образуются $2 \times 4 = 8$ новых контактных площадок (на рисунках обозначены $\square$). Дублирование в этой схеме элемента 3 обеспечивает возможность разбиения ее на два блока при введении лишь $2 \times 3 = 6$ контактных площадок (рис. 16.6, б). Иногда при

решении задач компоновки применяется дублирование не только элементов, но и сигналов в схеме. В этих случаях необходима проверка или пересмотр проектируемой схемы.

Таким образом, после выполнения компоновки и определения реальных задержек, а также корректировки схемы желательно провести временное логическое или схемотехническое моделирование с целью проверки функционирования проектируемого устройства.

16.4. АЛГОРИТМЫ РАЗМЕЩЕНИЯ

После компоновки элементов РЭУ требуется осуществить для каждого стандартного или уникального блока оптимальное размещение элементов в регулярном или нерегулярном монтажном пространстве. В общем случае нужно найти на множестве позиций монтажного пространства блока A_i такое размещение компонентов $\{T_1, T_2, \dots, T_n\}_i$, принадлежащих этому блоку, при котором достигается минимум заданного критерия качества размещения L , выраженного с использованием соотношений (16.1) [63, 64].

Особенностью критериев, используемых в задаче размещения, является их эвристический характер, так как все они косвенно отражают истинную цель размещения — максимальное упрощение последующего процесса трассировки и минимизацию числа непроведенных трасс. Кроме того, критерии качества размещения учитывают такие факторы, как временные задержки, возникновение помех, ослабление сигнала.

В задачах размещения используются следующие критерии качества:

1. Минимальная суммарная длина соединительных проводников.
2. Минимальная длина проводников, соединяющих две наиболее удаленные точки каждой цепи.
3. Минимальная длина проводников, соединяющих источник сигнала с наиболее удаленной нагрузкой.
4. Минимальная суммарная площадь зон реализации всех цепей (обеспечивает минимизацию временных задержек между компонентами).
5. Минимальное число проводников, длина которых превышает заданную величину.
6. Минимальная наибольшая длина соединительных проводников.
7. Максимально близкое размещение компонентов, имеющих наибольшее число общих цепей, с учетом допустимого расстояния между элементами.

В регулярном монтажном пространстве с числом позиций m при наличии в схеме n компонентов ($n \leq m$), каждый из которых занимает одну позицию, число возможных размещений

$$N_p(m, n) = n! C_m^n = \frac{m!}{(m-n)!}. \quad (16.6)$$

Очевидно, что использовать для определения оптимального варианта размещения метод прямого перебора в практических задачах, где $n > 20$, невозможно даже на современных ЭВМ. В связи с большим размером задач, а также сложностью учета указанных выше факторов оптимальности, для задачи размещения не существует методов, которые гарантируют оптимальное решение. При решении практических задач размещения широко применяются эвристические процедуры, алгоритмы которых имеют конструктивный или итерационный характер.

Конструктивные алгоритмы (например, последовательный метод связывания пар, или последовательно-параллельный метод расширения ядер) формируют размещение, начиная от некоторого заданного, например от контактов ИС, последовательным добавлением элементов с учетом критерия оптимальности и заданных ограничений до тех пор, пока все элементы не будут установлены. Конфигурация размещения при использовании конструктивных алгоритмов определяется только после полного завершения работы алгоритма.

Итерационные алгоритмы (например, метод парных перестановок, метод групповых перестановок, методы, использующие механические или электрические аналогии, методы ветвей и границ), которые улучшают начальное размещение, имеют преимущество перед конструктивными алгоритмами. Оно заключается в том, что на любом этапе работы итерационного алгоритма уже существует законченный вариант размещения, пригодный для практического использования.

Задача оптимизации ставится, как правило, как задача линейного или квадратичного назначения с введением весовых коэффициентов в функцию критерия качества [23, 65]:

$$\Phi = k_1 \sum_i \sum_j c_{ij} d_{p(i)p(j)} + k_2 \sum_i a_{ip(i)}, \quad (16.7)$$

где k_1, k_2 — весовые коэффициенты, учитывающие важность используемого критерия качества (выбираются эвристически); c_{ij} — элементы матрицы стоимости $C = [c_{ij}]$ назначения i -го компонента на j -ю позицию; $d_{p(i)p(j)}$ — элементы матрицы расстояний $D = [d_{hl}]$, вычисленные с учетом соотношений (16.1); $a_{ip(i)}$ — элементы стоимости назначения по другому критерию; p — заданная перестановка элементов. Если матрицы C и D симметричны, то квадратичная задача назначения (при $k_2 = 0$) часто формулируется следующим образом: найти перестановку p , которая минимизирует функционал

$$\Phi = \sum_{i < j} c_{ij} d_{p(i)p(j)}. \quad (16.8)$$

Следует обратить внимание на то, что, кроме большой размерности задачи разбиения [см. (16.6)], очень трудоемкой является подготовка исходной информации для представления как самой схемы, так и данных, требуемых для расчета критериев качества

размещения (16.7), которые обязательно должны отражать требования, вытекающие из технологии изготовления устройства.

С целью ускорения получения квазиоптимального или локально-оптимального решения используются эвристические приемы. Кроме того, разработка новых эвристических алгоритмов связана с возникающими при проектировании БИС специфическими требованиями, которые не учитываются метрическими критериями, построенными на основании (16.1) и используемыми в классической задаче назначения. К таким требованиям относятся равномерное распределение соединений по площади, определенное расположение элементов и контактов, ограничение числа пересечений и словес при монтаже.

Как правило, на практике используют сочетание методов, а также учитывают специфику технологии изготовления схем, что приводит к большому многообразию как постановок задач, так и программных реализаций, имеющих достаточно сложный характер.

В качестве примера простого алгоритма решения задачи размещения рассмотрим конструктивный алгоритм «обратного размещения» [63]. В этом методе задается матрица соединений n элементов $C = [c_{ij}]$ и матрица расстояний между n позициями $D = [d_{ij}]$. Для каждого элемента рассчитывается по матрице C суммарное число соединений с остальными $(n-1)$ элементами:

$$\left\{ c_i = \sum_{j=1}^n c_{ij}, \quad i = 1, 2, \dots, n \right\} \quad (c_{ij} = 0 \text{ при } i = j),$$

а для каждой позиции монтажного пространства вычисляется характеристика, определяющая суммарное расстояние этой позиции до остальных:

$$\left\{ d_i = \sum_{j=1}^n d_{ij}, \quad i = 1, 2, \dots, n \right\} \quad (d_{ij} = 0 \text{ при } i = j).$$

Очевидно, что центральные позиции монтажного пространства, имеющие меньшие значения d_i , наиболее благоприятны для расположения сильно связанных элементов, имеющих большие значения c_i . Минимизируя функцию (16.8), можно получить квазиоптимальное с точки зрения суммарной взвешенной длины соединений размещение, используя следующую эвристическую процедуру.

1. Элементы сортируют по возрастанию характеристики c_i ($c_{i1} \leq c_{i2} \leq \dots \leq c_{in}$).
2. Позиции сортируют по убыванию характеристики d_j ($d_{j1} \geq d_{j2} \geq \dots \geq d_{jn}$).
3. Правило размещения определяется перестановкой $p(i_k) = j_k$, $k = 1, 2, \dots, n$.

Таким образом, на «лучшие» позиции устанавливаются «худшие» элементы.

На рис. 16.8 показан пример применения рассмотренного алгоритма размещения. Видно, что из-за совпадения характеристик

$$a) \quad C = \begin{bmatrix} 0 & 2 & 2 & 3 & 9 & 4 \\ 2 & 0 & 1 & 10 & 5 & 6 \\ 2 & 1 & 0 & 3 & 4 & 7 \\ 3 & 10 & 3 & 0 & 7 & 8 \\ 9 & 5 & 4 & 7 & 0 & 11 \\ 4 & 6 & 7 & 8 & 11 & 10 \end{bmatrix}, \{c_i\} = \begin{bmatrix} 20 \\ 24 \\ 17 \\ 31 \\ 36 \\ 36 \end{bmatrix}; \quad D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 0 & 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & 1 & 2 & 3 \\ 3 & 2 & 1 & 0 & 1 & 2 \\ 4 & 3 & 2 & 1 & 0 & 1 \\ 5 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}, \{d_i\} = \begin{bmatrix} 15 \\ 11 \\ 9 \\ 9 \\ 11 \\ 15 \end{bmatrix}$$

упорядоченное $\{c_i\}$: 3 1 2 4 5 6 (номер элемента)

упорядоченное $\{d_i\}$: 1 6 2 5 3 4 (номер позиции)

б) k : 1 2 3 4 5 6

в)

1

2

3

4

5

6

г)

1

2

3

4

5

6

 $\Phi = 197$

д)

3

2

5

6

4

1

 $\Phi = 179$

е)

1

6

5

6

2

3

 $\Phi = 170$

Рис. 16.8. Пример применения алгоритма обратного размещения:

a — матрица соединений C и множество $\{c_i\}$, полученное суммированием по строкам C , матрица расстояний — D и множество $\{d_i\}$, полученное суммированием по строкам; *б* — определение расстановки элементов установлением соответствия элементов $\{c_i\}$ и $\{d_j\}$ согласно алгоритму; *в* — пронумерованные пустые позиции для расстановки элементов; *г* — начальное размещение; *д* — размещение в соответствии с полученным на рис. 16.8,б результатом; *е* — оптимальное размещение

внутри $\{c_i\}$ и $\{d_i\}$ (рис. 16.8,а) имеется ряд вариантов размещения, первый из которых (в соответствии с порядком следования элементов после сортировки) приведен на рис. 16.8,б. Этот вариант понижает критерий оптимальности от $\Phi = 197$ (рис. 16.8,г) до $\Phi = 179$ (рис. 16.8,д). Однако наилучшим является вариант размещения на рис. 16.8,е ($\Phi = 170$). Рассмотренный метод, как правило, дает решение, несколько уступающее по качеству размещению, выполненному опытным конструктором, в то время как, например, применение автоматизированного размещения с помощью последовательных алгоритмов сокращает длину соединений на 10...20% относительно неавтоматизированного размещения. Однако рассмотренный метод отличается чрезвычайной простотой, и он может использоваться для выбора начального варианта размещения.

Следует отметить, что даже на таком простом примере видны трудности решения задачи размещения, которые значительно возрастает при многокритериальной оптимизации размещения компонентов.

Задача трассировки состоит в построении соединений между выводами размещенных в заданном монтажном пространстве элементов в соответствии с принципиальной схемой устройства при учете конструктивных ограничений. Обычно трасса формируется в виде множества связанных отрезков, соединяющих точки электрической цепи. При решении задачи трассировки используются следующие критерии и условия:

1. Минимальная суммарная длина соединений.
2. Минимальное число соединений, длина которых превышает заданное значение.
3. Минимальное число переходов между слоями.
4. Минимальное число слоев.
5. Минимальные паразитные помехи.
6. Максимальная удаленность трасс соединений.
7. Число слоев не превышает заданного значения.
8. Длина соединения не превышает заданного значения.
9. Уровень помех, наводимых в каждой трассе, не превышает допустимого значения.
10. Число соединений (паек) к одному выводу не превышает заданного значения.
11. Задается преимущественное направление трасс в каждом слое. При проектировании многослойных структур, как правило, отдельно решается задача оптимального расслоения, в которой минимизируется показатель

$$P = \sum_{r=1}^z \sum_{k=1}^{M-1} \sum_{s=k+1}^M \pi_{ks} \delta'_{ks}, \quad (16.9)$$

где r — номер слоя; z — число слоев; M — число цепей в схеме; π_{ks} — характеристика степени пересечения k -й и s -й трасс;

$$\delta'_{ks} = \begin{cases} 0, & \text{если трассы } k \text{ и } s \text{ не принадлежат слою } r; \\ 1, & \text{если трассы } k \text{ и } s \text{ принадлежат слою } r. \end{cases}$$

Характеристика π_{ks} задается различными величинами, например: площадью перекрытия зон реализации цепей a_k и a_s ; числом пересечений полных подграфов цепей a_k и a_s ; числом пересечений минимальных связывающих сетей цепей a_k и a_s ; числом выводов цепей a_k и a_s , принадлежащих пересечению зон реализации этих цепей.

В общей проблеме автоматизации конструкторского проектирования трассировка соединений — это наиболее трудная задача, в которой прослеживается особенно тесная связь с конструктивно-технологической реализацией. Исходной информацией в задаче трассировки являются как список цепей проектируемой схемы, так и параметры конструкций элементов (форма и геометрические размеры) и монтажного пространства (допустимые расстояния, форма соединений, число пересечений и т. п.), а также данные по

размещению элементов. Поэтому трассировка осуществляется между точками, координаты которых заданы в абсолютных или относительных единицах.

Задача трассировки имеет метрический и топологический аспекты. Метрический аспект связан с учетом конструктивных размеров

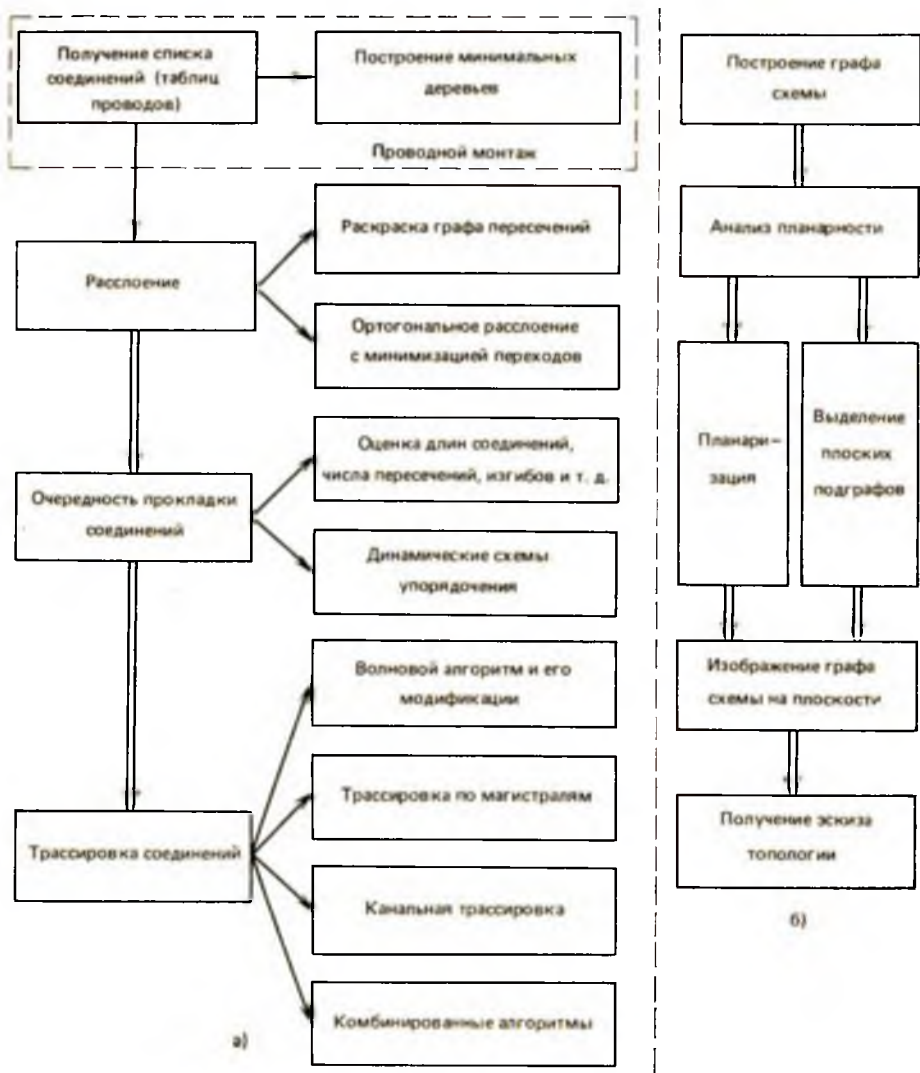


Рис. 16.9. Последовательность применения методов трассировки ($\Rightarrow$) к различным типам схем и используемые алгоритмы:

a — схемы регулярные, многослойные, из однотипных элементов (печатный монтаж или многослойные ИС и БИС); *б* — схемы нерегулярные, из разнотипных элементов (однослойный монтаж на печатных платах, гибридные ИС)

элементов, соединений и монтажного пространства (размещения). Топологический аспект связан с учетом ограничений на число допустимых пересечений, число слоев схемы, т. е. связан с пространственным расположением отдельных частей и соединений схемы.

Алгоритмические методы трассировки, классификация которых приведена на рис. 16.9 [62], несмотря на их многообразие, не гарантируют проведения 100% соединений. По способу построения трасс методы трассировки разделяются на конструктивные и итерационные.

В последовательных конструктивных алгоритмах трассы цепей проводятся в определенном порядке одна за другой. Выбор очередности — эвристический. Обычно начинают прокладку трасс либо с самых длинных соединений, так как в сильно заполненном монтажном пространстве их труднее формировать, либо с самых коротких

соединений, которые плотнее заполняют монтажное пространство. Проложенные трассы фиксируются и при дальнейшей трассировке рассматриваются как препятствия (занятые ячейки монтажного пространства). Таким образом, в последовательных алгоритмах осуществляется локальная оптимизация при прокладке каждой трассы, но в результате после проведения ряда трасс некоторые участки могут блокироваться, что не позволяет выполнить трассировку автоматически (рис. 16.10).

В итерационных алгоритмах после прокладки всех трасс, которая осуществляется без учета взаимного влияния трасс, определяется функция качества трассировки как взвешенная сумма параметров трассы (длина, число пересечений, число перегибов). Наихудшие трассы удаляются, и процесс трассировки повторяется с учетом множества лучших соединений как препятствий.

Для решения задач трассировки соединений разработаны также алгоритмы, в которых объединены последовательные и итерационные процедуры — «мешающие» трассы деформируются и

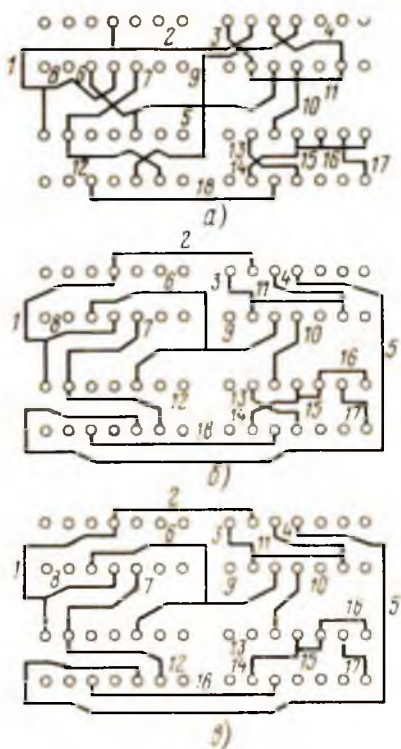


Рис. 16.10. Пример трассировки соединений:

а — по кратчайшим маршрутам; б — минимизация числа пересечений; в — устранение пересечений (непроведенное соединение 13)

удаляются, т. е. изменяется конфигурация уже проложенных соединений или очередность прокладки трасс.

Большинство известных универсальных алгоритмов трассировки основывается на волновом алгоритме определения пути (трассы соединений элементов), минимизирующем некоторую монотонную многомерную функцию качества этого пути. Идея волнового алгоритма иллюстрируется на рис. 16.11, где соединяются элементы *A*

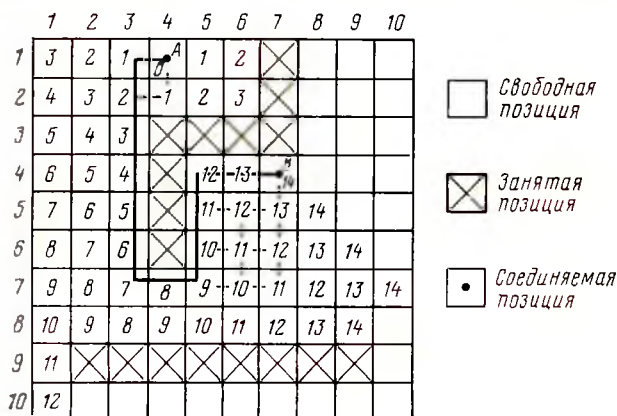


Рис. 16.11. Трассировка соединений волновым алгоритмом при наличии препятствий

и *B*, расположенные в точках (1, 4) и (4, 7) регулярного монтажного пространства. Цифрами 0, 1, 2, ..., 14 показаны «фронты» распространения волны от точки (1, 4) — состояние 0, до тех пор пока она не достигнет точки (4, 7) — состояние 14. Оптимальная трасса получается соединением точек в обратной последовательности 14-13-...-1-0, как трасса, имеющая минимальную длину, минимальное число изгибов и обеспечивающая максимальную плотность монтажа. Штрихами показаны трассы, имеющие ту же длину, но худшие по остальным критериям. Модификации волнового алгоритма направлены на повышение быстродействия и уменьшение требуемого объема памяти.

Трассировка с использованием другого эвристического алгоритма — лучевого — показана на рис. 16.12. Между соединяемыми точками *A* (6, 3) и *B* (5, 8) проводится луч. При переходе к следующей точке монтажного пространства определяется направление трассы, минимально отличающееся от направления луча с учетом указанного на диаграмме приоритета (нумерации направлений) и необходимости обхода препятствий, образованных занятыми позициями. Штриховой линией отмечен оптимальный путь (его длина в два раза меньше первого), проложенный при проведении луча от *B* к *A*, т. е. в обратном направлении.

Следует отметить, что приведенные при рассмотрении постановки задачи трассировки критерии оптимальности вводятся не для уп-

равления процессом трассировки, а лишь для оценки качества полученного решения.

В ряде случаев делается попытка учесть последующие шаги трассировки и организовать параллельную трассировку всех соединений. Примером являются алгоритмы, использующие канальное представление магистралей межсоединений. На рис. 16.13 пока-

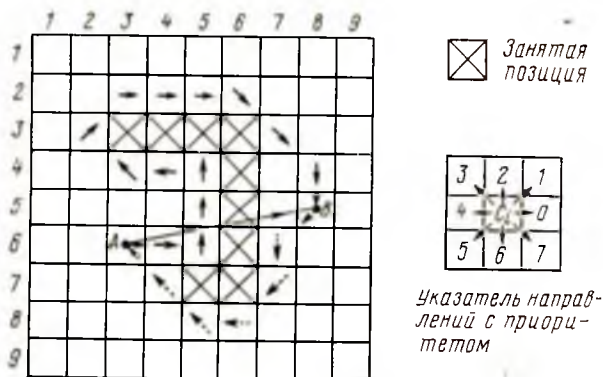


Рис. 16.12. Трассировка соединений лучевым алгоритмом при наличии препятствий

зана сеть вертикальных B_1, B_2, B_3 и горизонтальных $\Gamma_1, \Gamma_2, \Gamma_3$ каналов, на которой проведено соединение двух элементов. Приведенная конструкция с двумя слоями горизонтальной и вертикальной коммутации и возможностью введения контактных переходов в точках сопряжения горизонтальных и вертикальных слоев типична для устройств, реализуемых на двухсторонних печатных платах, а также для больших гибридных и монолитных интегральных схем. В алгоритмах, использующих представление о каналах, трассировка

осуществляется в два этапа: предварительная, с целью распределения трасс по каналам при равномерной их загрузке, и окончательная, в процессе которой уточняется расположение соединений на магистралях каналов.

Особый тип алгоритмов предназначен для трассировки соединений на плоскости без пересечений (см. рис. 16.10). Они используются при проектировании однослойного монтажа при наличии в устройстве разнотипных по форме и размерам элементов.

В связи с тем, что ни один из известных алгоритмов не гаран-

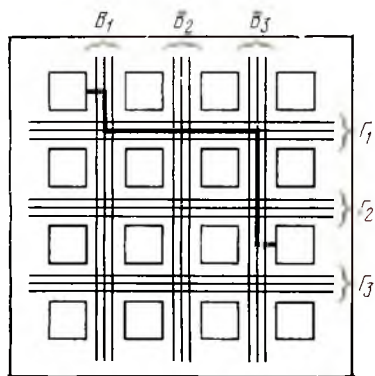


Рис. 16.13. Трассировка соединений с использованием каналов