

СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С МИКРОЭВМ



В.А. БЕСЕКЕРСКИЙ
В.В. ИЗРАНЦЕВ

СИСТЕМЫ
АВТОМАТИЧЕСКОГО
УПРАВЛЕНИЯ
С МИКРОЭВМ



ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ТЕХНИЧЕСКОЙ КИБЕРНЕТИКИ

В.А. БЕСЕКЕРСКИЙ, В.В. ИЗРАНЦЕВ

СИСТЕМЫ
АВТОМАТИЧЕСКОГО
УПРАВЛЕНИЯ
с микроЭВМ



МОСКВА "НАУКА"

ГЛАВНАЯ РЕДАКЦИЯ

ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ

1987

~~ББК 32-81~~

Б53

~~УДК 62-50~~

Бесекерский В.А., Изранцев В.В. Системы автоматического управления с микроЭВМ. — М.: Наука. Гл. ред. физ.-мат. лит., 1987 — 320 с. — (Теоретические основы технической кибернетики).

Систематическое изложение принципов построения и теории систем автоматического управления с микроЭВМ в контуре управления. Приводятся сведения по структуре подобных систем, их организации, устройствам ввода-вывода, программному обеспечению, этапам их разработки и отладки, а также математический аппарат исследования, методы анализа и синтеза при известных и неизвестных статистических характеристиках входных воздействий и методы реализации цифровых элементов систем.

Для специалистов, работающих в области проектирования и исследования систем автоматического управления, а также для студентов вузов. Табл. 31. Ил. 92. Библиогр. 234 назв.

Рецензенты

доктор технических наук *Э.Л. Ицкович* и
кандидат технических наук *Ю.Б. Любимов*

Б $\frac{1502000000-048}{053(02)-87}$ 145-87

© Издательство "Наука".
Главная редакция
физико-математической
литературы, 1987

ПРЕДИСЛОВИЕ

Появление и быстрое совершенствование микропроцессорных средств привело к широкому их использованию в различных областях народного хозяйства. Функциональная гибкость, высокая надежность, малые габариты и стоимость микропроцессорных средств обусловили целесообразность их применения в различной аппаратуре, в том числе в системах управления.

В связи с большими возможностями микропроцессорных средств по сравнению с аппаратными средствами аналоговой и цифровой техники, выполненными на основе использования схем малой и средней интеграции, появилась перспектива построения систем, реализующих более сложные функции управления при более высокой степени надежности.

Появление микропроцессорных средств привело к существенному изменению процесса проектирования цифровых систем управления. Это вызвано как использованием более сложных функциональных компонентов, так и новым архитектурным построением систем управления, заменой аппаратурной реализации решением посредством программных методов.

Проектирование цифровых систем управления с микроЭВМ требует от разработчиков знания вычислительной техники и программирования на качественно новом уровне, опыта проектирования как технических средств, так и программного обеспечения.

При проектировании цифровых систем управления с микроЭВМ разработчику приходится решать задачи, многие из которых возникают и при проектировании современных ЭВМ. Среди них – организация процессорных элементов и обеспечение их взаимодействия с памятью, построение каналов обмена информацией между микроЭВМ и внешними устройствами, согласование функционирования элементов системы, имеющих различную скорость работы и др.

Здесь проявляется важная особенность использования микроЭВМ в прикладных задачах управления. Она заключается в том, что микроЭВМ является частью системы управления. Ее выбор или, при необходимости, проектирование должны быть выполнены с учетом требований, налагаемых на систему конкретной сфе-

рой ее использования. Решение этой задачи не является тривиальным. В настоящее время издано значительное количество работ, посвященных описанию микропроцессоров и микроЭВМ. В то же время явно недостаточно публикаций, освещающих построение устройств и систем на базе микропроцессорной техники.

Настоящая монография посвящена вопросам построения и расчета систем управления с микроЭВМ. В ней рассматриваются общие проблемы, возникающие при их разработке, архитектура подобных систем, реализация цифровых функциональных элементов, а также методы расчета при детерминированных и случайных воздействиях.

Глава 3, параграфы 4.1, 4.2, 4.4, и глава 5 написаны В.А. Бесекерским. Главы 1 и 2, параграфы 4.3 и 4.5 – В.В. Изранцевым.

Авторы выражают глубокую признательность Э.Л. Ицковичу и Ю.Б. Любимову, сделавшим ряд ценных замечаний при рецензировании книги.

Глава 1

ОБЩИЕ СВЕДЕНИЯ О СИСТЕМАХ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С МИКРОЭВМ

§ 1.1. СТРУКТУРА ЦИФРОВЫХ СИСТЕМ УПРАВЛЕНИЯ

Терминология. Приведем определения некоторых общих понятий, применяемых в микропроцессорной технике. Отметим, что многие термины до настоящего времени не являются устоявшимися. Поэтому при составлении терминологического словаря использованы не только государственные и отраслевые стандарты, но и работы отечественных и зарубежных авторов [3, 7, 40, 68–70].

Наиболее полный толковый словарь по вычислительной технике [2] содержит более 4000 терминов; приводимый же ниже словарь содержит минимальное число определений, необходимых для правильного понимания содержания книги.

Интегральная схема (микросхема) — микроэлектронное изделие, выполняющее определенную функцию преобразования и обработки сигналов и имеющее высокую плотность упаковки электрически соединенных элементов (или элементов и компонентов) и (или) кристаллов, которое с точки зрения требований к испытаниям, приемке, поставке и эксплуатации рассматривается как единое целое [40].

Элемент интегральной микросхемы — часть интегральной микросхемы, реализующая функцию какого-либо электрорадиоэлемента (транзистор, диод, резистор, конденсатор и др.), которая выполнена нераздельно от кристалла или подложки и не может быть выделена как самостоятельное изделие с точки зрения требований к испытаниям, приемке, поставке и эксплуатации.

Цифровая интегральная микросхема (цифровая микросхема) — интегральная схема, предназначенная для преобразования и обработки сигналов, изменяющихся по закону дискретной функции. Степень интеграции интегральной микросхемы — показатель степени сложности микросхемы, характеризуемый числом содержащихся в ней элементов и компонентов.

Компонент интегральной микросхемы — часть интегральной микросхемы, реализующая функцию какого-либо электрорадиоэлемента, которая может быть выделена как само-

стоятельное изделие с точки зрения требований к испытаниям, приемке, поставке и эксплуатации.

Степень функциональной сложности цифровой интегральной микросхемы — показатель степени сложности цифровой интегральной микросхемы, характеризуемый числом содержащихся в ней логических элементов, как правило, выполняющих функцию И или И-НЕ.

Унифицированная большая интегральная схема (БИС) — БИС, которая может быть применена при построении различных средств обработки информации.

Заказная БИС — БИС, которая может быть применена при построении только одного изделия либо узкого класса средств обработки информации.

Комплект БИС — совокупность типов БИС одной или нескольких серий, выполняющих различные функции, совместимых по конструктивно-технологическому исполнению и предназначенных для совместного применения при построении средств электронной аппаратуры.

Микропроцессор — программно управляемое устройство, непосредственно осуществляющее процесс обработки цифровой информации и управление им, построенное на одной или нескольких БИС.

Регистр — функциональный узел, предназначенный для приема, хранения и выдачи коротких последовательностей двоичных знаков, объединенных общим признаком. Например, машинное слово, код команды, слово состояния и др. Регистр общего назначения (РОН) — программно доступный для пользователя регистр процессора, который может быть использован при выполнении различных операций.

Память — функциональная часть ЭВМ, предназначенная для запоминания и (или) выдачи данных.

Запоминающее устройство (ЗУ) — изделие, реализующее память.

Оперативное запоминающее устройство (ОЗУ) — ЗУ с изменяемым в процессе выполнения программы содержимым памяти.

Постоянное запоминающее устройство (ПЗУ) — ЗУ с неизменным содержимым памяти.

Сверхоперативное ЗУ (СОЗУ) — ОЗУ небольшой емкости, время обращения к которому незначительно по сравнению с временем выполнения операций.

Перепрограммируемое постоянное запоминающее устройство (ППЗУ) — ПЗУ, запись содержимого которого может осуществляться многократно с использованием автоматических средств.

Интерфейс — средство стандартного сопряжения (соединения) устройств, отличающееся унификацией способов и средств физического соединения и процедур установления связи, обмена и завершения передачи информации.

Архитектура — совокупность общих принципов построения и характеристик технических и программных средств обработки данных, определяющих функциональные и эксплуатационные параметры изделия и существенных для организации его эффективного применения.

Структура — упорядоченное множество объектов и отношений между ними.

Вычислительная система (ВС) — совокупность нескольких ЭВМ с обобщенными или индивидуальными периферийными устройствами, взаимно координирующих свою работу при решении одной или различных задач и воспринимаемых каждым из пользователей как функционально единое целое.

Мини-ЭВМ — малая вычислительная машина, имеющая широкое применение благодаря малым габаритам, низкой стоимости, относительно высокой производительности. Длина слова в мини-ЭВМ также мала и составляет от 8 до 18 разрядов, в большинстве случаев — 16 разрядов.

МикроЭВМ — ЭВМ, состоящая из микропроцессора, полупроводниковой памяти, средств связи с периферийными устройствами и, при необходимости, пульта управления и источника питания, объединенных общей несущей конструкцией.

Управляющая микроЭВМ — микроЭВМ, которая использует данные (информацию) о процессе, вырабатывает выходные сигналы, управляющие работой объектов, участвующих в этом процессе.

Встраиваемая управляющая микроЭВМ — микроЭВМ, лишенная индивидуальных органов управления, конструктивно введенная в состав аппаратуры управляемого объекта.

Специализированная микроЭВМ — микроЭВМ, предназначенная для решения определенного класса задач.

Универсальная микроЭВМ — микроЭВМ, предназначенная для решения широкого класса задач.

Виртуальная ЭВМ — функциональный эквивалент вычислительной машины, внешних устройств и программного обеспечения, эмулируемый на конкретной вычислительной системе.

Производительность — характеристика системы, отражающая ее способность производить определенный объем работ. Определяется такими факторами, как пропускная способность, время ответа, доступность и т.п.

Программа — алгоритм преобразования данных в форме последовательности команд ЭВМ.

Команда — указание, определяющее один шаг в общем процессе выполнения программы.

Операнд — часть команды, указывающая адрес, по которому расположены данные, участвующие в операции.

Данные — информация, представленная в формализованном виде и предназначенная для обработки ее техническими средствами (например ЭВМ) или уже обработанная ими.

Разряд — позиция, которая может быть занята одним знаком.

Бит — двоичная цифра.

Компилятор (транслятор) — обслуживающая программа, выполняющая перевод на машинный язык программы, записанной на исходном языке программирования низкого уровня (высокого уровня).

Ассемблер — программа, осуществляющая перевод предположений языка ассемблера на машинный язык.

Мнемоника — символическое обозначение кода машинной операции.

Кросс-компилятор — компилятор, осуществляющий преобразование (перевод) программы, написанной на одном из языков программирования, на машинный язык ЭВМ с системой команд, отличной от той, на которой выполняется компиляция.

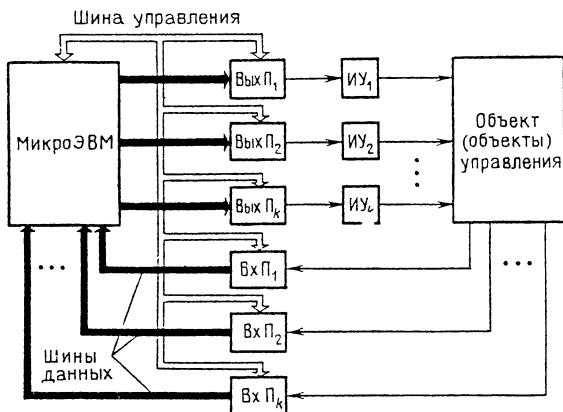
Эмуляция — имитация функционирования одной системы средствами другой системы без какой-либо потери функциональных возможностей или искажения получаемых результатов.

Прерывание — аппаратное прекращение выполнения программы для обеспечения возможности выполнения других программ.

Файл — набор логически связанных данных.

Влияние структуры цифровых систем на показатели стоимости и надежности. Значительные вычислительные и логические возможности микроЭВМ, сочетающиеся со многими достоинствами микропроцессоров и систем на их основе, определяют целесообразность их использования для автоматического и автоматизированного (с участием человека) управления объектами. МикроЭВМ, как и любые другие устройства цифровой обработки сигналов, имеют важные преимущества перед аналоговыми устройствами. Здесь можно упомянуть о высокой стабильности характеристик, отсутствии дрейфа нуля, высокой точности выполнения арифметических операций, малых весе и габаритах, высоком быстродействии, возможности гибкой оперативной перестройки структуры и др. [23].

В настоящее время при создании систем автоматического управления принципиально возможно идти по двум направлениям.



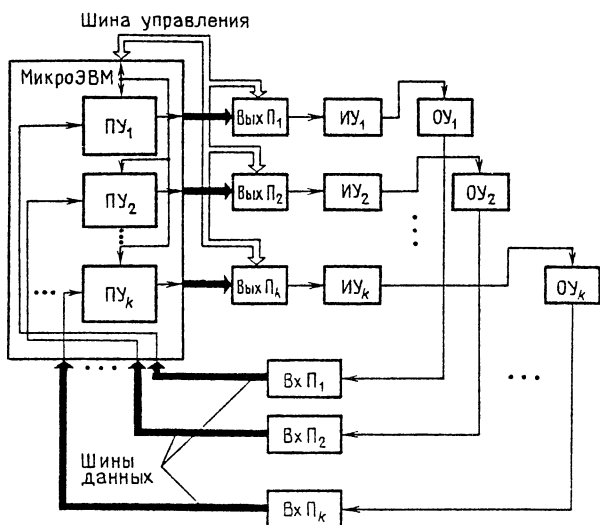
Р и с. 1.1. Структурная схема системы управления с центральной управляющей микроЭВМ

Первое из них связано с использованием центральных управляющих микроЭВМ. Подобные системы могут применяться для управления сложными объектами (самолетами, ракетами, прокатными станами, доменными печами и т.п.) или группами объектов при комплексной автоматизации в различных отраслях промышленности и сельского хозяйства (металлургической, химической, нефтеперерабатывающей, в тепличном овощеводстве, животноводстве и т.д.).

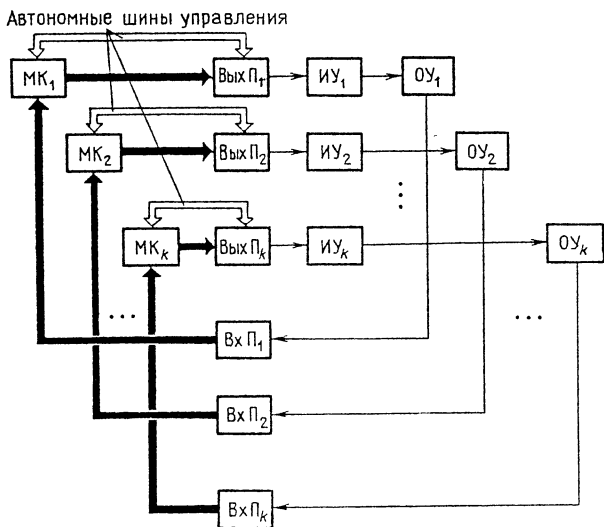
Общая структурная схема автоматической системы с микроЭВМ для этого случая показана на рис. 1.1 [23, 175]. Система содержит ряд входных $ВхП_1, ВхП_2, \dots, ВхП_k$ и выходных $ВыхП_1, ВыхП_2, \dots, ВыхП_k$ преобразователей, обменивающихся сигналами управления с микроЭВМ в процессе преобразования данных по шине управления. Сигналы с выходных преобразователей поступают на исполнительные устройства, воздействующие на объект (или объекты) управления.

Такая система может быть связанной многомерной, если осуществляется управление сложным, многомерным объектом, и несвязанной многомерной, если осуществляется управление группой несвязанных одномерных объектов. В последнем случае САУ с микроЭВМ распадается на совокупность одномерных систем; ее виртуальная (кажущаяся) структура представляется в виде совокупности одноконтурных систем автоматического управления (рис. 1.2), каждая из которых имеет свою программу управления $ПУ_1, ПУ_2, \dots, ПУ_k$.

При управлении сложным объектом или группой объектов процессор обслуживает по очереди отдельные каналы управления.



Р и с. 1.2. Виртуальная структура САУ с центральной микроЭВМ



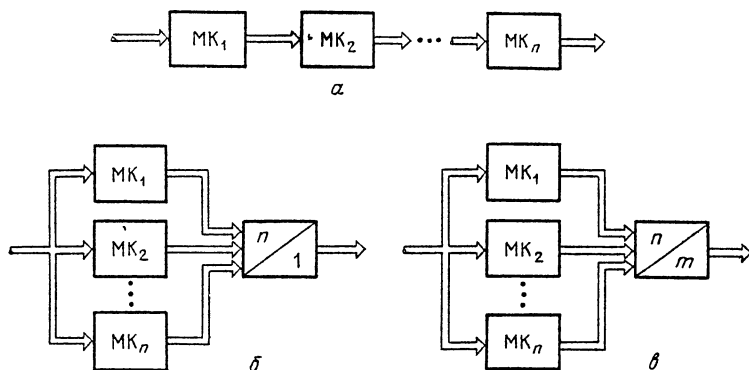
Р и с. 1.3. Схема системы с автономными управляющими микроЭВМ



Р и с. 1.4. Зависимость относительного изменения стоимости микропроцессорных средств в САУ от числа объектов управления



Р и с. 1.5. Зависимость надежности САУ с микроЭВМ от числа каналов управления



Р и с. 1.6. Способы соединения микроконтроллеров в контуре САУ

Эта очередь может осуществляться по жесткой программе или по мере поступления заявок от отдельных каналов с возможностью использования в последнем случае приоритетного обслуживания.

Второе направление, по которому развиваются в настоящее время САУ с микроЭВМ, — это использование в каждом контуре управления автономной микроЭВМ, называемой часто микроконтроллером [158, 231, 232]. Структурная схема САУ с автономными микроЭВМ $МК_1, МК_2, \dots, МК_k$ показана на рис. 1.3. Микроконтроллеры представляют собой упрощенные варианты микроЭВМ, размещаемых в непосредственной близости от управляемого объекта. В микроконтроллерных системах центральная ЭВМ либо отсутствует совсем, либо вводится для передачи ей функций диспетчера или супервизора.

Выбор одного из двух направлений построения САУ с микроЭВМ связан, в частности, с проблемами надежности и стоимости. Стоимость систем, использующих центральную управляющую микроЭВМ, обычно при большом числе управляемых объектов ниже стоимости микроконтроллерных систем (рис. 1.4, где $K_{ц}$ — стоимость систем с центральной управляющей микроЭВМ; $K_{МК}$ — стоимость микроконтроллерных систем) [228]. Эта закономерность, впрочем, с развитием технологии производства микропроцессоров, приведшей к созданию высокоэффективных однокристалльных микроЭВМ, проявляется все в меньшей степени. Кроме того, системы на базе центральных управляющих микроЭВМ являются технологически менее надежными (рис. 1.5). Они нуждаются в дорогостоящих, помехоустойчивых линиях связи. Поэтому принцип децентрализованного (микроконтроллерного) управления в микропроцессорных системах постепенно становится преобладающим.

Микроконтроллерное управление с позиций теории надежности может быть организовано (рис. 1.6) одним из следующих способов: а) управление с конвейерной (последовательной) обработкой информации; б) управление с параллельной обработкой информации; в) мажоритарное управление. Определим для них коэффициент готовности [167]. Коэффициент готовности микроконтроллерной системы определяется по формуле

$$V = T_{ср} / T_0,$$

где $T_{ср}$ — среднее время нахождения систем в работоспособном состоянии; T_0 — общее время работы системы. Если обозначить V_i коэффициент готовности i -го микроконтроллера, то коэффициент готовности системы для каждого из трех способов организации микроконтроллерного управления составляет соответственно [231]

$$V_S = V_i^n, \quad (1.1)$$

$$V_K = 1 - (1 - V_i)^n, \quad (1.2)$$

$$V_M = \sum_{r=m}^n \binom{n}{r} V_i^r (1 - V_i)^{n-r}. \quad (1.3)$$

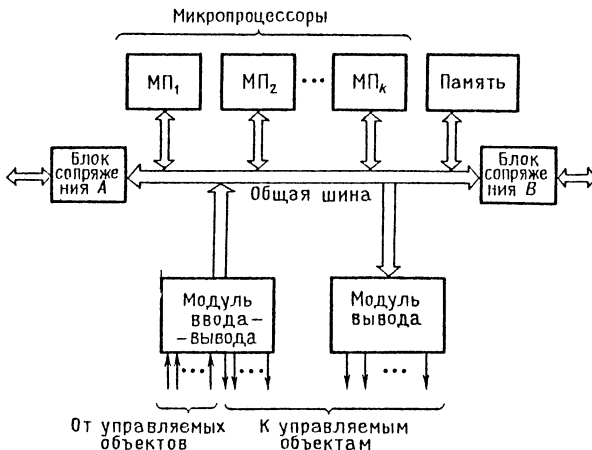
Как следует из соотношений (1.1) — (1.3), наименьший коэффициент готовности обеспечивает последовательный способ организации микроконтроллерного управления. Промежуточное положение по этому параметру занимает мажоритарное микроконтроллерное управление.

Два класса структур цифровых САУ. Для увеличения надежности, производительности и гибкости микроЭВМ в системах автоматического управления применяют многомикропроцессорные

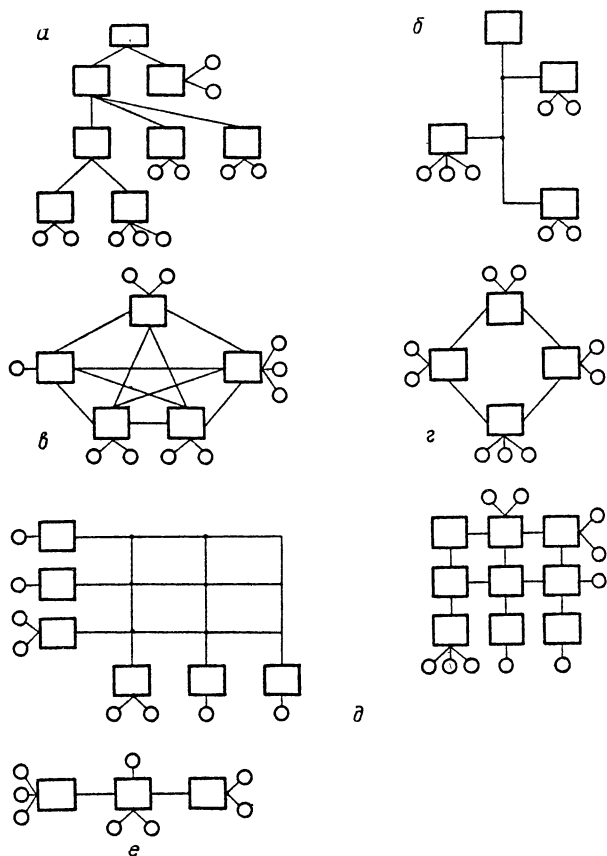
системы и системы на базе многих микроЭВМ – многомашинные системы.

Первый класс структур цифровых САУ характеризуется тем, что большое число микропроцессоров работает на одну общую шину (рис. 1.7). Организация параллельной работы нескольких микропроцессоров позволяет реализовать один из следующих режимов: 1) одновременное выполнение нескольких команд программы; 2) одновременное выполнение нескольких программ или подпрограмм. В первом режиме каждый процессор выполняет команды определенного класса, например, логические команды, арифметические команды, команды по определению адресов и др. Во втором режиме каждый процессор имеет возможность работать с полным списком команд. Оба режима позволяют увеличить производительность управляющих микроЭВМ. Так, например, в микропроцессорной системе [230], состоящей из двух микропроцессоров 18080А, каждый из которых обладает быстродействием 2. млн. операций в секунду, при использовании памяти с временем цикла 850 нс и реализации первого режима работы производительность возросла почти вдвое.

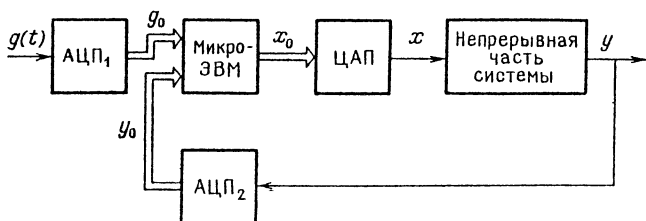
Топология второго класса структур цифровых САУ определяется требованиями надежности, стоимости, гибкости и производительности. Возможные варианты топологии многомашинных систем (соответственно иерархическая, шинная, сетевая, кольцевая, матричная двух типов, последовательная) изображены на рис. 1.8, где прямоугольниками схематически показаны микроЭВМ, а кружками – объекты управления.



Р и с. 1.7. Структурная схема многомикропроцессорной системы



Р и с. 1.8. Топология цифровых САУ



Р и с. 1.9. Одноконтурная система управления с микроЭВМ

Структуры данного класса обладают высокой надежностью. Например, при выходе из строя любой микроЭВМ в системе с сетевой топологией, показанной на рис. 1.8, в, сохраняются по меньшей мере два канала связи со всеми другими микроЭВМ.

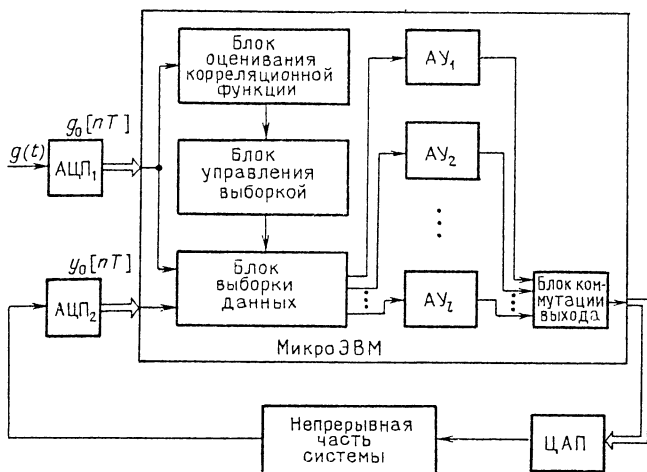
Автоматические системы с перестраиваемой структурой. Типовая структура одноконтурной САУ содержит аналого-цифровые преобразователи АЦП₁ и АЦП₂, цифро-аналоговый преобразователь ЦАП, микроЭВМ и непрерывную часть системы (рис. 1.9) [23]. Кодироваться в АЦП может входное задающее воздействие $g(t)$, представляющее собой желаемое значение управляемой величины $y(t)$, управляемая величина, а также другая поступающая извне информация. В результате такого кодирования на вход микроЭВМ в дискретные моменты времени поступают цифровые представления этих величин g_0 и y_0 . При жесткой программе обслуживания каналов период дискретности оказывается обычно постоянным. При обслуживании по гибкой программе период дискретности оказывается случайной величиной с некоторым математическим ожиданием и дисперсией.

В последнем случае виртуальная структура САУ оказывается переменной: она перестраивается под управлением программы в зависимости от меняющихся условий управления:

Примером САУ с перестраиваемой структурой может быть адаптивная система, работающая с нестационарным сигналом $g(t)$: корреляционная функция этого сигнала зависит от времени. В этом случае период дискретности T сигнала $g(t)$ должен быть переменным. Виртуальная структура САУ для рассматриваемого примера показана на рис. 1.10 [23]. В системе реализованы несколько алгоритмов управления $AУ_1, AУ_2, \dots, AУ_l$, каждый из которых выбирается в зависимости от результата оценивания корреляционной функции входного воздействия $g_0[nT]$ блоками выборки данных и коммутации выхода. Все элементы схемы, расположенные в блоке, названном на рис. 1.10 "микроЭВМ", реализованы программными средствами.

Реализуемость перестройки структуры САУ, осуществляемой в масштабе реального времени на программном уровне, является следствием применения в автоматических системах высокопроизводительных микроЭВМ, на которые возлагаются функции статистического экспресс-анализа случайных процессов, их классификации, идентификации, распознавания образов и многие другие. Это, в конечном счете, улучшает показатели качества управления.

Управляющие микроЭВМ подобного типа в настоящее время приобретают различную специфику в зависимости от той отрасли техники, где они используются (управление движущимися объектами: аэродинамическими, космическими, морскими; управление технологическими процессами и др.). В связи с использованием



Р и с. 1.10. Виртуальная структура адаптивной САУ

микропроцессоров и микроЭВМ интелтуация автоматических систем является в изложенном смысле одним из генеральных направлений их развития.

§ 1.2. ЗАДАЧИ, РЕШАЕМЫЕ МИКРОЭВМ В СИСТЕМАХ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

Основные требования, предъявляемые к САУ с микроЭВМ. В развитии человечества было несколько этапов научно-технических преобразований, связанных с вопросами управления. Первая промышленная революция характеризовалась заменой физического труда человека работой машин. В дальнейшем оказалось, что многие машины, в частности паровые, нуждаются в постоянном управлении. Так появился первый в мире регулятор Дж. Уатта.

Затем были изобретены и начали широко применяться многие другие регулирующие и управляющие автоматические системы, улучшившие качество управления и освободившие человека от выполнения ряда монотонных функций умственной и физической деятельности. Особая роль в этом принадлежит электронным вычислительным машинам (ЭВМ).

Современные САУ с ЭВМ должны удовлетворять общим требованиям, отражающим интересы народного хозяйства. Основными из этих требований являются простота и удобство использования, гибкость, живучесть и экономичность.

Простота и удобство использования связаны с необходимостью освоения систем без привлечения дефицитных, высококвалифи-

цированных специалистов. Сроки обучения специалистов и освоения ими техники должны быть минимальными.

Гибкость систем характеризуется их способностью к модернизации. Известно, что в процессе эксплуатации свойства управляемого объекта, его структура могут изменяться. Это требует изменения алгоритма и внесения поправок в программы управления. Если система управления не предусматривает таких возможностей, то из прогрессивного фактора она становится фактором консервативным.

Понятие "живучесть", являющееся несколько более широким, чем понятие "надежность", связывают с сохранением работоспособности системы не только в нормальных условиях эксплуатации, но и при внешних аварийных воздействиях. При этом допускается некоторое ухудшение качества управления. Живучесть систем обычно обеспечивается введением резервирования, диагностирования и тестирования, правильным построением структуры и изысканием более надежных методов измерения и управления [25].

Экономичность обуславливается малыми капитальными вложениями и малыми эксплуатационными расходами. Известно, что в составе системы управления относительная стоимость технических средств с каждым годом уменьшается, а доля стоимости проектирования и программирования возрастает, достигая 60–90% общих затрат на систему. Некоторые зарубежные фирмы предоставляют пользователям ЭВМ даже бесплатно, требуя оплаты только математического обеспечения. В связи с этим исключительно актуальным становится широкое использование систем автоматизированного проектирования (САПР), снижающих трудоемкость и продолжительность проектирования программного обеспечения.

Изложенным требованиям удовлетворяют системы автоматического управления, построенные на основе микропроцессоров и микроЭВМ.

Широкое применение микропроцессоров обещает революционизировать производство, транспорт, научные исследования. По своему воздействию микропроцессоры не только могут соперничать, но и превосходят своих прославленных предшественников—электронную лампу, транзистор, интегральную логическую схему.

В отличие от обычных вычислительных машин микроЭВМ целиком выполняется по интегральной технологии и может быть расположена на миниатюрном кристалле площадью 20–30 квадратных миллиметров. Такая микроЭВМ легко встраивается в любой объект управления — станок, электрическую пишущую машинку, электронные весы, кассовый аппарат, кухонную печь, в автомобиль, в светофор, в сложный научный прибор и во многие другие устройства, возможности которых в результате существенно расширяются.

Микропроцессорные системы с позиций пользователя просты и удобны в эксплуатации. Они высоконадежны, самотестируемы и ремонтпригодны. Высокая технологическая надежность обеспечивается малым числом выводов БИС при высокой степени их интеграции. Самотестируемость является следствием программируемости. Она реализуется путем решения контрольных задач с выводом результатов проверок на индикацию. Контрольные (тестовые) задачи могут быть решены также по запросу пользователя. Ремонтпригодность обуславливается модульным принципом конструирования, позволяющим производить оперативную замену неисправных блоков.

При работе с микроЭВМ, включенной в контур САУ, специалист имеет, по существу, в своем распоряжении высококвалифицированного "партнера", берущего на себя многие функции по поиску и локализации неисправностей, принятию решений, хранению справочных сведений и других данных, обеспечивающих достижение высокой эффективности эксплуатации автоматических систем.

Программируемость микропроцессорных систем, перестраиваемость их структуры определяют высокую гибкость САУ с микроЭВМ, их способность к модернизации. САУ с микроЭВМ могут быть легко приспособлены для решения новых или модифицированных задач. Вносимые при этом изменения в микроЭВМ сводятся зачастую к замене одной БИС памяти на другую. В микроЭВМ с перестраиваемой структурой изменение структуры происходит автоматически в процессе решения задачи [174, 177]. Большими возможностями по адаптации к решаемой задаче обладают многомикропроцессорные вычислительные системы [109].

Большая гибкость представляет собой одно из основных достоинств микропроцессорных систем. Это свойство обеспечивает возможность внесения изменения в структуру и в программу работы системы при ее испытании и наладке, а также в процессе эксплуатации.

Высокая живучесть САУ с микроЭВМ объясняется многими причинами, из которых отметим две: наличие программной и аппаратурной избыточности в сочетании с высокой технологической надежностью элементной базы.

МикроЭВМ, составляющие основу соответствующих автоматических систем, являются относительно дешевыми техническими средствами. Это связано с высокой технологичностью их массового производства, высокой надежностью и меньшими затратами на поддержание систем в работоспособном состоянии.

Применение систем с программируемой логикой (т.е. систем на базе микроЭВМ) в сравнении с системами на основе аппаратной (жесткой) логики позволяет сократить сроки разработки новой

техники, повысить качество решения проектных задач. Особенно сильно это проявляется там, где требуется создать систему, подобную существующим, но с отличающимися свойствами. Если разработчик приобрел опыт при проектировании одной системы на основе микроЭВМ, то разработка последующих систем существенно ускорится.

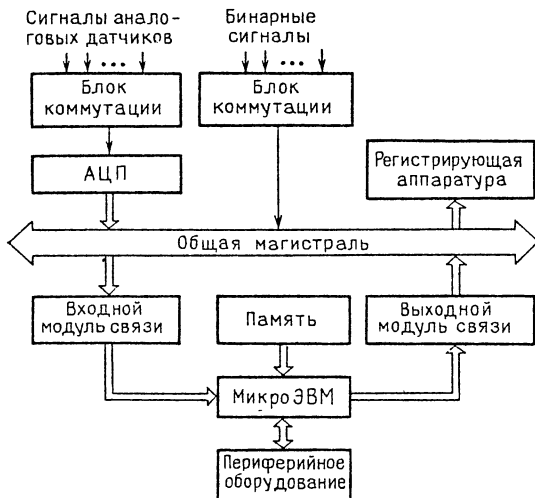
Отмеченные характеристики микропроцессорных систем, сочетающиеся с низким энергопотреблением и малогабаритностью, определяют целесообразность применения микропроцессоров и микроЭВМ в совершенно новых областях науки и техники, т.е. там, где ранее применение вычислительных машин было неприемлемым по соображениям стоимости, надежности, размеров и потребления энергии.

Наряду с нетрадиционными на микропроцессоры и микроЭВМ все более широко возлагаются традиционные задачи управления, решавшиеся ранее схемами с жесткой логикой. Появилась возможность реализации в автоматических системах алгоритмов, приближающихся по своим характеристикам к оптимальным. Намечился сдвиг в стратегии управления в направлении децентрализации.

МикроЭВМ в промышленных системах автоматического управления и регулирования. Внедрение микроЭВМ в системы управления технологическими объектами и процессами наряду с простой заменой элементной базы и некоторым расширением функций аппаратуры создает также принципиально новые возможности в построении децентрализованных или распределенных комплексов. При этом децентрализация предполагает не только рассредоточение аппаратуры, но и распределение функций обработки информации и управления между автономными микроЭВМ. Такие системы могут включать в свой состав и центральную ЭВМ, выполняющую функции оптимизации, учета, планирования и координации работ.

Построение промышленных систем автоматического управления на базе микроЭВМ осуществляется обычно в два этапа. Первый этап соответствует переводу на микропроцессорную технику только центральной части средств автоматического регулирования. На втором этапе аналоговые датчики и исполнительные механизмы заменяются цифровыми [175]. Оба эти этапа при проектировании новых систем управления технологическими объектами и процессами целесообразно совместить, так как экономически более выгодным является создание нового производства, чем его модернизация.

Примером реализации централизованного автоматического контроля и управления является комплекс А360-34 с базовой моделью А330-25, выполненный на микроЭВМ "Электроника С5-02" [203]. Комплекс обеспечивает наряду со сбором, измерением, обработкой



Р и с. 1.11. Структурная схема модуля А330-25

и регистрацией параметров управляемых процессов также формирование оптимальных воздействий, поступающих на объект управления.

Базовое микропроцессорное устройство А330-25 выполнено по схеме, показанной на рис. 1.11. В состав устройства входят блок коммутации аналоговых сигналов с 64 входами, блок коммутации бинарных сигналов с 40 входами, аналого-цифровой преобразователь, входной и выходной модули связи, перепрограммируемая память, периферийное и регистрирующее оборудование, обеспечивающее ввод данных и программ с фотосчитывающего устройства, телетайпа, вывод информации на перфоленту, видеоконтрольное устройство и на алфавитно-цифровое печатающее устройство.

В комплексе А360-34 используются два базовых модуля А330-25. Они соединены таким образом, что могут работать в режиме взаимодополнения для увеличения информационной и вычислительной мощности и в режиме резервирования. В этом комплексе на базе одного центрального обрабатывающего устройства реализован принцип многоконтурного управления. Высокая надежность обеспечена введением резервирования и возможностью перехода на ручное управление.

В многоконтурных системах автоматического управления важной является задача сбора информации от многих десятков и сотен датчиков.

Для дистанционного сбора информации используются так называемые вынесенные измерительные станции [149]. Такие станции

могут обрабатывать входные сигналы от аналоговых датчиков и передавать выходные цифровые сигналы по недорогой линии в виде скрученной пары проводов в центральный процессорный блок.

Вынесенные измерительные станции имеют собственный микропроцессор, мультиплексор и аналого-цифровой преобразователь. Вследствие этого система может обходиться без сложных и дорогих кабельных линий и схем обработки сигналов, необходимых для передачи аналоговых входных сигналов в центральный блок обработки данных.

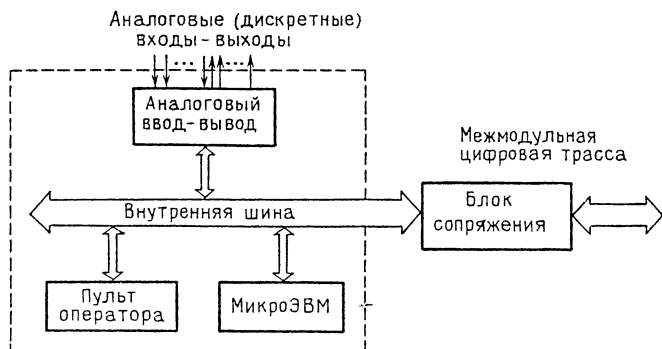
Вынесенные микропроцессоры позволяют также разгрузить центральный процессорный блок от многих дополнительных задач, связанных с реализацией локальных функций управления в одноконтурных регуляторах.

Одноконтурные регуляторы могут быть реализованы с использованием разных принципов построения. Наибольшее распространение получили принципы, положенные в основу организации регулирующих микроконтроллеров — ремиконтов [105].

Ремиконт является центральным звеном системы автоматического регулирования. Входные цепи ремиконта рассчитаны на подключение аналоговых и дискретных датчиков, а его выходные цепи формируют аналоговые и дискретные сигналы для управления исполнительными устройствами.

Поступающая в ремиконт аналоговая и дискретная информация преобразуется в цифровую форму, обрабатывается в микроЭВМ, после чего при необходимости осуществляется процесс обратного преобразования. Программируется ремиконт с помощью специализированной клавишной панели.

Указанные три вида устройств — средства аналогового ввода-вывода, микроЭВМ, пульт оператора — являются основой аппаратного обеспечения ремиконта (рис. 1.12). Средства аналогового ввода-



Р и с. 1.12 Структурная схема ремиконта

вывода содержат узлы гальванического разделения сигналов, узел мультиплексирования аналоговых входных сигналов и группу преобразователей: АЦП, ЦАП, дискретно-цифровой (ДЦП) и цифро-дискретный (ЦДП) преобразователи. Узлы гальванического разделения подавляют помехи общего вида и обеспечивают работу с источниками информации, находящимися под различными потенциалами. Назначение мультиплексора и преобразователей является традиционным. Особенность ремиконта проявляется в том, что на его выходе не используется мультиплексирование: число ЦАП равно числу выходных цепей контроллера. Такое построение ремиконта связано с необходимостью запоминать значение каждого управляющего сигнала после прекращения вычислительного процесса. Учитывая, что в САУ число выходных сигналов ремиконта невелико, затраты на ЦАП и АЦП оказываются относительно небольшими.

Перечисленные узлы ввода-вывода аналоговой и дискретной информации позволяют сопрягать ремиконт с аналоговыми и дискретными датчиками, с исполнительными механизмами пропорционального, позиционного, интегрирующего и другого действия, а также с различными устройствами дискретного и логического управления.

МикроЭВМ ремиконта имеет традиционную компьютерную организацию. Особенностью микроЭВМ является то, что большая часть его памяти программируется на заводе-изготовителе, отсутствуют

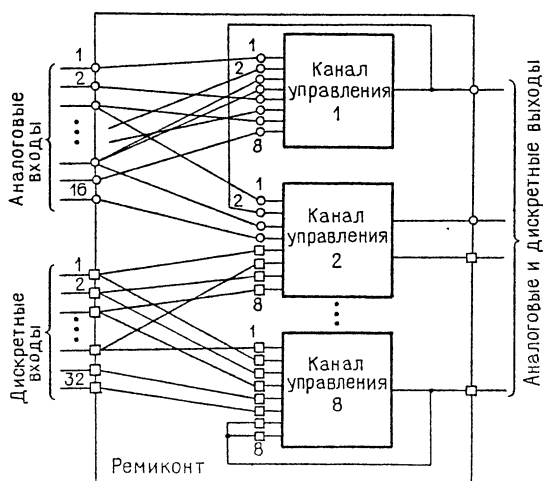


Рис. 1.13. Виртуальная структура восьмиканального варианта ремиконта

обычные средства ввода и отладки программ, а также модули сопряжения с ними. Указанная особенность вычислителя позволяет упростить ремиконт и сделать рентабельным его применение для обработки сравнительно небольших массивов информации.

Пульт оператора в ремиконте используется для установки требуемой конфигурации регулирующего контура, выбора алгоритма управления, контроля значений технологических переменных, оперативного вмешательства в процесс управления и других целей.

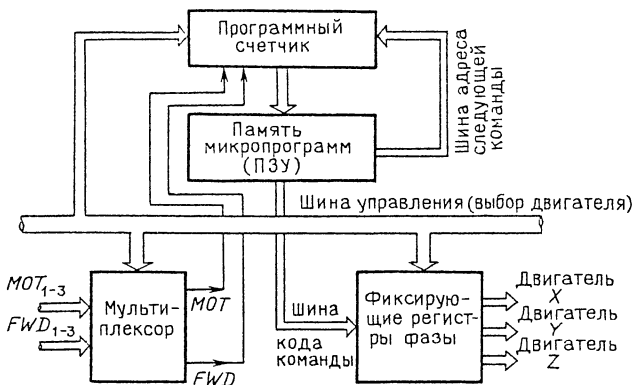
Программное обеспечение ремиконта состоит из программ диспетчера, координирующего весь вычислительный процесс, рабочих программ, программ обслуживания пульта и диагностической программы. Для программирования используется десятичный код, набираемый на панели пульта.

Все алгоритмы ремиконта достаточно универсальны и в функциональном отношении эквивалентны типовым звеньям системы автоматического регулирования или типовой "связке" таких звеньев. Структура ремиконта имеет ту особенность, что она описывает свойства контроллера в традиционных для систем регулирования понятиях, основными из которых являются каналы управления, с системной точки зрения эквивалентные отдельному прибору или типовому сочетанию приборов аналоговой системы регулирования, и конфигурация, определяющая систему связи каналов со входами и выходами ремиконта, а также варианты взаимодействия каналов.

На рис. 1.13 показан восьмиканальный вариант структуры ремиконта. В распоряжение потребителя он предоставляет 16 аналоговых и 32 дискретных входа, 8 автономных каналов управления с 8 входами в каждый канал. Аналоговые и дискретные выходы каналов жестко связаны с соответствующими выходами ремиконта. Остальные связи, образующие конфигурацию, могут устанавливаться при программировании и задаются с пульта оператора.

По оценкам специалистов имеется ограниченное число (порядка 20–25) алгоритмов, комбинация которых позволяет автоматизировать управление процессами и объектами практически любой степени сложности. Эти алгоритмы, оформленные в виде библиотеки программ, хранятся в постоянной памяти и могут быть использованы в любом заданном сочетании. В результате существенно упрощаются наладка и эксплуатация автоматических систем, построенных на основе ремиконта.

Основные принципы, составляющие идеологию ремиконтов, реализованы в децентрализованной микроконтроллерной системе TDC 2000 [73], в системе прямого цифрового управления Tosdic-200, распределенной микропроцессорной системе управления Σ Line UNITROL [175]. Регулирующие микроконтроллеры



Р и с. 1.14: Схема специализированного микроконтроллера

являются в значительной степени универсальными- техническими средствами широкого применения.

Для реализации конкретных функций управления используются специализированные микроконтроллеры. Они имеют структуру и систему команд, оптимизированные на некоторый класс применения. Так, схема управления тремя электродвигателями [113] позволяет создать компактную и чрезвычайно гибкую систему позиционного управления. Основу схемы составляют программный счетчик и память микропрограмм (рис. 1.14). Программный счетчик обеспечивает управление каждым двигателем, которое зависит от информации, хранимой в ПЗУ. В последнем хранятся коды программных последовательностей прямого и обратного хода, а также коды выбора и отключения каждого двигателя и командного поля для программного счетчика. Каждый двигатель имеет собственные коды последовательностей шагов программы, хранимые в ПЗУ, так что не требуется, чтобы все три шаговых двигателя были одинакового типа и имели совпадающие законы управления.

Память микропрограмм хранит слово длиной в 8 разрядов, включающее код команды P_0, P_1, P_2 , код адреса следующей команды P_3, P_4, P_5 и код выбора двигателя P_6, P_7 . Информация, задающая направление вращения каждого двигателя, вводится в программный счетчик по линиям FWD_1-FWD_3 мультиплексора. Активное состояние каждого двигателя определяется линиями MOT_1-MOT_3 . Схема синхронизируется с частотой, зависящей от требований к системе и параметров шаговых двигателей.

Специализированные микроконтроллеры редко используются автономно. В основном они работают совместно с универсальными микропроцессорными системами общего назначения в качестве

дополнительного периферийного микроконтроллера, управляются ими и увеличивают их функциональные и вычислительные возможности.

Специализированные микроконтроллеры открывают новые возможности в построении адаптивных и обучающихся роботов, гибких автоматических и автоматизированных производств [122, 152, 175].

Применение микроЭВМ в авиационно-космической технике. Значительная доля применений микропроцессоров и микроЭВМ приходится на авиационно-космическую технику [122, 125, 184, 191]. МикроЭВМ могут выступать при этом в роли "интеллектуальных" датчиков, осуществляющих адаптивное выделение сигналов из шумов и распознавание объектов земной поверхности, преобразователей координат и формирователей управляющих воздействий на исполнительные органы и многого другого [59, 64, 148, 205].

В качестве примера рассмотрим цифровую систему управления ориентацией космического летательного аппарата (КЛА) [234]. Она предназначена для одноосной ориентации КЛА. Рабочим телом в системе является реактивный маховик. Его динамика задается равенством

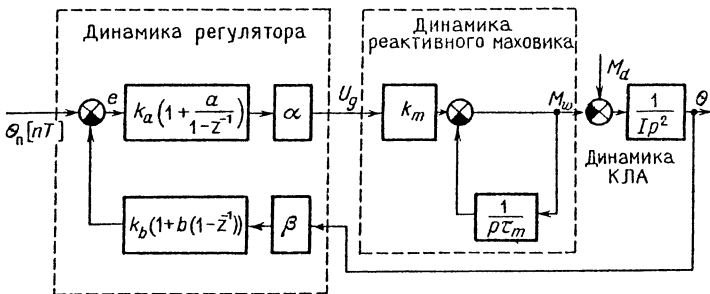
$$M_w = k_m U_g - M_w / \rho \tau_m,$$

где M_w — момент вращения маховика; k_m , τ_m — соответственно коэффициент передачи и постоянная времени электродвигателя, применяемого в реактивном маховике; U_g — напряжение на входе электродвигателя.

Уравнение динамики КЛА имеет вид

$$I p^2 \Theta = M_d - M_w,$$

где I — момент инерции КЛА; Θ — угол поворота оси стабилиза-



Р и с. 1.15. Модель системы ориентации космического летательного аппарата

ции КЛА относительно заданного направления; M_d – момент вращения, создаваемый возмущениями.

Алгоритм работы цифрового регулятора задается уравнениями

$$e[n] = \Theta_n[n] - \gamma_1 \Theta[n] - \gamma_2 \Theta[n] + \gamma_2 \Theta[n-1],$$

$$U_g[n] = \gamma_3 e[n] - \gamma_3 e[n-1] - \gamma_4 e[n] + U_g[n-1], \quad (1.4)$$

$$\gamma_1 = k_b \beta, \quad \gamma_2 = k_b \beta b, \quad \gamma_3 = k_a \alpha, \quad \gamma_4 = k_a \alpha a,$$

где $e[n]$ – ошибка ориентации; $\Theta_n[n]$ – заданное значение угла ориентации; $\Theta[n]$ – фактическое угловое положение КЛА; b – коэффициент передачи цепочки дифференцирования; k_b – масштабирующий коэффициент цепочки дифференцирования; β – коэффициент передачи АЦП; $U_g[n]$ – напряжение на выходе цифрового регулятора; a – коэффициент передачи цепочки интегрирования; k_a – масштабирующий коэффициент цепочки интегрирования; α – коэффициент передачи ЦАП.

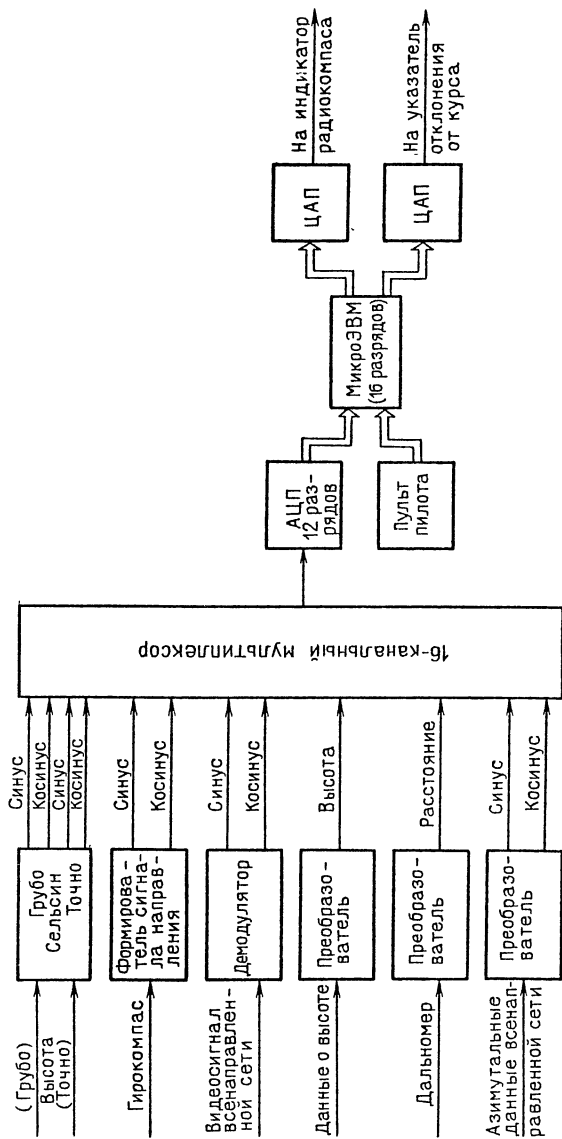
Структурная схема системы ориентации показана на рис. 1.15.

При использовании в системе ориентации микропроцессора разностные уравнения (1.4) реализуются на программном уровне.

Применение микропроцессора в системе ориентации КЛА позволяет по сравнению с использованием традиционных аналоговых регуляторов повысить точность управления, гибкость и надежность системы. Уменьшаются также габариты, масса, энергопотребление, что особенно важно в условиях использования аппаратуры на борту КЛА.

Еще один пример использования микроЭВМ связан с навигацией самолетов. В системе R-Nav микроЭВМ управляет автоматическим полетом по прямому курсу к месту назначения [215]. В обычном ручном режиме управления пилоты ведут самолеты по зигзагообразному курсу, от одного опорного узла сети радиомаяков до другого. Для вычисления прямого курса требуется достаточно мощное малогабаритное оборудование, способное производить вычисления в реальном масштабе времени. Таким оборудованием является микроЭВМ IMP-16, входящая в систему R-Nav.

При работе с микроЭВМ пилот задает с пульта конечную точку маршрута, а также азимутальные данные и частоты опорных пунктов сети радиомаяков, расположенных по курсу. Он может ввести также другие параметры, например, необходимую высоту в какой-то конкретной точке маршрута. В процессе полета с приемника сигналов радиомаяков и с дальномера (рис. 1.16) поступают значения реальных параметров для сравнения с параметрами запланированного курса полета. На основании этих данных микроЭВМ производит все вычисления, необходимые для ручного



Р и с. 1.16. Структурная схема радионавигационной системы

или автоматического управления полетом (в частности азимутальные данные и расстояния до узлов сети), что позволяет пилоту лететь по прямой к пункту назначения.

Поскольку радионавигационная система R-Nav взаимодействует с существующей самолетной измерительной аппаратурой, имеющей другие входные и выходные сигналы, многие аппаратные решения по сопряжению элементов заменены здесь подпрограммами работы микроЭВМ. Это сводит к минимуму объем и сложность схем формирования сигналов. Такой подход позволяет не только минимизировать габариты, но также делает возможным вместо многократно повторяющихся затрат на изготовление технических средств ограничиться разовыми затратами на разработку программного обеспечения.

Функциями микроЭВМ в системе R-Nav являются вычисление арктангенса угла полета, отклонения от желаемого курса полета, азимута и расстояния между самолетом и очередной точкой маршрута (точкой, где курс полета пересекает луч, исходящий из опорного пункта сети под углом 90°), времени полета до очередной точки маршрута, расстояния и времени полета до пункта назначения, скорости полета, отклонения от желаемой высоты полета. МикроЭВМ реализует также функции самоконтроля и контроля устройств индикации.

Навигационное устройство на базе микроЭВМ IMP-16, содержащее все аналоговые интерфейсы, источники питания и т.п., выполнено в виде блока с габаритными размерами $316 \times 190 \times 120$ мм. Его применение позволило снизить эксплуатационные расходы более, чем на 10%.

Развитие техники микропроцессоров и микроЭВМ оказывает значительное влияние на процесс усовершенствования тактических ракет [154]. Микропроцессоры позволяют не только существенно расширить возможности инерциальных систем наведения на маршевом участке траектории, но и упростить электронные схемы головок самонаведения на конечном участке, обеспечить их более полное сопряжение с инерциальным блоком.

Например, в системе Northrop, используемой для наведения тактических ракет, микропроцессор рассчитывает углы ориентации блока наведения относительно связанной системы координат ракеты, определяет изменение скорости ракеты, вырабатывает на маршевом участке полета команды для исполнительных устройств системы управления.

Приведенный перечень задач, решаемых микроЭВМ в системе управления, соответствует традиционным функциям, выполняемым блоками САУ тактических ракет.

Система AQS 901 представляет собой бортовой комплекс самолета противолодочной обороны. Она выполнена на базе микропро-

цессора Am 2901 и обладает высокой чувствительностью и гибкостью применения. Система работает в комплексе с пассивным звуколокационным бумом направленного действия, обеспечивающим пеленгование цели, позиционную засечку в гиперболической системе координат и обработку значений доплеровского сдвига частоты. Акустические сигналы этой аппаратуры по каналу радиотелеметрии посылаются на борт самолета и подвергаются обработке. Функции обработки сигналов распределены между 30 секционными микропроцессорами, каждый из которых решает вполне определенную задачу. Так, например, один из процессоров выполняет быстрое преобразование Фурье. Этот процессор способен выполнить 2048 сложных преобразований в течение 1,25 мс. Благодаря такой организации многопроцессорной вычислительной системы достигается высокое быстродействие и большая эксплуатационная гибкость [152].

МикроЭВМ в автоматизированных системах научных исследований. В настоящее время существует тенденция к автоматизации управления экспериментом, регистрации данных в цифровой форме и обработки их при помощи ЭВМ [12].

Программируемая система управления экспериментом [232] выполнена на основе микропроцессора I8080A. В ее состав входят одноплатная микроЭВМ, где кроме процессора смонтированы тактовый генератор, ППЗУ на 4096 байтов, ОЗУ на 1024 байта, БИС интерфейса для 4 каналов ввода-вывода, БИС последовательного интерфейса (рис. 1.17), а также плата ОЗУ емкостью 2048 байтов.

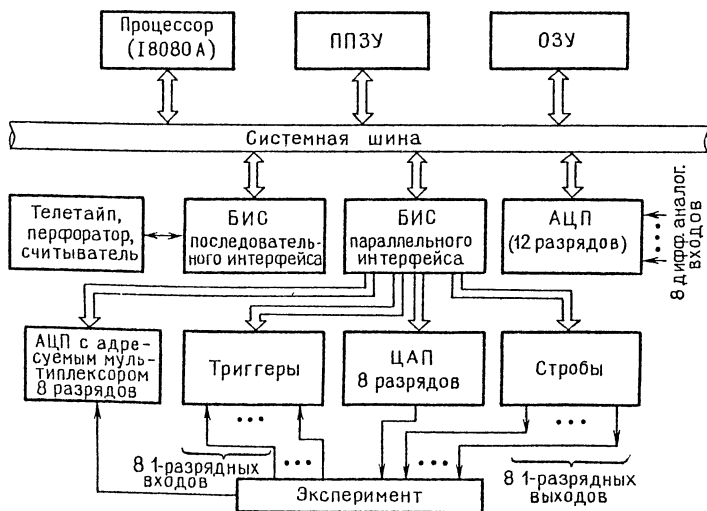


Рис. 1.17. Схема системы управления экспериментом

Работа системы управления экспериментом производится под действием команд программы, организованной в виде двух отдельных частей: исполнительной и системной.

Исполнительная программа (монитор) содержит подпрограммы связи с телетайпом и загрузки программ на машинном языке в ОЗУ для их отладки.

Системная программа используется для составления и исполнения программ пользователя на специальном языке программ пользователя (ЯПП), который является языком высокого уровня, предназначенным для управления аппаратурой. Системная программа может работать в трех режимах: управления, проверки формата и исполнения.

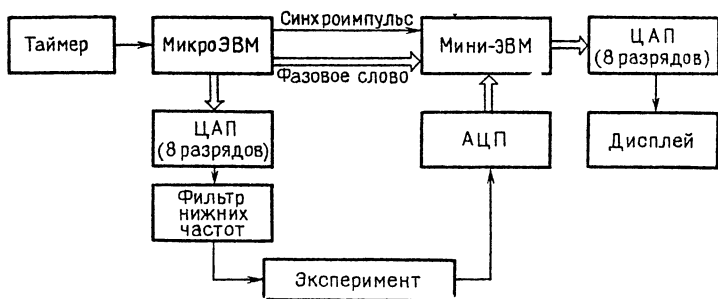
В режиме управления в микроЭВМ вводится программа пользователя. Режим проверки формата используется для обнаружения синтаксических ошибок. После исправления всех ошибок система переходит в режим исполнения. Исполнение программы пользователя осуществляет интерпретатор, который транслирует команды программ пользователя в машинные команды. Существует несколько условий, которые при работе интерпретатора могут вызвать печать сообщения об ошибке или переход в режим управления для диагностирования характера и места ошибки и внесения соответствующих исправлений.

Программируемая система управления экспериментом эффективно использовалась в качестве дискретизатора кривых и графопостроителя, таймера, для управления работой шагового электродвигателя, для обработки сигналов, выполнения некоторых математических и др. операций, связанных с проведением и анализом результатов экспериментальных исследований.

Достоинством системы является ее функциональная ориентация на управление экспериментом, что проявилось в чрезвычайной простоте ее программирования для работы с различными внешними устройствами, участвующими в научном исследовании.

Еще одна из разновидностей программируемой системы управления экспериментом связана с гармоническим анализом сигналов, обусловленных эффектом Хааса — Ван Альфена в металлах. Схема системы показана на рис. 1.18 [43]. В состав схемы входят две ЭВМ: микроЭВМ и мини-ЭВМ. МикроЭВМ генерирует синусоидальный сигнал модуляции и когерентные синхроимпульсы, а также передает фазовую информацию по шине данных. Мини-ЭВМ ведет накопление данных, характеризующих исследуемый сигнал, выполняет усреднение сигнала и вычисляет спектр сигнала, включающий 16 гармоник основной частоты.

В качестве микроЭВМ используется система Cromemco Z-2 Z-80. Для проведения гармонического анализа микроЭВМ генерирует синусоидальный сигнал модуляции и вырабатывает последователь-



Р и с. 1.18. Система управления физическим экспериментом

ность когерентных синхроимпульсов с частотой, в 32 раза превосходящей частоту модуляции. Генерируемый синусоидальный сигнал состоит из 32 ступенек, формируемых табличным методом.

При каждом прерывании от таймера микроЭВМ переходит к программе выборки очередного значения синусоиды. При этом она извлекает из памяти также информацию о частоте и фазе синусоиды. Полученное фазовое слово, характеризующее частоту и фазу синусоиды, и синхроимпульс передаются в мини-ЭВМ.

Получив синхроимпульс от микроЭВМ, мини-ЭВМ запрашивает отсчет у быстродействующего АЦП и принимает фазовую информацию от микроЭВМ. Отсчет, выданный АЦП, записывается в ячейку ассоциативной памяти, номер которой соответствует фазовому слову. 32 отсчета, соответствующих одному периоду сигнала, образуют группу. Каждая последующая группа отсчетов записывается в те же ячейки памяти, добавляясь к уже имеющемуся содержанию этих ячеек. Этот процесс повторяется до завершения 16 периодов модуляции, после чего мини-ЭВМ перестает реагировать на синхроимпульсы от микроЭВМ и переходит к выполнению преобразования Фурье с расчетом 16 гармоник, причем используются арифметические операции двойной точности. Результаты анализа выводятся на дисплей.

Благодаря высокому качеству графического изображения системе можно использовать не только для гармонического анализа в ходе эксперимента, но и для настройки экспериментальной установки путем оптимизации параметров модулирующего сигнала.

Автоматическое управление параметрами сигналов в ходе эксперимента по исследованию эффекта Керра осуществляется в установке, схема которой показана на рис. 1.19 [1]. Установка предназначена для исследования двойного лучепреломления и дихронизма проводящих растворов под действием электрического поля высокого напряжения. МикроЭВМ решает две задачи: управление параметрами высоковольтных импульсов и контроль темпе-



Р и с. 1.19. Состав аппаратуры для формирования высоковольтных импульсов и контроля температуры

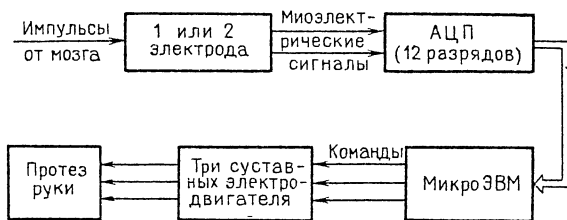
ратуры в ячейке. Управление параметрами последовательности импульсов производится во избежание чрезмерного нагрева раствора.

Авторы работы [47] отмечают также еще одну возможность, предоставляемую использованием микроЭВМ, — возможность применения статистических методов обработки результатов экспериментальных исследований.

Другие области применения микроЭВМ в автоматических системах. Важной областью использования микроЭВМ являются автоматические системы медицинской техники. Функции микроЭВМ в таких системах чрезвычайно разнообразны. Рассмотрим некоторые из них.

В протезе руки [160] микроЭВМ распознает принятый от мозга электрический импульс, называемый миоэлектрическим сигналом, формирует команды управления двигателями локтевого и лучезапястного суставов, а также суставов кисти (рис. 1.20).

МикроЭВМ построена на базе микропроцессора 18080 и памяти емкостью 2К байт. Эта система управления способна распознавать команду мозга и приводить в действие электродвигатели за 0,2 с. Наибольшей трудностью при создании протеза руки было, по мнению авторов, создание программных средств, обеспечивающих распознавание и быструю реакцию на различные импульсы мозга.



Р и с. 1.20. Система управления протезом руки

Система имеет программное обеспечение, решающее задачу идентификации функций различных суставов и превращающее миоэлектрические сигналы в команды.

Для людей, утративших возможность пользоваться своими конечностями, разработана система, управляемая свистом [49]. Эта система может помочь им включать свет, переворачивать страницы книги или набирать номер телефона, а также совершать другие действия. Идея ее построения заключается в том, что каждому из управляемых объектов присваивается номер. Эти номера последовательно автоматически воспроизводятся на светодиодном индикаторе передней панели системы. Когда на индикаторе появляется номер, соответствующий объекту, пользователь просто свистит. Система воспринимает свист и включает нужный прибор.

Микропроцессор в такой системе обеспечивает большую гибкость, реализуя, например, хранение номеров вызываемых телефонов, повышение помехоустойчивости системы и ее адаптацию к пользователю.

В приведенных примерах в контур управления введен человек. Поэтому эти системы не являются автоматическими – их называют автоматизированными. Далее приводится пример автоматической системы управления солнечным обогревом с жидкостным теплоносителем [155]. Система обеспечивает поддержание нормальных условий в отапливаемых помещениях и экономный расход тепла. Число каналов управления равно 16.

В состав системы входят микроЭВМ, аналого-цифровой преобразователь, цифровой индикатор, клавиатура, преобразователь команд управления.

В памяти микроЭВМ, включающей ОЗУ емкостью 1К байт и ППЗУ емкостью 4К байта, хранится программа решения системы логико-арифметических уравнений. Коэффициенты уравнений находятся в ОЗУ. Модификация коэффициентов может производиться через клавиатуру, что позволяет видоизменять систему без аппаратурной перестройки.

В режиме управления обогревом на вход АЦП поступают сигналы от термисторных датчиков температуры. Цифровые бинарные сигналы от контактов переключателей и клавиатуры поступают в микроЭВМ напрямую – через 16 входов БИС интерфейса. Результатом работы программы являются сигналы двух видов: цифровые данные о температуре и дискретные управляющие сигналы, поступающие на электромеханические реле или твердотельные переключатели, которые обеспечивают включение и выключение насосов и открытие (закрытие) клапанов системы теплоснабжения.

Данная система может использоваться также для управления другими объектами, где требуется на основе сигналов множества датчиков формировать дискретные управляющие сигналы. Пример

другой системы управления обогревательными приборами содержится в работе [224].

Использование микропроцессоров и микроЭВМ в системах рационального использования топливно-энергетических ресурсов является особо актуальным. Выравнивание нагрузок, отключение второстепенных нагрузок в заданное время с автоматическим включением по программе, отключение нагрузок при приближении расхода энергии к запрограммированной величине, равномерное распределение нагрузки в течение заданных периодов времени и т.п. обеспечивают уменьшение потребления энергии на 20–30%. Применение микропроцессоров в системах управления отопительно-производственными котельными крупными зданиями позволяет сократить потребление энергии на 30% [176].

В СССР создается отраслевая система автоматизированного управления энергетикой страны [14]. В состав системы входят различные устройства и системы автоматического управления и регулирования центрального, системного, станционного и агрегатного уровней.

В настоящее время микропроцессорная техника стала базой для автоматизации технологических процессов энергетики на качественно новом техническом уровне. Одними из первых объектов автоматизации стали энергетические агрегаты: турбина и генератор. Это обусловлено требованиями расширения функциональной полноты алгоритмов управления, повышения качества производимой энергии, сокращения объема и материалоемкости аппаратуры, повышения надежности работы силового энергетического оборудования и др. При этом движущим фактором внедрения микроЭВМ в системы управления генерирующими агрегатами являются их функциональные преимущества.

Функции микроЭВМ в системах автоматического регулирования возбуждения энергетических агрегатов сводятся, например, к следующим: обеспечение дистанционного управления автоматическим регулятором возбуждения (с центрального пульта); передача информации о режиме работы агрегата на центральный пульт управления; обеспечение заданных показателей качества регулирования возбуждения в экстремальных режимах; взаимодействие с системой автоматического регулирования частоты и активной мощности агрегата; самоконтроль и сигнализация о неисправностях. Перечисленные функции являются в определенной степени нетрадиционными. Наряду с ними сохраняются и традиционные функции: стабилизация напряжения на выходах генератора, обеспечение статической и динамической устойчивости генератора, автоматизация управления током возбуждения синхронного генератора, распределение реактивной нагрузки между параллельно работающими генераторами и др.

Цифровой регулятор возбуждения, выполненный на базе двух микроЭВМ "Электроника 60" [14], имеет развитую структуру ремиконта (рис. 1.12). Датчики измерительной информации, подключаемые к блоку аналогового ввода, представлены группой преобразователей (измерительным преобразователем напряжения, измерительным преобразователем частоты и т.д.).

Для связи с центральным пультом управления в блоке сопряжения предусмотрен специальный регистр, обеспечивающий, в частности, возможность ввода команд от обычных контактных устройств (ключей, реле), а также от противоаварийной автоматики. Дистанционная перестройка коэффициентов регулирования осуществляется по межмодульной цифровой трассе с центрального пульта. Реализованы следующие основные режимы управления возбуждением: пуск, синхронизации, регулирование напряжения, регулирование тока возбуждения. Программное обеспечение включает загрузочный модуль, модуль основного времени и модуль резервного времени.

Загрузочный модуль служит для подготовки микроЭВМ к основному циклу вычислений. После включения таймера начинает работать программа основного времени, которая содержит все необходимые операции по обработке информации и управлению за один проход. К модулю резервного времени относятся все программы, организующие технологические и защитные функции регулятора. Программы модуля резервного времени обладают меньшим приоритетом, чем программа основного времени.

Введение в контур регулирования микроЭВМ позволило улучшить характеристики системы регулирования возбуждения генераторов мощных электростанций (табл. 1.1) [14].

Т а б л и ц а 1.1

Показатели	Прототип без микроЭВМ	Регулятор с микроЭВМ
Разрешающая способность по напряжению, %	1	0,06
Допускаемое отклонение реактивной мощности при отсутствии группового регулятора напряжения, %	50	менее 1
Время уравнивания напряжения при синхронизации, с	20	0,5
Дистанционная настройка регулятора	нет	есть
Погрешность задания уровня напряжения, %	до 1	0,06
Самонастройка канала стабилизации	нет	есть
Возможность перехода в режим регулирования тока возбуждения	нет	есть

Т а б л и ц а 1.2

Элемент САУ	Функции микроЭВМ
1. Датчик	<p>Линеаризация амплитудной характеристики датчика</p> <p>Масштабирование сигналов</p> <p>Аналого-цифровое преобразование</p> <p>Коррекция погрешностей аналого-цифрового преобразования</p> <p>Сжатие и помехоустойчивое кодирование данных</p> <p>Фильтрация сигналов из помех</p>
2. "Интеллектуальный" датчик	<p>Адаптивная фильтрация сигналов</p> <p>Измерение параметров сигналов в присутствии помех</p> <p>Поворот и масштабирование многомерных изображений</p> <p>Выделение параметров рассогласования</p> <p>Адаптивное сжатие данных, компактное представление изображений</p> <p>Распознавание изображений и классификация сигналов</p>
3. Блок обработки и преобразования сигналов	<p>Помехоустойчивый прием, восстановление сигналов, измерение их параметров</p> <p>Масштабные, геометрические и другие преобразования сигналов</p> <p>Накопление данных, статистический экспресс-анализ</p> <p>Прогнозирование сигналов, параметров сигналов и траекторий движения объектов</p> <p>Распознавание образов, классификация сигналов</p> <p>Идентификация процессов и систем</p> <p>Преобразование координат объектов</p> <p>Обработка сигналов комплексированных систем</p> <p>Задание программных функций и сигналов эталонного времени</p> <p>Выделение параметров рассогласования</p> <p>Обеспечение заданных динамических свойств САУ</p> <p>Регистрация параметров управляемых процессов и объектов</p> <p>Сопряжение с другими системами комплекса</p>
4. Блок формирования команд управления	<p>Реализация оптимальных законов управления и регулирования в одно контурных и многоконтурных системах</p> <p>Помехоустойчивое кодирование</p> <p>Преобразование команд в аналоговую форму</p> <p>Передача сигналов управления к исполнительным механизмам</p>
5. Устройство контроля и проверки	<p>Индикация состояний системы</p> <p>Поиск и локализация неисправностей</p> <p>Самовосстановление функций САУ</p> <p>Реализация сервисных функций</p>

Обобщенные функции микроЭВМ в системах автоматического управления. Анализ существующих схем построения и задач, решаемых микроЭВМ в автоматических системах, позволяет выделить ряд основных функций, которые выполняют микроЭВМ. Эти

функции могут быть разделены на группы, определяющие в случае их программной реализации возможную структуру автоматических систем (табл. 1.2). Функциями микроЭВМ, связанными с приемом и предварительной обработкой сигналов датчика, являются линеаризация амплитудной характеристики датчика и масштабирование сигналов [183]. Для реализации этих функций используются системы с управлением от центрального процессора и распределенные системы сбора данных, основанные на периферийных микропроцессорах.

Линеаризация амплитудных характеристик датчиков и масштабирование сигналов обычно сопрягаются с их аналого-цифровым преобразованием. Возникающие в процессе работы АЦП погрешности автоматически корректируются с использованием программных средств микроЭВМ [212].

На микроЭВМ возлагаются также функции сжатия информации методами адаптивной телеметрии и их помехоустойчивое кодирование [64]. Такая обработка информации производится, как правило, в системах автоматического управления с линиями связи, открытыми для проникновения помех и имеющими ограниченную пропускную способность.

Использование в датчиках микроЭВМ позволяет передать им для выполнения некоторые "интеллектуальные" функции, связанные с адаптацией датчиков к меняющимся условиям наблюдения сигналов с целью их помехоустойчивой фильтрации, а также с адаптивным сжатием данных, распознаванием изображений и классификацией сигналов, масштабными и другими преобразованиями сложных многомерных изображений [75, 82, 104, 194].

Эти и многие другие функции выполняют микроЭВМ также в процессе преобразования сигналов и формирования управляющих воздействий. Прием сигналов, поступающих по открытым линиям связи, требует их фильтрации и восстановления, измерения информативных параметров, выявления рассогласований [102, 103, 207]. Важной функцией микроЭВМ, связанной со многими приложениями, является идентификация сигналов, прогнозирование их параметров [28, 45]. При управлении аэрокосмическими объектами микроЭВМ осуществляют преобразование координат, реализацию оптимальных алгоритмов управления в комплексированных системах [164, 191, 205], задание программных функций, прогнозирование траекторий [88, 185].

МикроЭВМ представляют отличные возможности для организации в системах управления самоконтроля и самовосстановления утраченных функций [174]. Снабженные специальными программами микроЭВМ способны выполнять сервисные функции, связанные с решением задач обслуживающего персонала.

Приведенный перечень функций микроЭВМ является далеко не полным. Функции микромашин с развитием науки, техники и технологии непрерывно расширяются и усложняются, что обеспечивает неуклонный рост показателей эффективности работы автоматических систем.

§ 1.3. ПРОБЛЕМЫ РАЗРАБОТКИ И РЕАЛИЗАЦИИ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ С МИКРОЭВМ

Особенности использования микропроцессоров и микроЭВМ в САУ. Внедрение микропроцессоров и микроЭВМ в САУ связано с принципиальными изменениями как структуры, так и характеристик самих САУ. Появляется аппаратура новых видов, которую раньше, до появления микропроцессоров и микроЭВМ, разработать и изготовить было невозможно. Пересматриваются установившиеся взгляды на объем и методы использования вычислительной техники в системах автоматике. Меняются также методы и технические средства проектирования. Все это связано с рядом особенностей микропроцессоров и микроЭВМ на их основе.

Главными их особенностями, как мы уже отмечали, являются малые габариты, масса, энергопотребление и стоимость. Микропроцессоры обладают относительно большой вычислительной мощностью.

Микропроцессоры сделали технически и экономически целесообразным распределение вычислительных средств по всей системе, превращая ее в децентрализованную, функционально более надежную и эффективную.

Появились принципиально новые структуры многопроцессорных систем, предназначенных для решения вычислительных задач в реальном времени. В результате перед разработчиками микропроцессорных систем встал ряд проблем разной степени сложности, которые условно можно объединить в группы, связанные с выбором микропроцессорных средств, обеспечением заданных показателей качества управления, разделением функций САУ между аппаратными и программными средствами, выбором архитектуры микроЭВМ, сопряжением цифровой и аналоговой частей САУ, выбором средств и методов проектирования, обеспечением заданных показателей надежности.

Проблемы перечислены в порядке, определяемом, в основном, структурой алгоритма процесса проектирования. Среди них трудно выделить важнейшие, так как все они неразрывно связаны, и от качества решения каждой из них зависит в целом эффективность применения микроЭВМ в автоматических системах.

Подробнее эти проблемы характеризуются ниже. Путям и методам решения отдельных проблем посвящены последующие главы книги.

Проблема выбора микропроцессорных средств. Современная электронная промышленность предоставляет разработчикам САУ широкий ассортимент микропроцессорных средств. Это различные универсальные и специализированные микроЭВМ, а также разнообразные микропроцессорные комплекты БИС, отличающиеся своими характеристиками, параметрами, функциональными возможностями, способами и языками программирования, принципами организации вычислительных процессов, полнотой и качеством программного обеспечения, доступностью, популярностью и т.д.

Разобраться в обилии микропроцессорных средств непросто. Еще труднее осуществить их правильный выбор. От того, насколько успешно осуществлен выбор микропроцессорных средств, решающим образом зависит эффективность всего проектирования САУ.

Выбор микропроцессорных средств невозможно осуществить в отрыве от задачи управления, ее алгоритмического выражения. Он опирается также на требования технического задания по быстродействию САУ, точности и другим показателям качества управления и основывается на предварительных расчетах, а также знаниях и, в определенной степени, интуиции разработчика.

Разработчик САУ с микроЭВМ должен детально представлять себе особенности тех или иных микропроцессорных средств, владеть основными принципами их выбора и руководствоваться этими принципами на практике.

Принципы выбора микропроцессорных средств относительно просты. Проиллюстрируем некоторые из них на несложном примере. Предположим, что задачей проектирования является разработка цифрового фильтра с заданной передаточной функцией. Фильтр должен иметь высокую верхнюю граничную частоту полосы пропускания (этим требованием мы ставим перед собой задачу выбора наиболее быстродействующих как микропроцессорных средств, так и алгоритма обработки данных). Анализ алгоритма цифровой фильтрации, задаваемого в виде разностного уравнения общего вида

$$y[n] = \sum_{l=0}^q a_l x[n-l] + \sum_{k=1}^m b_k y[n-k], \quad (1.5)$$

показывает, что основными являются операции взвешенного суммирования $x[n-l]$ и $y[n-k]$. Эти операции повторяются в каждом цикле вычисления очередного значения $y[\cdot]$. Для хранения значений $x[n-l]$ и $y[n-k]$ с учетом алгоритма (1.5) целесообразно использовать два блока стековой памяти.

Операцию умножения следует реализовать аппаратными средствами, обеспечивающими наибольшее быстродействие. Тогда становится очевидным, что несложные задачи по управлению вычисли-

тельным процессом могут быть возложены на быстродействующий микропроцессор с микропрограммным принципом управления.

Приведенные нами рассуждения могут быть осознанно восприняты только подготовленным читателем. Для специалиста, впервые знакомящегося с микропроцессорной техникой, выделим три основных принципа, которыми мы руководствовались в рассмотренном примере.

Принцип первый: многократно повторяющиеся операции умножения (деления) для получения наивысшего быстродействия целесообразно реализовать аппаратными средствами.

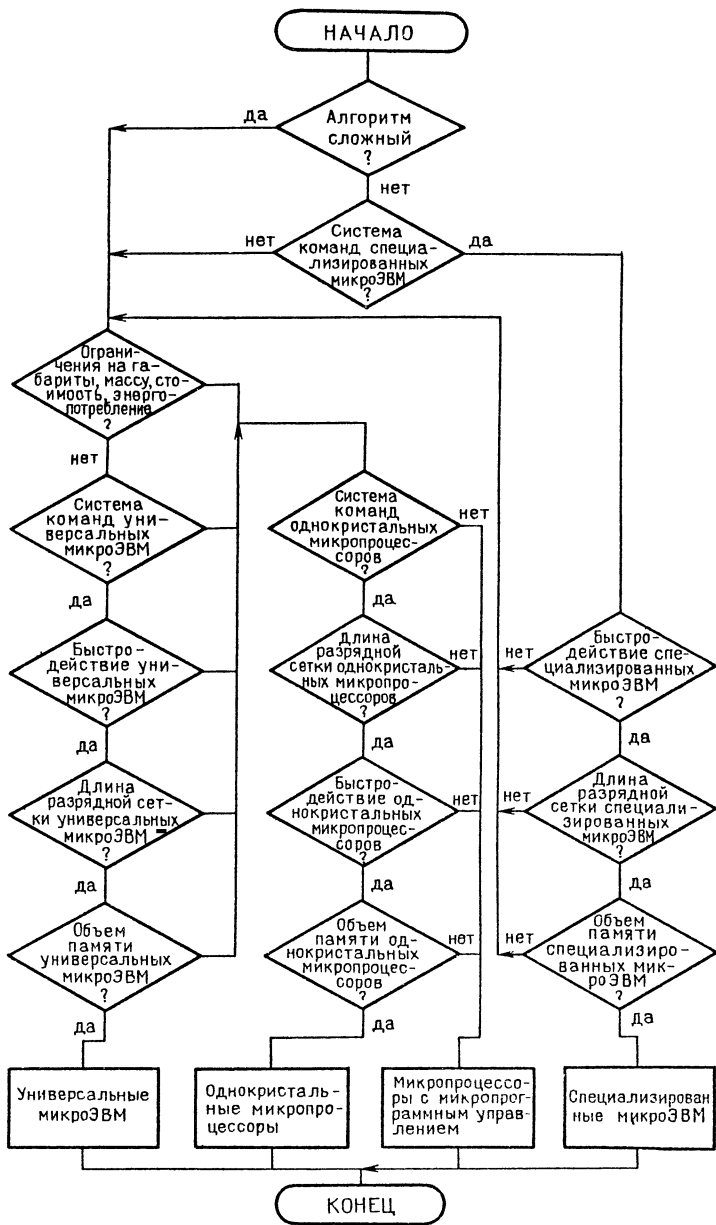
Принцип второй: для хранения и извлечения из памяти упорядоченных массивов данных вида $x[n - i]$, $l = 0, q$, или $y[n - k]$, $k = 1, m$, целесообразной организацией памяти является стековая.

Принцип третий: эффективной реализацией небольшого числа однотипных операций типа сложение, пересылка из регистра в регистр и т.д. является программная реализация на основе комплекта БИС с микропрограммным принципом управления.

Принципы выбора микропроцессорных средств обычно рассматриваются и используются во взаимосвязи. Это обусловлено тем, что многие из них являются взаимно противоречивыми. Например, реализация цифрового фильтра на основе рассмотренных принципов приводит к значительной специализации микроЭВМ, что снижает гибкость ее использования. Для увеличения гибкости микропроцессорных средств следует ориентироваться на применение универсальных микроЭВМ. Но, с другой стороны, универсальные микроЭВМ могут обладать чрезмерной избыточностью аппаратных и программных средств, что делает экономически нецелесообразным использование их, например, в децентрализованных многопроцессорных вычислительных системах и т.д.

Трудность выбора микропроцессорных средств усугубляется также тем, что решение этой задачи можно считать найденным лишь тогда, когда система разработана, испытана и начато ее промышленное производство. Поэтому при выборе микропроцессорных средств на начальном этапе проектирования всегда имеется некоторая доля риска, размер которой может быть уменьшен правильной организацией соответствующей работы.

На рис. 1.21 приведена схема алгоритма выбора микропроцессорных средств, основанного на учете ограниченного числа факторов, таких, как требуемые быстродействие, длина разрядной сетки, объем памяти, степень соответствия системы команд алгоритму задачи, наличие технико-эксплуатационных ограничений на проектируемую аппаратуру. На практике при проектировании САУ с микроЭВМ руководствоваться данным алгоритмом необходимо, но не достаточно. Следует учитывать также ряд других факторов, рассматриваемых далее.



Р и с. 1.21. Схема алгоритма выбора микропроцессорных средств

Обеспечение заданных показателей качества управления. На такие показатели качества управления, как динамика, точность, вероятность выполнения функций управления и др., влияют особенности цифровой реализации алгоритма управления в микроЭВМ. Как известно, при цифровом представлении информации осуществляются дискретизация процессов по времени и квантование по уровню.

Дискретизация процессов по времени, осуществляемая в соответствии с некоторым критерием качества функционирования САУ, предъявляет требование к быстродействию микроЭВМ, а квантование по уровню приводит к возникновению шумов квантования [23].

Рассмотрим этот вопрос несколько подробнее. На практике для уменьшения шумов квантования прибегают к увеличению числа уровней квантования. Однако применение в контуре САУ микроЭВМ налагает определенные ограничения на возможное число уровней квантования, которое нецелесообразно делать большим, чем 2^N , где N — длина разрядной сетки микроЭВМ. Например, для микроЭВМ "Электроника К1-20", имеющей длину разрядной сетки, равную 8 или 16, число уровней квантования может быть соответственно 256 или 65536.

Выбор требуемого числа уровней квантования зависит от особенностей конкретной задачи, решаемой микроЭВМ в САУ. Например, при цифровой фильтрации необходимая разрядность входного преобразователя микроЭВМ определяется в соответствии с выражением [207]

$$N = \max [1 + \log_2 (x_{\max} / x_{\min}) \log_2 (F / 6 \sigma^2 \lg 2)] - 1,$$

где x_{\max} и x_{\min} — соответственно максимальный и минимальный уровни квантования; F — уровень шумов квантования; σ^2 — дисперсия входного сигнала.

Вопросы выбора числа уровней квантования для обеспечения заданных динамических параметров и точности САУ с микроЭВМ рассмотрены в работе [23].

При вычислениях в микроЭВМ возникают также новые составляющие шумов, называемые шумами округления [182]. Это связано с тем, что, например, при умножении 8-разрядных чисел с фиксированной запятой результат получается 16-разрядным. Для последующих расчетов обычно используют только восемь старших разрядов. Отбрасывание восьми младших разрядов уменьшает точность вычислений и может привести к снижению запаса устойчивости и даже к неустойчивости цифровой системы. Поэтому в следящих системах отбрасывают старшие разряды. В этом случае система работает на линейном участке амплитудной характеристики чувствительного элемента. Важным является также следующее обстоятельство. В современных системах автоматического управления со сложной коммутируемой структурой широко используются

различные решающие устройства. Ими могут быть, например, классификаторы, схемы диагностики, статистической экспресс-обработки и др. [47, 104]. Результаты работы этих устройств влияют на структуру автоматических систем, а значит, и на показатели эффективности их функционирования.

Рассмотрим в качестве примера задачу цифровой классификации одномерных квазистационарных нормально распределенных случайных процессов x_i ($i = \overline{1, 2}$) с различными априорно неизвестными средними значениями \bar{x}_1 и \bar{x}_2 , одинаковыми, но также априорно неизвестными дисперсиями σ^2 . Предположим, что $i = 1$ соответствует альтернативе, $i = 2$ — гипотезе. Один из возможных в данном случае алгоритмов классификации содержит операции статистического оценивания \bar{x}_1 , \bar{x}_2 и σ^2 , формирования с использованием этих данных аналоговой величины порогового уровня $x_{\text{п}}$ и сравнения наблюдаемого значения процесса x с величиной порога для принятия решения [139].

При цифровой реализации классификатора диапазон возможных значений порога принятия решений квантуется на N уровней. Численное значение величины порогового уровня устанавливается с погрешностью, достигающей величины шага квантования. Это объясняется тем, что вероятность ложных тревог при аналоговой реализации согласно выбранному в примере критерию Неймана—Пирсона задана и не может быть увеличена при цифровой обработке сигналов.

В связи с погрешностью установки порога принятия решения устройство теряет оптимальность и вероятность правильной классификации снижается. Нетрудно показать, что максимальное изменение вероятности правильной классификации в сторону снижения равно [104]

$$\Delta P_m = \Phi[\Delta \bar{x}_n - \Phi^{-1}(1 - \alpha)] - \Phi[\Delta \bar{x}_n - \Delta x_n - \Phi^{-1}(1 - \alpha)], \quad (1.6)$$

где $\Phi[\cdot]$ — интегральная функция нормальной случайной величины; $\Phi^{-1}[\cdot]$ — обратная интегральная функция; $\Delta \bar{x}_n = (\bar{x}_2 - \bar{x}_1) / \sigma$; Δx_n — нормированный шаг квантования; α — вероятность ложных тревог.

Из анализа выражения (1.6) следует, что шаг квантования диапазона изменения порога принятия решений существенно влияет на величину ΔP_m , характеризующую снижение вероятности правильного обнаружения. Так, например, если $\Delta \bar{x}_n = 6$ (это соответствует при $\alpha = 10^{-4}$ вероятности правильного обнаружения оптимальным классификатором 0,99) и $\Delta x_n = 0,1 \Delta \bar{x}_n$, то $\Delta P_m = 0,04$.

Уравнение (1.6), преобразованное к виду

$$\Delta x_n = \Delta \bar{x}_n - \Phi^{-1}(1 - \alpha) - \Phi^{-1} \{ \Phi[\Delta \bar{x}_n - \Phi^{-1}(1 - \alpha)] - \Delta P_m \},$$

позволяет вычислить необходимое количество уровней квантова-

ния диапазона изменения величины порога принятия решений, которое обеспечивает снижение вероятности ΔP_m на величину, не выше заданной. Для этого на основе априорных сведений о статистических свойствах процессов устанавливается размер диапазона D . На нижней границе диапазона по заданной величине ΔP_m и α вычисляется величина Δx_n .

Приведенные примеры показывают, что решение проблемы обеспечения заданных показателей качества управления связано, в частности, с выбором периода дискретизации и числа уровней квантования входного сигнала, которые зависят как от частотных и энергетических характеристик процесса, так и от алгоритма обработки информации, и ограничены соответствующими характеристиками современных микроЭВМ, такими, как быстродействие и длина разрядной сетки.

Например, однопроцессорные микроЭВМ имеют быстродействие до 1 млн. операций в секунду, а длина разрядной сетки для однокристалльных микропроцессоров не превышает 16 разрядов (имеется возможность выполнения операций с двойной точностью, но при этом снижается быстродействие) и может достигать 64 разрядов у многокристалльных микропроцессоров с микропрограммным принципом управления. Залогом дальнейшего увеличения быстродействия микроЭВМ являются достижения технологии. Уже сегодня наиболее быстродействующие микропроцессоры способны выполнять десятки миллионов операций в секунду.

Однако, ориентируясь на современный уровень технологии производства микропроцессоров и микроЭВМ, разработчику для достижения необходимой производительности микроЭВМ САУ можно рекомендовать использовать распараллеливание вычислительных задач между многими микропроцессорами, выполняемое различными методами, в том числе и описанными в § 1.4.

Рациональное распределение функций САУ между аппаратными и программными средствами. Применение микропроцессоров и микроЭВМ в САУ основано на использовании различных концепций построения микропроцессорных средств.

Так, характерные особенности микропроцессоров и микроЭВМ, о которых упоминалось выше, привели к широкому развитию децентрализованных автоматических систем. Многие функции центральных управляющих ЭВМ переданы периферийным микроЭВМ, на базе которых созданы и разрабатываются "интеллектуальные" датчики. В результате резко повысилась живучесть систем.

В системах автоматического управления все большее применение находят адаптивные микроконтроллеры, используемые в тех случаях, когда в состав систем входят нелинейные элементы, элементы с изменяющимися во времени параметрами или с заранее неизвестной моделью, элементы с большими постоянными времени и т.д.

Особая роль в системах автоматического управления отводится системам с нетрадиционной, так называемой перестраиваемой структурой. В этих системах общее управление осуществляется над множеством параллельно работающих микропроцессоров. Структура такой системы может изменяться в процессе решения задачи, адаптируясь под ее особенности. В результате производительность таких систем достигает сотен миллионов операций в секунду [174].

Таким образом, микропроцессорные средства предоставляют разработчикам аппаратуры хорошие возможности для реализации различных алгоритмов управления с использованием программируемых структур. В итоге на практике проявляется важное достоинство САУ с микроЭВМ, связанное с их гибкостью, т.е. возможность модификации функций программным путем. С этих позиций передача в наибольшем объеме функций элементов САУ для использования микроЭВМ является оправданной. Но в этом случае может быть не обеспечена заданная производительность микроЭВМ. Поэтому в микроЭВМ наряду с программной широко используется аппаратная реализация алгоритмов обработки информации.

Задача распределения функций САУ на аппаратно и программно реализуемые не является тривиальной. Ее решение связано с необходимостью анализа многих факторов, первостепенными из которых, наряду с обеспечением заданной производительности микроЭВМ, являются структура системы, характер представления входной и выходной информации (аналоговая, бинарная, цифровая) и особенности цифровых алгоритмов обработки информации.

Разработчикам САУ с микроЭВМ можно рекомендовать во всех случаях, когда выполнение функций САУ обеспечивается программной реализацией алгоритмов, использовать этот путь создания автоматических систем.

Выбор архитектуры управляющей микроЭВМ. Проблема выбора архитектуры микроЭВМ возникает не только в том случае, когда в ходе проектирования САУ разрабатывается специализированная управляющая микроЭВМ. Эта проблема возникает уже на начальном этапе, связанном с выбором микропроцессорных средств.

Не останавливаясь детально на анализе понятия "архитектура" управляющих микроЭВМ (это сделано в § 1.4), отметим, что понятие включает, в частности, аппаратную организацию (структуру) микроЭВМ, ее систему команд, аппаратные средства, принципы организации взаимодействия с внешними устройствами и др.

При выборе архитектуры управляющей микроЭВМ разработчиком САУ решается ряд задач, в совокупности обеспечивающих выполнение САУ возложенных на нее функций.

Первая задача связана с многовариантностью структур управляющих микроЭВМ. Микропроцессоры и микроЭВМ предоставляют возможность применения многих вариантов структур вычислительных средств. В результате перед разработчиком всегда возникает вопрос, как распределить вычислительную мощность системы по ее элементам и блокам. Принципиально имеются два полярных варианта ответа на этот вопрос: все вычисления делаются только центральной управляющей микроЭВМ; все локальные микроЭВМ специализированы на решении какой-то местной специфической задачи, и дополнительно имеется только диспетчер, осуществляющий функции координатора всей работы. Очевидно, что существует чрезвычайно большое число промежуточных вариантов, и чем сложнее автоматическая система, тем большее число вариантов может считаться логичным. Многообразными являются также внутренние структуры микроЭВМ, а также способы организации и состав аппаратных средств сопряжения микроЭВМ с внешними устройствами. Для примера упомянем синхронный и асинхронный способы организации обмена данными, режим прямого доступа к памяти, организуемые с использованием программируемых и непрограммируемых БИС интерфейса и контроллера прямого доступа к памяти.

Внутренняя структура микроЭВМ связана прежде всего с методами организации системной шины и памяти (в том числе стековой). Известны системы с одной, двумя и тремя шинами, память может быть "блокнотного" типа, матричная и т.д.

Вторая задача связана с многообразием систем команд. Системы команд микроЭВМ определяют в значительной степени их вычислительные мощности и степень универсальности применения. Так, например, если требуется разработать свою собственную систему команд, то в этом случае надо обратиться к микроЭВМ с наращиваемым списком команд (микроЭВМ семейств "Электроника НЦ", "Электроника С5"), либо разработать специализированную управляющую микроЭВМ на базе комплекта БИС с микропрограммным принципом управления.

Для большинства задач управления достаточно эффективными оказываются системы команд микроЭВМ семейств "Электроника К1" и "Электроника 60". Списки команд этих машин нельзя изменить. Но команды в этих списках составлены настолько удачно, что вот уже много лет являются образцовыми и воспроизводятся в новых поколениях микропроцессоров и микроЭВМ.

Таким образом, выбор архитектуры управляющей микроЭВМ является важной проблемой, решение которой связано с глубоким пониманием общих принципов построения управляющих микроЭВМ и возможностей их использования на практике.

Сопряжение цифровой и аналоговой частей САУ. Для сопряжения аналоговой части САУ с микроЭВМ используются аналого-цифровые и цифро-аналоговые преобразователи. Наибольшие трудности обычно составляют разработка и подключение к микроЭВМ АЦП, к которым предъявляются высокие требования по быстродействию, точности и линейности преобразования.

САУ с микроЭВМ являются системами, работающими в режиме реального времени. Это предъявляет повышенные требования к частоте дискретизации аналогового сигнала, которая может достигать нескольких десятков мегагерц. В течение времени, равного периоду дискретизации сигнала, АЦП должен сформировать на своем выходе цифровой эквивалент входного сигнала. При этом должна быть обеспечена разрешающая способность в 6–12 и более разрядов. Создание таких преобразователей представляет собой крупную техническую и технологическую проблему.

Большинство существующих типов АЦП имеют разрешение до 6 разрядов. Поэтому важным представляется вопрос организации АЦП в 12 разрядов и более на основе преобразователей с меньшим разрешением.

Наконец, АЦП подключается к микроЭВМ, имеющей зачастую длину разрядной сетки, превосходящую число разрядов АЦП. В связи с этим возникает необходимость обеспечения схемотехнического и программного сопряжения АЦП меньшей разрядности с шиной данных микроЭВМ.

Эти и некоторые другие задачи являются составным элементом в том комплексе вопросов, которые возникают в связи с использованием цифровых методов обработки информации в аналоговых по своей природе автоматических системах.

Схемотехническое проектирование, разработка программного обеспечения и отладка САУ с микроЭВМ. Особенностью микропроцессорных средств как элементной базы современной аппаратуры систем автоматического управления является то, что схемотехническое проектирование и разработка программного обеспечения производятся практически параллельно. При этом стоимость и трудоемкость разработки программного обеспечения в 3–10 раз превосходят соответствующие затраты на схемотехническое проектирование. Поэтому при разработке САУ с микроЭВМ широко используются резидентные системы автоматизированного проектирования (САПР), позволяющие не только разработать и отладить программное обеспечение и схемотехнические компоненты управляющей микроЭВМ, но и проверить работоспособность всей САУ в целом на основе полунатурного эксперимента.

Резидентные САПР имеют, как правило, развитое математическое обеспечение, необходимые приборный состав и периферийное

оборудование. В результате резко сокращаются сроки проектирования, повышается достоверность результатов проектирования.

Практическое использование САПР теми специалистами, которые знакомы со схемотехническими особенностями микропроцессорных систем и программированием на языке ассемблера, не встречает особых затруднений. Проблема здесь заключается в необходимости широкого распространения резидентных и в еще большей степени комбинированных САПР. Вот что об этом говорит академик АН ГССР И.В. Прангишвили: "... необходимо развернуть широкие исследовательские и опытно-конструкторские работы по созданию и освоению как относительно дешевых, но специализированных (на некоторый набор БИС), так и более дорогих, но и более универсальных (на широкую номенклатуру микропроцессорных БИС) систем проектирования, макетирования и отладки микропроцессорных систем. . ." [176].

Новые акценты в проблеме надежности микропроцессорных систем. Надежность микропроцессорных систем по сравнению с системами на другой элементной базе (БИС малой и средней степени интеграции) является высокой. По данным, опубликованным в работе [54], надежность микропроцессорных систем в 5 – 10 раз выше, чем аппаратуры, выполненной на БИС. Одна из причин повышения надежности состоит в резком уменьшении числа внешних "паяных" и разъемных соединений между элементами. Типовая БИС с 12 – 20 отводами требует до 30 внешних соединений. Поскольку, например, микропроцессор заменяет более 50 таких БИС и имеет около 50 внешних выводов, то очевидно, что применение микропроцессора исключает примерно 1500 внешних соединений. Так как вероятность "катастрофических" отказов в микропроцессорных системах становится незначительной, то на первое место в отношении повышения общей технологической надежности выходят методы борьбы с деградационными сбоями и отказами.

§ 1.4. ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ МИКРОЭВМ В САУ

Архитектура и производительность микроЭВМ в САУ. Системы автоматического управления решают возложенные на них задачи в темпе, совпадающем со скоростью протекания управляемых процессов.

В течение отрезка времени, равного длительности машинного цикла, микроЭВМ САУ выполняет значительную работу, связанную с задачами фильтрации сигналов, их спектрального или корреляционного анализа, идентификации, преобразования координат объекта управления, формирования управляющих воздействий и т.п. Для решения этих и ряда других задач микроЭВМ систем ав-

томатического управления должны обладать значительной вычислительной мощностью.

Характеристикой вычислительной мощности является производительность, учитывающая как быстродействие процессора микроЭВМ, так и метод организации вычислительных средств. Производительность вычислительной системы оценивается отношением [6]

$$\pi = S/\tau_k, \quad (1.7)$$

где S — коэффициент производительности вычислительной системы, учитывающий влияние ее архитектуры; τ_k — среднее время выполнения короткой операции в процессоре.

Архитектура вычислительной системы определяет ее основные функциональные возможности. Она зависит в свою очередь от особенностей решаемой задачи.

В настоящее время сформировался относительно общий подход к анализу понятия "архитектура" вычислительных систем [6].

Говоря об архитектуре вычислительных систем цифровых САУ, следует отметить, что такие вычислительные системы являются специализированными. Каждому алгоритму может соответствовать своя архитектура вычислительной системы, в которой оптимально с точки зрения производительности объединены аппаратные и программные средства.

Содержание понятия "архитектура" вычислительных систем цифровых САУ иллюстрируется рис. 1.22. В состав этого понятия входят аппаратные и программные средства.

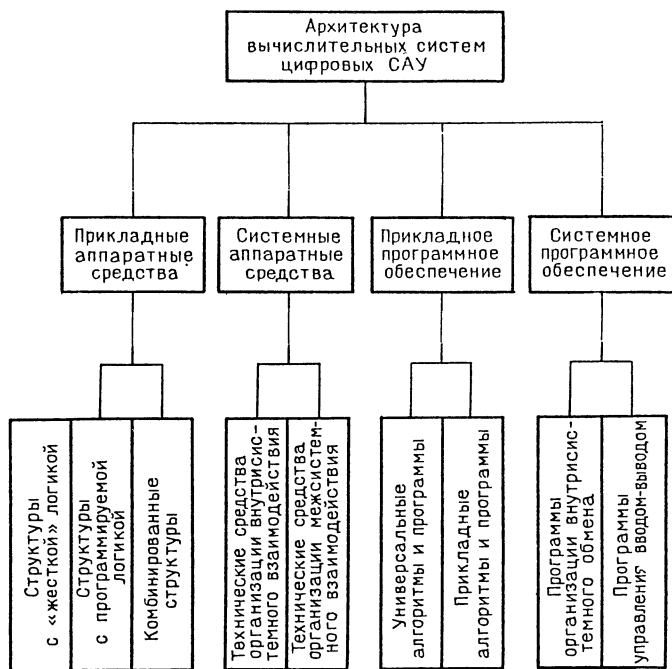
Аппаратные средства представлены двумя большими группами (прикладные аппаратные средства и системные аппаратные средства), каждая из которых оказывает влияние на производительность вычислительных систем.

Прикладные аппаратные средства представлены структурами с непрограммируемой, так называемой "жесткой", программируемой и комбинированной логикой. Структуры с программируемой логикой обеспечивают в среднем меньшую производительность вычислительных систем, чем структуры с "жесткой логикой" и комбинированные структуры.

Системные аппаратные средства включают технические средства организации внутри- и межсистемного взаимодействия.

Программные средства включают прикладное программное обеспечение и системное программное обеспечение.

Прикладное программное обеспечение оказывает влияние на производительность вычислительных систем в силу многообразия программных алгоритмов, которые могут быть использованы для решения одной и той же задачи. Так, например, преобразование Фурье может быть выполнено на основе дискретного или одного



Р и с. 1.22. Архитектура вычислительных систем цифровых САУ

из быстрых преобразований. Диапазон изменения производительности вычислительной системы в этом случае может изменяться в десятки и сотни раз [13].

Системное программное обеспечение предназначено для эффективной эксплуатации аппаратных и программных средств системы. К числу важнейших возможностей, предоставляемых программным обеспечением, относится управление процессом организации вычислений, которое реализуется операционной системой.

Правильный выбор операционной системы является во многих случаях важнейшим условием, обеспечивающим решение задачи управления в реальном масштабе времени [78].

Операционные системы, состоящие из программных модулей, управляют ресурсами технических средств, разрешают конфликтные ситуации, обеспечивая режим приоритетного обслуживания, оптимизируют производительность вычислительной системы цифровой САУ в целом. Операционные системы как распорядители ресурсов управляют процессорами, памятью, устройствами ввода-вывода. Они следят также за использованием информационных ресурсов, обеспечивают требования защиты информации и т.д.

Реализация многообразных функций операционных систем по управлению вычислительными системами обеспечивается на основе использования аппаратных средств организации памяти и взаимодействия с внешними устройствами.

Так, например, аппаратные средства системы прерываний позволяют операционной системе обслуживать одновременно большое количество внешних устройств разными модулями системы.

Итак, влияние архитектуры вычислительных систем на их производительность учитывается в соотношении (1.7) коэффициентом S . Для однопроцессорных вычислительных систем $S = 1$. Увеличение производительности этих систем может быть обеспечено совершенствованием структуры основных арифметико-логических блоков, совершенствованием принципов управления и организации обмена информацией с памятью и внешними устройствами, повышением быстродействия элементной базы. Предполагается, что производительность однопроцессорных вычислительных систем достигнет к 1990 г. 230–250 млн. операций в секунду.

Величина коэффициента производительности вычислительных систем для многопроцессорных базовых структур с программируемой логикой зависит от числа используемых процессоров M , коэффициента системных затрат $k(M)$ и определяется по формуле

$$S_n = M - k(M).$$

Коэффициент $k(M)$ характеризует ту часть вычислительной мощности системы, которая расходуется на выполнение функций управления вычислительным процессом. Ожидается, что к 1990 г. максимальная производительность многопроцессорных вычислительных систем достигнет 500–700 млн. операций в секунду [6].

Перспективным направлением повышения производительности вычислительных систем САУ с микроЭВМ является применение комбинированных аппаратных средств реализации алгоритмов.

В таких вычислительных системах отдельные наиболее часто используемые процедуры реализуются в виде структур с "жесткой" логикой. Примерами таких структур являются спецпроцессор БПФ, спецпроцессор двоичного умножения чисел с плавающей запятой, цифровые фильтры и др. [186, 194].

Производительность вычислительных систем, основанных на использовании комбинированных аппаратных средств реализации алгоритмов, зависит от величины коэффициента

$$S_k = (1 - k'(M)) / (1 - \gamma(1 - k_y)),$$

где $\gamma = T_{ж}/T_p$ — безразмерный коэффициент, учитывающий долю (в единицах времени) задачи, решение которой передано спецпроцессору, в общем времени решения всей задачи вычислительной системой без спецпроцессора; $T_{ж}$ — время решения вычислитель-

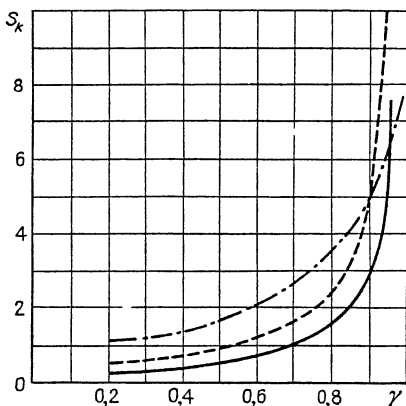


Рис. 1.23. Зависимость коэффициента производительности вычислительной системы от коэффициента γ

ной системой без спецпроцессора той части всей задачи, которая будет передана спецпроцессору; T_p — время решения всей задачи вычислительной системой без спецпроцессора; $k_y = T_{pж}/T_{ж}$ — безразмерный коэффициент, определяемый отношением времени решения $T_{pж}$ части задачи спецпроцессором к

времени решения этой же задачи вычислительной системой без спецпроцессора; $k'(M)$ — коэффициент системных затрат.

Вычислительную систему со спецпроцессором и программируемым процессором можно приближенно считать эквивалентной в смысле системных затрат системе с двумя программируемыми процессорами, так как $k'(2) \approx k(2) = 0,2 - 0,4$.

На рис. 1.23 показан график зависимости коэффициента производительности такой двухпроцессорной вычислительной системы от коэффициента γ . Коэффициент k_y для зависимостей, показанных сплошной ($k'_M = 0,4$) и пунктирной ($k'_M = 0$) линиями, выбран переменным и таким, чтобы для каждого γ значение

$$k_y = (1 - \gamma)/\gamma. \quad (1.8)$$

Этим условием обеспечивается выравнивание времени решения "своей" части задачи спецпроцессором и оставшейся части задачи программируемым процессором.

Анализ приведенных зависимостей показывает, что для выбранных значений коэффициентов k_y, k_M введение в структуру вычислительной системы спецпроцессора повышает производительность вычислительной системы только при значениях коэффициента γ , превышающих 0,5.

Производительность вычислительной системы резко возрастает, если значение коэффициента γ приближается к единице и выполняется условие (1.8). Зависимость, показанная штрихпунктирной линией, получена при $k'_M = 0$ и $k_y = 0,1$.

Аппаратные средства реализации прикладных алгоритмов. В системах автоматического управления значительное число алгоритмов обработки информации может быть реализовано в структурах

с "жесткой" логикой или в спецпроцессорах. Рассмотрим некоторые из них.

Спецпроцессор, выполняющий операцию умножения, может быть реализован по следующему принципу [229].

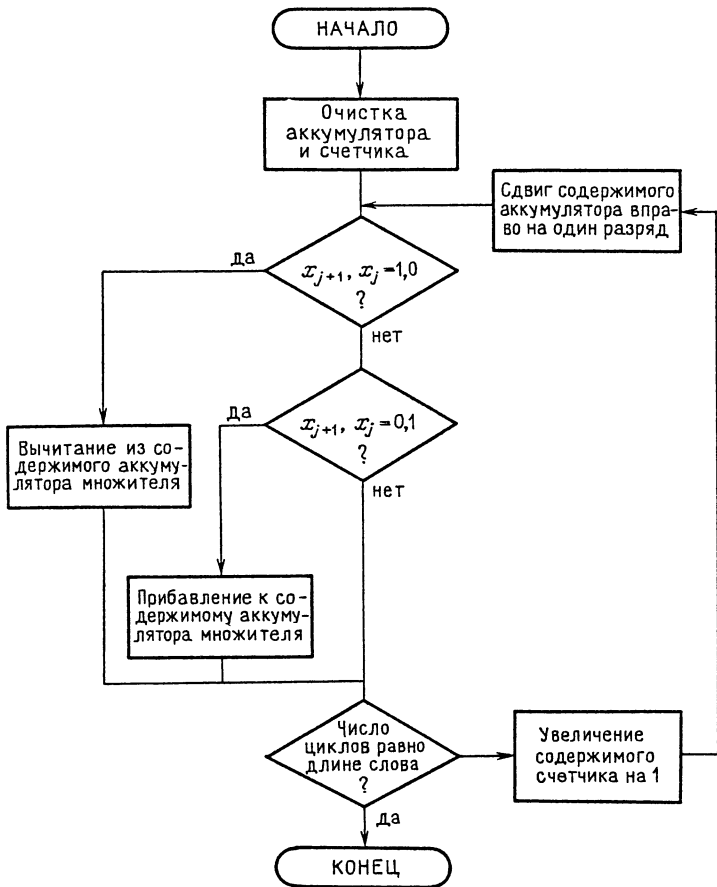
Как известно, увеличение скорости умножения может быть достигнуто заменой обычно имеющихся тактов сложения и сдвига тактом сдвига в том случае, когда в соответствующем разряде множителя записан 0, а также умножением под контролем двух разрядов множителя. При этом умножение происходит по правилам: 00 – нет суммирования; 01 – прибавление множимого; 10 – двукратное прибавление множимого; 11 – трехкратное прибавление множимого. После каждого такта множитель смещается на два разряда.

Одним из вариантов этого метода является суммирование или вычитание под контролем двух разрядов множителя со сдвигом множителя на один разряд. Этот метод нашел применение в микропроцессоре AM2505. Структурная схема алгоритма умножения показана на рис. 1.24, где x_{j+1}, x_j – два разряда множителя. Для умножения по данному алгоритму двух шестнадцатиразрядных чисел требуется максимум 16 тактов сдвига и 16 тактов суммирования. Если предположить, что такты сдвига и суммирования имеют одинаковый период, равный 100 нс, то весь цикл умножения равен 3,2 мкс.

Другим методом умножения является квазипоследовательный метод. Сущность метода заключается в следующем. При умножении двух n -разрядных чисел X и Y формируются промежуточные произведения, представленные ниже в форме скошенного массива:

$$\begin{array}{cccccc}
 y_n & \dots & y_3 & y_2 & y_1 & y_0 \\
 x_n & \dots & x_3 & x_2 & x_1 & x_0 \\
 \hline
 x_0 y_n & \dots & x_0 y_3 & x_0 y_2 & x_0 y_1 & x_0 y_0 \\
 x_1 y_n & \dots & x_1 y_2 & x_1 y_1 & x_1 y_0 & \\
 x_2 y_n & \dots & x_2 y_1 & x_2 y_0 & & \\
 \\
 x_n y_n & & & & & x_n y_0 \\
 \hline
 S_{2n-1} \dots S_i & & \dots & S_3 & S_2 & S_1 & S_0
 \end{array}$$

Окончательный результат получают одним из трех способов: 1) полностью параллельное сложение n -разрядных чисел в определенном порядке; 2) сложение n -разрядных чисел по два разряда одновременно; 3) сложение "в столбик". Первый способ – самый быстроедействующий, самый медленный способ – третий.



Р и с. 1.24. Алгоритм ускоренного умножения двоичных чисел

Параллельное сложение n -разрядных чисел выполняется в квазипоследовательном множителе. Структурная схема 16-разрядного умножителя показана на рис. 1.25. Операция умножения начинается с выравнивания множимого по левому (старшему) разряду в регистре множимого, который является сдвигающим, и засылки множителя в регистр множителя.

В начале счета, если значения младших (нулевых) разрядов обоих чисел равны 1, схема совпадения I_0 вырабатывает сигнал, поступающий на вход комбинационного счетчика. Сигнал с выхода комбинационного счетчика одновременно поступает в сумматор и в младший разряд регистра сдвига. Далее множимое сдвигается на один разряд вправо и процесс повторяется.

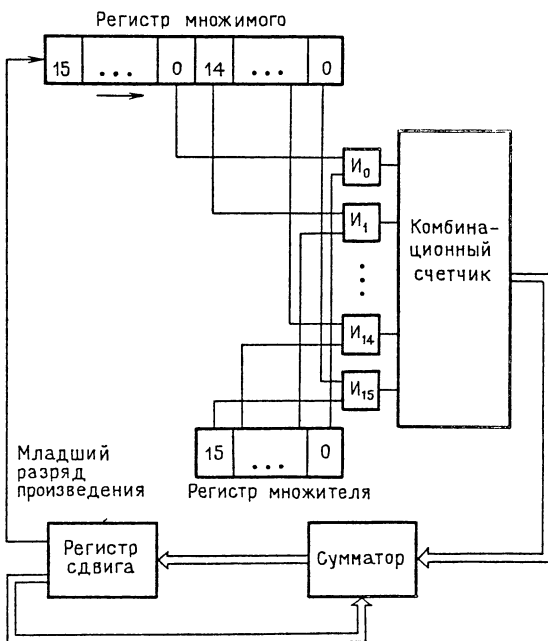
В течение 15 тактов множимое последовательно сдвигается на один разряд вправо и сравнивается со старшими разрядами множителя. По мере выполнения операции умножения младшие разряды произведения засылаются в регистр множимого.

При частоте тактовых импульсов, равной 10 МГц, время выполнения операции умножения 16-разрядных чисел составляет 3,2 мкс. Это время сравнимо со временем выполнения операции сложения 16-разрядных чисел, хранящихся в регистре общего назначения и в памяти, большинством микропроцессоров на МОП-структурах.

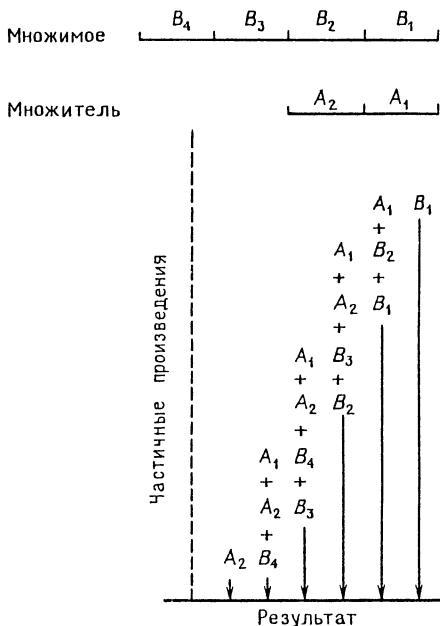
Дальнейшее сокращение времени умножения двоичных чисел достигается использованием техники табличного поиска, циклической матрицы, пирамиды Wallace и др.

В устройствах умножения использование табличного поиска в чистом виде затруднено из-за необходимости иметь большой объем памяти. Компромиссным решением является сочетание матриц постоянной памяти с суммирующими блоками. Пример умножения 16-разрядного числа на 8-разрядное показан на рис. 1.26. Умножение производится следующим образом.

Множимое B и множитель A разбиты на группы по 4 разряда каждая. Программа умножения осуществляет поиск частичных



Р и с. 1.25. Схема квазипоследовательного умножителя



Р и с. 1.26. Умножение с использованием таблицы частичных произведений и сумматоров

произведений этих групп в матрице постоянной памяти. Для хранения частичных произведений 4-разрядных чисел требуется постоянная память с объемом в 256 8-разрядных чисел. Выбранные из постоянной памяти частичные произведения складываются в 4-разрядных двухвходовых сумматорах по правилам, показанным на рис. 1.26.

Быстродействие этой схемы определяется временем выборки из постоянной памяти, быстродействием сумматоров и временем прохождения сигнала переноса.

Время выполнения операции умножения T_y 16-разрядных чисел для некоторых методов и относительная стоимость оборудования показаны в табл. 1.3.

Примером аппаратной реализации одного из алгоритмов обработки информации, обеспечивающих повышение помехоустойчивости последовательного приема и передачи кодов в бортовую локальную сеть, является интегральная схема интерфейса, в составе которой имеется блок кодирования по манчестерскому методу [79]. БИС интерфейса подключается к системной шине бортовой ЭВМ и обеспечивает прямой доступ к памяти в режиме прерывания. Интерфейс позволяет обнаруживать ошибки, выявлять случаи, когда сигналы ответа от дистанционно расположенных ведомых устройств не поступают. Это приводит к разгрузке бортовой ЭВМ и повышению ее производительности.

Вычислительные структуры с программируемой логикой. Однопроцессорные и многопроцессорные вычислительные системы образуют комплекс специализированных вычислительных средств, последовательность обработки информации в которых задается рабочей программой, хранящейся в постоянной памяти.

Повышение производительности вычислительных систем с программируемой логикой достигается обычно применением несколь-

ких процессоров с параллельным выполнением вычислительной задачи [6, 100]. Такие системы содержат, как правило, несколько процессоров, основную память общего пользования, собственную память процессоров, каналы общего доступа к устройствам ввода-вывода, единую интегрированную операционную систему.

Структуры многопроцессорных вычислительных систем зависят в значительной степени от особенностей алгоритмов управления, которые должны быть реализованы в этих структурах, и потоков обрабатываемых данных.

Так, например, если поток данных допускает использование одного потока команд, то в этом случае возможно применение матричных, параллельных и ассоциативных структур вычислительных систем. Множественный поток команд и множественный поток данных требуют применения классической структуры многопроцессорной вычислительной системы, содержащей для каждого потока данных свой процессор и свое устройство управления.

Основываясь на классификации, приведенной в [6], проанализируем возможности применения многопроцессорных вычислительных систем для задач управления в реальном времени. Для этого воспользуемся данными табл. 1.4.

В вычислительных системах с шинной организацией для передачи данных, команд управления и адресов используются соответствующие шины. Процесс обмена информацией осуществляется последовательно во времени. Это приводит к тому, что из-за возникновения очередей при передаче пакетов данных производительность таких вычислительных систем оказывается в сравнении с многопроцессорными вычислительными системами других типов относительно низкой. Кроме того, выход из строя любой из указанных трех шин приводит к нарушению работоспособности всей системы в целом.

Однако многопроцессорные вычислительные системы с шинной организацией обладают важным достоинством — высокой гиб-

Т а б л и ц а 1.3

Метод умножения	Относительная стоимость	T_y , мкс
Сдвиг и суммирование	1	3,2
Квазипоследовательный	1	3,2
Табличный	8	0,8
Циклическая матрица	21	0,15
Пирамида Wallace	37	0,12
Однокристалльный спецпроцессор	10	0,3

Таблица 1.4

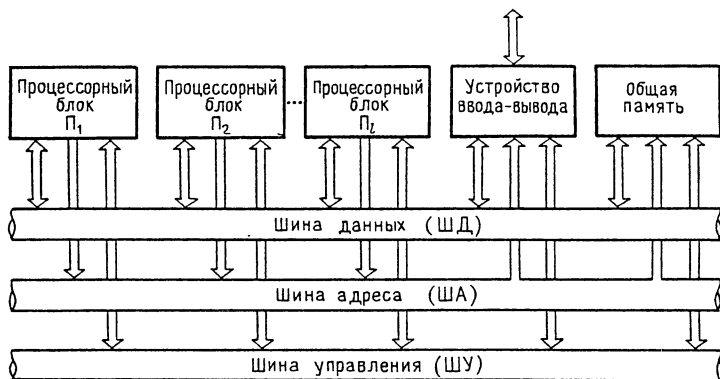
Наименование структур вычислительных систем	Основные характеристики			
	Способ разделения вычислительных операций	Производительность	Надежность	Гибкость
1. Структуры с общей шиной	временной	низкая	низкая	высокая
2. Структуры с перекрестной коммутацией	пространственный	относительно высокая	высокая	высокая
3. Структуры с многовходовой памятью	пространственный	высокая	высокая	высокая
4. Матричные и векторные структуры	пространственно-временной	очень высокая	средняя	относительно низкая
5. Ассоциативные системы	пространственно-временной	очень высокая	средняя	низкая
6. Структуры конвейерного типа	пространственно-временной	высокая	низкая	низкая

костью. Они эффективно могут быть приспособлены для решения новых и модернизированных задач системы управления [210].

В структурах с перекрестной коммутацией производительность может быть несколько выше, так как обмен информацией возможен одновременно по нескольким путям передачи данных. Однако структура усложняется вследствие введения коммутатора. Гибкость таких вычислительных систем также является высокой.

Многопроцессорные структуры с многовходовой памятью используют несколько путей одновременной передачи информации. В результате производительность системы может быть существенно выше производительности однопроцессорной системы. Структуры с многовходовой памятью позволяют одновременно решать большое число задач, образующих алгоритм обработки информации при управлении объектом или процессом. Отказы отдельных элементов не приводят к отказу системы, так как элементы могут логически заменяться действующими в процессе работы. Топология схемы с многовходовой памятью более экономична, чем конфигурация с перекрестной коммутацией.

В указанных многопроцессорных системах параллельное выполнение операций достигается увеличением числа процессоров, каж-



Р и с. 1.27. Схема вычислительной системы конвейерного типа

переменная; t — независимая переменная; $f_i, i = \overline{1, l}$, — заданная нелинейная (в частности — линейная) функция.

Наиболее распространенными являются три метода [201]. Согласно первому каждый процессор решает одно уравнение системы (1.9). Общее количество процессоров равно порядку системы дифференциальных уравнений.

На рис. 1.27 показана структурная схема вычислительной системы конвейерного типа, предназначенной для численного решения системы дифференциальных уравнений по первому методу. В состав системы входят l процессорных блоков, устройство ввода-вывода и общая память. Каждый процессорный блок содержит процессор и память, в которой хранится соответствующая рабочая программа.

Общая память используется для хранения вычисленных значений $y_i[\cdot]$. Ввод независимой переменной $x[\cdot]$ и вывод результатов счета производятся через устройство ввода-вывода.

Второй метод численного решения системы дифференциальных уравнений (1.9) предполагает разделение функций вычисления правых частей уравнений системы и интегрирования между двумя процессорами. Общее число процессоров возрастает в этом случае в два раза.

Структурная схема вычислительной системы, реализующей второй метод, совпадает со схемой, показанной на рис. 1.27. Отличия заключаются в том, что вместо процессорных блоков $\Pi_1, \Pi_2, \dots, \Pi_l$ используются конвейерные пары процессорных блоков (рис. 1.28), содержащие по два процессорных блока и модуль памяти.

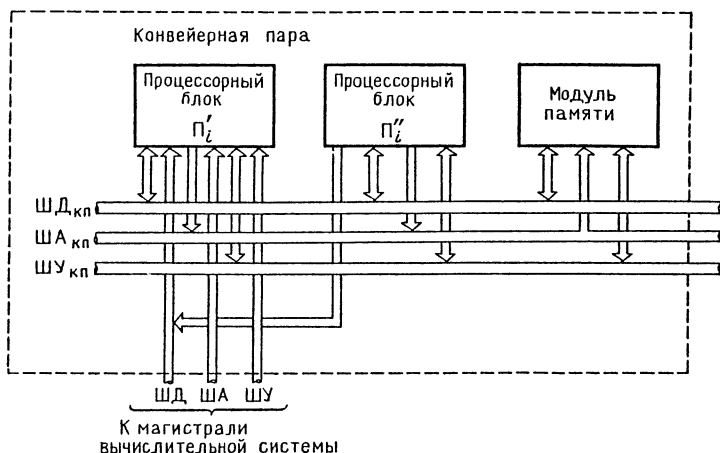
В каждой конвейерной паре вычислительной системы один процессорный блок вычисляет правую часть соответствующего уравнения системы (1.9), другой решает задачу численного интегрирова-

ния. Обмен результатами счета внутри конвейерной пары происходит с использованием модуля памяти, а между конвейерными парами – через общую память.

Третий метод решения системы уравнений (1.9) требует для вычисления каждой из правых частей этих уравнений нескольких процессоров. Структурная схема вычислительной системы включает в этом случае вместо процессорных блоков, показанных на рис. 1.27, конвейерную цепочку процессорных блоков, передача данных между которыми производится через модуль памяти. Конвейерная цепочка процессорных блоков организуется по схеме, показанной на рис. 1.28, где вместо двух должно быть изображено столько процессорных блоков, сколько их имеется в конвейерной цепочке.

Охарактеризуем производительность вычислительных систем, реализующих указанные три метода, в сравнении с однопроцессорной структурой. Для этого первоначально введем параметр, функционально связанный с производительностью и равный длительности интервала интегрирования. Этот параметр обычно называют шагом интегрирования.

Шаг интегрирования для однопроцессорной вычислительной системы зависит от принятого метода интегрирования дифференциальных уравнений и особенностей конкретной программной реализации алгоритма. Допустим, что процессор последовательно решает каждое уравнение системы (1.9), обращая всякий раз к памяти для чтения значений $y_i [\cdot]$, полученных в предыдущем цикле счета. Будем также полагать, что значение $x [\cdot]$ введено



Р и с. 1.28. Схема конвейерной пары процессорных блоков

в память вычислительной системы. В этом случае длительность цикла интегрирования оказывается не менее величины

$$t_0 = l [(l + 1)t_{чт} + t_{в} + t_{и} + t_{зп}], \quad (1.10)$$

где l — порядок системы дифференциальных уравнений; $t_{чт}$, $t_{и}$, $t_{зп}$, $t_{в}$ — соответственно длительности циклов чтения данных из памяти, интегрирования каждого уравнения системы, записи данных в общую память и среднего времени вычисления правых частей уравнений системы. Для конкретности, но без потери общего принципиального характера результатов, допустим, что вычислительная система реализуется на микропроцессоре серии КР580. В этом случае выражение (1.10) можно записать в виде

$$t_0 = T_{м}l [6(l + 2) + N_{в} + N_{и}], \quad (1.11)$$

где $T_{м}$ — длительность машинного цикла; $N_{и}$ и $N_{в}$ — соответственно число машинных циклов и среднее число машинных циклов, необходимых для реализации операций интегрирования и вычисления функций в правой части уравнений системы (1.9).

Шаг интегрирования вычислительной системой конвейерного типа, реализующей первый метод, определяется из уравнения [24]

$$t_1 = lt_{чт} + t_{п} + t_{в} + t_{и} + lt_{зп}, \quad (1.12)$$

где $t_{п}$ — длительность цикла параллельной пересылки данных из внешнего устройства-источника в процессоры и результата из процессора во внешнее устройство-приемник. В соотношении (1.12) величина $t_{в}$ принимается максимальной из возможных.

С учетом микропроцессорной реализации вычислительной системы величина

$$t_1 = T_{м} [6(2l + 1) + N_{в} + N_{и}]. \quad (1.13)$$

Второму методу соответствует одна из следующих формул для шага интегрирования:

$$t_2 = lt_{чт} + 0,5t_{п} + t_{в} + t_{зк}, \quad (1.14)$$

$$t_2' = t_{чк} + 0,5t_{п} + t_{и} + lt_{зп}, \quad (1.15)$$

где $t_{зк}$ — длительность цикла записи данных в память конвейерной пары микропроцессоров; $t_{чк}$ — длительность цикла чтения данных из памяти конвейерной пары.

Выражение (1.14) применяется, если $t_2 > t_2'$. Если $t_2' > t_2$, то используется соотношение (1.15).

С учетом сделанного предположения о реализации вычислительной системы на микропроцессоре КР580ИК80 выражения (1.14)

и (1.15) переписываются в виде

$$t_2 = T_M [6l + 3 + N_B], \quad (1.16)$$

$$t'_2 = T_M [6l + 3 + N_{II}]. \quad (1.17)$$

Третий метод целесообразно использовать тогда, когда длительность цикла интегрирования существенно меньше длительности цикла вычисления правой части каждого из уравнений системы. В этом случае шаг интегрирования

$$t_3 = dt_{\text{чТ}} + t_{\text{п}} + kt_{\text{в}} + mt_{\text{ЭК}}, \quad (1.18)$$

где d — число переменных того промежуточного выражения в правой части уравнения системы, вычисление значения которого требует максимального количества машинных циклов; m — количество вычисленных значений, направляемых в память конвейерной пары; $k = (0 - 1)$ — коэффициент, учитывающий долю затрат машинного времени на вычисление промежуточного выражения.

Выражение (1.18) можно записать в виде

$$t_3 = T_M [6(d + m) + 3 + kN_B]. \quad (1.19)$$

Введем далее показатель эффективности многопроцессорной вычислительной системы $\vartheta_{ij} = t_i/t_j$, характеризующий отношение производительностей вычислительных систем, построенных по разным методам. Полагая, что t_0 — шаг интегрирования в однопроцессорной вычислительной системе, для первого метода имеем

$$\vartheta_{10}(l) = t_0/t_1 = l/(1 + \kappa_{10}), \quad (1.20)$$

где $\kappa_{10} = (l - 1)/[l + 2 + (N_B + N_{II})/6]$. Для второго метода показатель эффективности

$$\vartheta'_{20}(l) = t_0/t_2 = l/(1 + \kappa'_{20}); \quad (1.21)$$

$$\vartheta''_{20}(l) = t_0/t'_2 = l/(1 + \kappa''_{20}), \quad (1.22)$$

где

$$\kappa'_{20} = - (N_{II} + 9)/[6(l + 2) + N_B + N_{II}];$$

$$\kappa''_{20} = - (N_B + 9)/[6(l + 2) + N_B + N_{II}].$$

На рис. 1.29 приведены зависимости показателей эффективности ϑ_{10} , ϑ_{20} многопроцессорных вычислительных систем конвейерного типа в зависимости от порядка системы l . Анализ приведенных зависимостей показывает, что эффективность конвейерных методов построения вычислительных систем растет пропорционально порядку системы дифференциальных уравнений. Второй метод почти вдвое эффективнее первого. Следует, однако, помнить, что для второго метода требуется в два раза больше процессоров. Эффективность вычислительных систем растет при увеличении

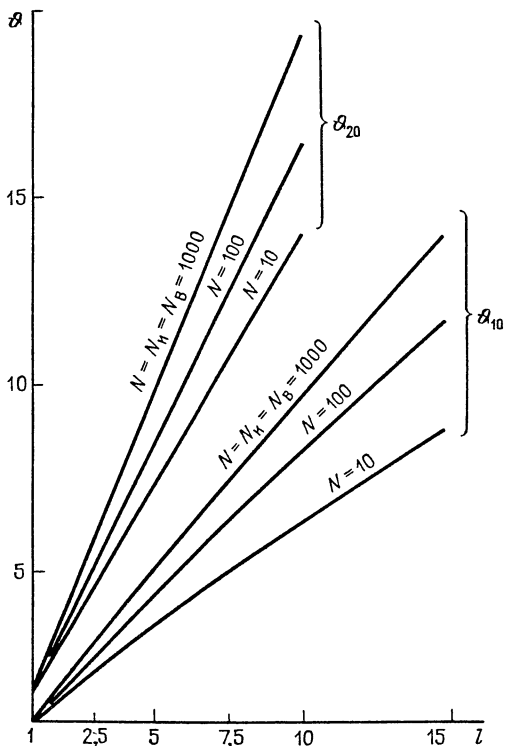


Рис. 1.29. Зависимости показателей эффективности от порядка системы уравнений

числа машинных циклов операций интегрирования и вычисления правых частей уравнений системы (1.9). Рост эффективности замедляется при N_n и N_b , больших ста.

Сравнение второго и третьего методов дает значение показателя ϑ_{32} , определяемое выражением

$$\vartheta_{32} = \frac{6l + 3 + N_b}{6[d + m + N_b(k - 1)/6] + 3 + N_b} \quad (1.23)$$

Так как коэффициент $k < 1$, а величина $N_b \gg 1$, то обычно $d + m + N_b(k - 1)/6 < l$. Следовательно, третий метод эффективнее (по производительности) второго. Точное значение показателя эффективности ϑ_{32} зависит от значений величин d , m , k и N_b , которые могут быть определены для каждого конкретного случая.

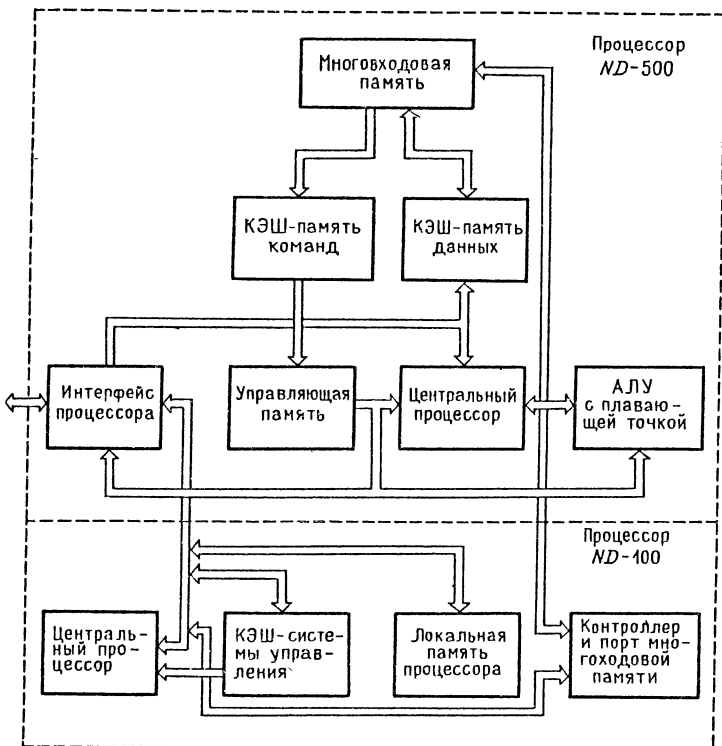
Дальнейшее повышение производительности вычислительных систем при решении дифференциальных уравнений может быть

получено при использовании сдвижокорректирующих методов численного интегрирования, методов поразрядного вычисления операндов и других, приводящих к структурам конвейерного типа [23, 89, 201].

Высокая производительность может быть получена в универсальных многопроцессорных вычислительных системах. Примером такой системы является супермини-компьютер ND-500 [122], содержащий различные варианты кэш-памяти и характеризующийся высокой степенью параллелизма процесса обработки данных.

Компьютер имеет двухпроцессорную базовую архитектуру (рис. 1.30). Один процессор служит для выполнения задач ввода-вывода, операционной системы, а другой выполняет основную вычислительную работу.

Достижению высокой производительности служит применение многоходовой и виртуальной памяти, блоков кэш-памяти. В компьютере имеется аппаратная реализация операций с плаваю-



Р и с. 1.30. Высокопроизводительная двухпроцессорная ЭВМ

щей точкой и двоично-десятичными числами. Для повышения быстродействия в процессоре ND-500 предусмотрены отдельные тракты обработки команд и данных. Благодаря такой структуре можно одновременно адресовать команды, выбирать их, обрабатывать, адресовать и перемещать данные между центральным процессором и памятью.

Все обмены между основной памятью и центральным процессором осуществляются через кэш-память. Для уменьшения времени ожидания центрального процессора в кэш-памяти предусмотрены высокоскоростные буферы, в которых хранятся данные, например, записываемые в память в то время, когда процессор продолжает выполнять основную работу. Кэш-память работает не с физическими адресами, а с адресами виртуальной памяти. Это означает, что ее операции осуществляются параллельно с операциями устройства управления памятью, которое просматривает физические адреса.

Разделение кэш-памяти на секции для команд и данных является одним из факторов, отражающих высокий уровень параллелизма структуры этой системы.

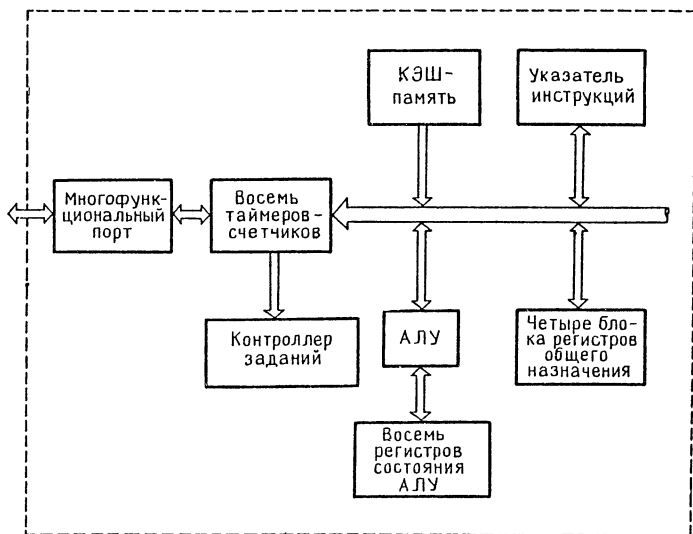
Аппаратные средства планирования задач. Производительность вычислительных систем можно существенно увеличить, если в их структуру включить так называемый планировщик задач, являющийся частью операционной системы реального времени.

Основное преимущество этого подхода заключается в том, что операции смены задач или изменения уровня их приоритета происходят под управлением планировщика задач, размещаемого в кристалле аппаратных средств. Вследствие этого соответствующие операции реализуются исключительно быстро.

Микропроцессор [63], обслуживающий в псевдопараллельном режиме до восьми задач, содержит контекстную память (кэш-память) с восемью указателями инструкций, восемь регистров состояния арифметико-логического устройства (АЛУ), четыре блока регистров общего назначения, контроллер заданий, многофункциональный порт (рис. 1.31).

Для повышения гибкости обработки прерываний в микропроцессоре имеются восемь выводов для подачи запросов на прерывание и восемь 20-разрядных счетчиков-таймеров для управления обработкой с задержкой внешних и внутренних прерываний. Программа решения каждой задачи может маскировать свой запрос перезапуска и управлять своим многофункциональным портом ввода-вывода.

Для повышения быстродействия в микропроцессоре предусмотрена сверхбыстродействующая кэш-память с организацией типа "первый вошел — первый вышел", где могут храниться до 16 программных инструкций.



Р и с. 1.31. Микропроцессор с аппаратной реализацией некоторых функций операционной системы

В отличие от операционных систем, реализованных программно, операционная система реального времени на кристалле аппаратных средств обеспечивает возможность смены задач после каждой инструкции, причем это не сопряжено с дополнительными затратами времени.

Большими функциональными возможностями обладает операционная система ВЕРТЕКС, реализованная в виде кремниевых программных компонентов [233].

Исполнительная программа ВЕРТЕКС обеспечивает планирование задач с управлением по прерываниям: синхронизацию задач и взаимодействие между ними, динамическое и статическое распределение памяти, работу с часами реального времени и т.д. Она является компактной и быстродействующей, требует около 100 мкс для переключения с задачи на задачу.

Базовой логической единицей программы ВЕРТЕКС является задача. Под управлением программы ВЕРТЕКС могут находиться до 256 оригинальных задач с 256 уровнями приоритета.

Задачи могут создавать другие задачи, а также уничтожать, "подвешивать", возобновлять выполнение и менять приоритет для самих себя или других задач.

Ядро операционной системы ВЕРТЕКС предусматривает возможность применения программ обработки прерываний для планирования критически важных задач с управлением по неожиданным

событиям. Максимальное время блокировки прерываний не превышает $16 + 8n$ мкс, где n — количество управляемых задач. Эффективный метод ускоренной передачи данных из внешних устройств в память микропроцессорной системы под управлением операционной системы реализуется следующим образом.

Обычно процесс передачи данных в память сопрягается с ожиданием сигнала "ГОТОВНОСТЬ" от внешнего устройства. Этот процесс часто связан с неоправданным расходом времени, вызванным ожиданием этого сигнала.

Ускорение передачи данных достигается согласованием в цикле чтения данных времени преобразования аналоговой величины в цифровую с временем выполнения команд программы быстрой передачи данных. Цикл ожидания исключается.

В частности, после инициализации (первичного запуска) системы программа входит в цикл чтения данных. При каждом обращении программы к некоторому дешифратору адреса происходит формирование сигнала, запускающего АЦП в новый цикл преобразования. Пока микропроцессор выполняет очередные команды программы передачи данных, цикл преобразования завершается и данные могут быть введены в память.

Для реализации данного метода в состав устройства введены быстродействующий АЦП, два дешифратора состояния адресной шины и генератор импульса пуска преобразователя [187].

Прикладное программное обеспечение. Производительность вычислительных систем связана с качеством программного обеспечения, которое зависит от характеристик используемых алгоритмов обработки информации и квалификации программиста.

Обычно прикладное программное обеспечение формируется на основе использования универсальных алгоритмов и программ, разработанных специалистами для широкого круга задач, и уникальных алгоритмов и программ, созданных в процессе проектирования системы управления на базе микроЭВМ.

Универсальные алгоритмы и программы могут входить в состав математического обеспечения конкретной микроЭВМ, размещаться в фонде алгоритмов и программ, публиковаться в печатных изданиях и т.д.

Универсальные алгоритмы и программы разрабатываются, как правило, специалистами высокой квалификации, и это является залогом их высокой эффективности. Вместе с тем, являясь алгоритмами и программами широкого назначения, они могут с позиций конкретного применения быть чрезмерно сложными, обеспечивать точность существенно выше заданной и т.д. или же, напротив, по каким-то параметрам не удовлетворять разработчика САУ. В этом случае следует разработать свое программное обеспечение, которое в значительной степени может оказаться уникальным.

Примерами универсальных алгоритмов и программ являются, например, программы транспонирования матриц GMTRA, формирования случайных чисел RANDOM и др. [206]. К данной группе программ можно отнести программы умножения и деления, применимые для микроЭВМ, построенной на микропроцессорном комплексе БИС серии КР580 [135].

Программы превосходят все другие аналогичные программы по таким характеристикам, как время выполнения и требуемый для ее хранения объем запоминающего устройства.

Так, например, максимальное время умножения двух 16-разрядных операндов составляет $T_y = 1023 T_M$, где T_M — длительность машинного цикла, а время деления $T_d = 1745 T_M$.

ПОСТРОЕНИЕ СИСТЕМ УПРАВЛЕНИЯ С МИКРОЭВМ

§ 2.1. МИКРОЭВМ (СОСТАВ, НАЗНАЧЕНИЕ, ВЫБОР И ОЦЕНКА)

Особенности подхода к проектированию цифровых автоматических систем. МикроЭВМ своим рождением обязаны микропроцессору, который является ядром микроЭВМ. В свою очередь появление микропроцессоров обусловлено, во-первых, возникновением МОП-технологии, обеспечивающей высокую степень интеграции, и, во-вторых, потребностями создания дешевых цифровых систем, обладающих высокой гибкостью применения.

Создание микропроцессоров требует от разработчиков, во-первых, умения практически использовать для объединения технических средств и программного обеспечения те традиционные способы, которые применялись ранее при разработке больших и сложных вычислительных систем. С появлением микропроцессоров специалисты-системотехники, которые до настоящего времени проектировали малые системы, должны, во-вторых, повысить свою квалификацию до такого уровня, когда они окажутся способными проектировать всю систему в целом, включая разработку программного обеспечения.

Другими словами, классическим разработчикам систем с аппаратной логикой, реализуемых на интегральных микросхемах малой и средней степени интеграции, следует тщательно изучить архитектуру микропроцессоров и микроЭВМ, узнать их особенности, характеристики, научиться обоснованно выбирать те или иные микропроцессорные средства и проектировать на их основе сложные цифровые системы автоматического управления. Причем необходимостью становятся знание и умение программировать не только на языках высокого уровня, но и на машинно-ориентированных языках – ассемблерах.

Микропроцессоры и микроЭВМ обеспечивают высокую гибкость их применения вследствие программируемой логики работы: для изменения назначения зачастую достаточно изменить только программу. Программа работы микроЭВМ является основным и самым дорогим результатом труда проектировщика новой системы.

Разработка программы требует основных затрат времени и интеллектуальных усилий.

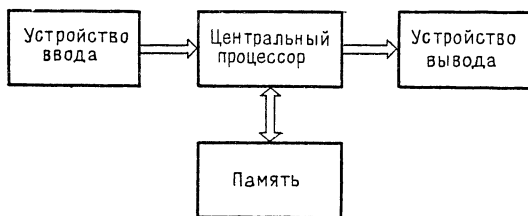
Важным остается также этап схмотехнического проектирования. Однако с появлением микропроцессоров схмотехническое проектирование значительно упростилось: перед проектировщиком возникает относительно простая задача, напоминающая известную игру "Конструктор", где из готовых элементов по заданным правилам составляется изделие. Следует учесть только, что степень готовности этих элементов может быть различной.

Состав и назначение отдельных частей микроЭВМ. Состав микроЭВМ может быть различным. Детальные структурные схемы разных типов микроЭВМ могут значительно отличаться друг от друга. Однако в общем случае микроЭВМ, как и ЭВМ вообще, состоит из центрального процессора, памяти и устройств ввода-вывода (рис. 2.1). Отдельные составные части микроЭВМ соединяются между собой группами линий (шинами), по которым передаются управляющие сигналы и информация (данные, команды, адреса).

Если память является общей для данных и программы, то такая микроЭВМ называется машиной фон-неймановского типа. В ней используется идея Дж. фон Неймана, датой рождения которой считается 1946 г., о том, что с командами можно обращаться так же, как с информацией. Тогда программа может модифицироваться сама в ходе ее исполнения. С помощью команд перехода или ветвления выполнение или невыполнение конкретных частей программы будет зависеть от результата определенных операций.

На основе этой идеи была разработана первая ЭВМ с так называемой хранимой программой UNIVAC-1. В этой ЭВМ информация (данные) и программа хранились в памяти и включались в работу по мере необходимости.

Деление больших и миниЭВМ на отдельные составные части обычно не совпадает с делением микроЭВМ на составные части. Так, например, в ЭВМ обычно выделяют самостоятельными частями арифметическое устройство и устройство управления. В микроЭВМ такое деление не производится.



Р и с. 2.1. Структурная схема микроЭВМ

Рассмотрим назначение и принципы организации отдельных частей микроЭВМ.

В структуре микроЭВМ обычно выделяют центральный процессор, который представляет собой сочетание устройства управления (УУ), арифметико-логического устройства (АЛУ) и связанных с ними регистров.

Такое толкование термина "центральный процессор" объясняется тем, что в микроЭВМ устройство управления, АЛУ и регистры часто реализуются в одном кристалле БИС и образуют микропроцессор.

Назначением центрального процессора (микропроцессора) является обработка информации, которая осуществляется в АЛУ под действием сигналов устройства управления.

Арифметико-логическое устройство предназначено для выполнения арифметических, логических и сдвиговых операций над числами. Эти операции выполняются под контролем устройства управления. АЛУ может обрабатывать информацию последовательно (по одному разряду), параллельно (по словам), либо последовательно-параллельно. АЛУ осуществляет арифметические и логические операции, как правило, только над двумя числами. Устройство управления в соответствии с программой, хранимой в памяти, последовательно производит выборку отдельных команд программы, затем, согласно этим командам, управляет составными частями микроЭВМ, извлекает и пересылает для хранения и обработки операнды. В соответствии с программой устройство управления организует также работу устройств ввода-вывода. Таким образом, устройство управления с помощью программы управляет потоками информации в микроЭВМ.

Регистры микропроцессора предназначены для временного хранения данных и команд. Важным является то, что содержимое любого регистра может быть операндом. Команды, оперирующие с содержимым регистров микропроцессора, оказываются короткими и выполняются за минимальное время, так как цикл считывания операнда из памяти в них отсутствует.

Занесение входных данных в микропроцессор и в память микроЭВМ является задачей, решаемой устройствами ввода, которые осуществляют связь между микроЭВМ и остальной частью системы автоматического управления. Устройства ввода имеют, как правило, цифровой вход. Они позволяют вводить информацию последовательным или параллельным кодом.

Устройства вывода обеспечивают вывод полученных результатов вычислений (в виде электрических сигналов) во внешние устройства, являющиеся потребителями "интеллектуальной" продукции микроЭВМ. Устройства вывода, так же как и устройства ввода, обеспечивают связь микроЭВМ, являющейся устройством цифро-

вой обработки информации, с соответствующими блоками САУ, например, с цифро-аналоговым преобразователем и усилителем мощности управляющих сигналов. Часто устройства ввода и вывода называют портами.

Входные и выходные данные, результаты промежуточных вычислений и программы записываются и хранятся в памяти. Память подразделяется на основную, внешнюю и регистровую. Основная память включает память данных и память программ.

В памяти программ хранятся команды и константы. Если предполагается использовать микроЭВМ для решения определенного ограниченного круга задач (в простейшем случае — для решения одной задачи), то программа может постоянно храниться в основной памяти. Основная память является постоянным запоминающим устройством (ПЗУ) микроЭВМ. Часто в микроЭВМ функции ПЗУ выполняет перепрограммируемая память (ППЗУ).

Входные и выходные данные, результаты промежуточных вычислений хранятся в памяти данных, которая является оперативной (ОЗУ).

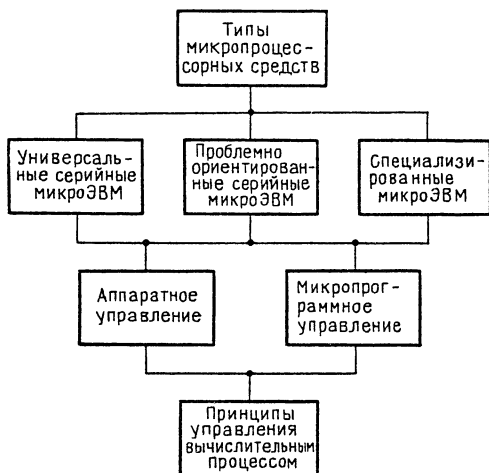
Если оперативная память должна хранить большое количество информации, записывать и считывать эту информацию с высоким быстродействием, сравнимым с быстродействием АЛУ, то тогда используют внешнюю память (например, внешние запоминающие устройства на магнитных лентах или магнитных дисках). Стоимость хранения одного разряда информации в такой памяти значительно ниже, чем в памяти с произвольной выборкой, используемой в качестве оперативной памяти данных.

Информация в памяти хранится в виде отдельных слов, причем каждое слово содержит от 4 до 65 разрядов. Одни типы микроЭВМ обрабатывают слова, имеющие постоянную, т.е. фиксированную длину, а другие — слова с переменной длиной (кратной чаще всего восьми разрядам).

Регистры представляют собой вспомогательную память для временного хранения информации, распределенную по всей микроЭВМ. Регистры имеются, например, в устройствах ввода-вывода.

Основные характеристики, принципы выбора и оценка микроЭВМ. Одним из ответственных этапов проектирования систем автоматического управления с микроЭВМ является выбор микропроцессорных средств.

Понятие "микропроцессорные средства" не является устоявшимся и общепризнанным. Обычно под микропроцессорными средствами понимают совокупность БИС центрального процессора, микросхем памяти, ввода-вывода, таймеров, генераторов синхронизации и других программно и аппаратно совместимых интегральных микросхем, необходимых и достаточных для создания мик-



Р и с. 2.2. Классификация микропроцессорных средств

роЭВМ. Тогда под выбором микропроцессорных средств понимают выбор не только центрального процессора, но и всего комплекта БИС, используемого в микроЭВМ.

Этот подход к выбору микропроцессорных средств является оправданным, например, при разработке микроЭВМ.

В прикладных задачах, связанных с применением средств вычислительной техники в системах автоматического управления, понятие "микропроцессорные средства" целесообразно отождествлять с понятием "микроЭВМ", разделяя последние по их функциональному назначению и методам организации вычислительного процесса (рис. 2.2).

С этих позиций к микропроцессорным средствам можно отнести серийные микроЭВМ универсального назначения, серийные проблемно ориентированные микроЭВМ и специализированные микроЭВМ.

Серийные микроЭВМ изготавливаются промышленностью и поставляются потребителю по его запросу в комплекте, включающем, например, наряду с микроЭВМ также средства отладки программного обеспечения, программирования памяти и т.д. Это обуславливает относительно низкую стоимость и сравнительно невысокую степень сложности проектирования САУ: длительность и трудоемкость этапа схемотехнического проектирования микроЭВМ резко сокращаются, так как они сводятся к несложной задаче сопряжения микроЭВМ с остальной частью САУ. Наличие развитого математического обеспечения и отладочных средств снижает трудозатраты при разработке программного обеспечения.

В то же время серийные микроЭВМ по своим функциональным возможностям не всегда могут перекрыть тот круг задач, которые поставлены перед ними. Например, недостаточными могут оказаться быстродействие и (или) длина разрядной сетки, объем адресуемой памяти, несовершенной может быть система команд и т.д. Следует также учитывать, что серийные микроЭВМ могут обладать недостатком противоположного характера, связанным с наличием функциональной избыточности. Так, с позиций конкретного применения микроЭВМ может иметь чрезмерно высокое быстродействие и (или) длину разрядной сетки, объем адресуемой памяти, существенно превосходящий требуемый объем, и т.п.

В обоих приведенных случаях необходимо применять специализированные микроЭВМ, так как отмеченная функциональная недостаточность делает невозможным, а функциональная избыточность — экономически не оправданным выбор для решения соответствующих задач серийных микроЭВМ. Термин "специализированные микроЭВМ" не является строгим, поскольку в данном случае речь идет преимущественно о специализированных микропроцессорных устройствах обработки информации. Однако учитывая, что по функциональному назначению, составу, методам организации взаимодействия с внешней средой эти устройства имеют много общего с серийными микроЭВМ, применяемыми для обработки информации в системах автоматического управления, будем ориентироваться на применение этого термина.

Специализированные микроЭВМ разрабатываются в ходе проектирования разработчиками САУ. Они обладают, как правило, функциональными возможностями, строго ориентированными на их конкретное применение. Система команд и структура специализированных микроЭВМ являются в этом смысле функционально ориентированными. Вследствие этого специализированные микроЭВМ высокоэффективны, например, при решении задач управления в реальном времени (подробнее см. § 1.4). Однако создание таких микроЭВМ является более трудоемкой задачей, так как в этом случае разрабатывается полный комплекс вопросов, связанных как со схемотехническим проектированием, так и с проектированием программного обеспечения.

С точки зрения разработчика САУ с микроЭВМ важным является метод организации вычислительного процесса. В серийных, проблемно ориентированных и в специализированных микроЭВМ могут быть использованы как аппаратный, так и микропрограммный принцип управления вычислительным процессом.

Аппаратный принцип управления основан на использовании фиксированной (жесткой, аппаратной) логической схемы управления вычислительным процессом, реализованной в жестких связях между элементами микропроцессора и не допускающей ее измене-

ния при проектировании САУ. Такое управление называется внутренним микропрограммированием. Микропрограммы образуют список команд. Аппаратный принцип управления реализуется в однокристальных микропроцессорах. Эти микропроцессоры имеют относительно невысокое быстродействие. Длина слова и состав набора команд фиксированы. Связанная с этим негибкость исключает возможность использования микроЭВМ на базе микропроцессоров с аппаратным принципом управления в тех прикладных задачах, в которых нужны высокое быстродействие или специальные команды.

Специальные команды можно реализовать с помощью микрокоманд и подпрограмм. Однако такой подход обычно не удовлетворяет требованиям по быстродействию. Достоинствами микропроцессорных средств с аппаратным принципом управления являются относительная легкость и доступность схмотехнического и программного проектирования. Это связано с тем, что микропроцессор и система команд поставляются потребителям в готовом виде в форме БИС, в которой схемами с жесткой логикой реализована совокупность базовых команд.

Однокристальные микропроцессоры выполняются преимущественно по МОП-технологии. Вследствие этого плотность упаковки элементов в кристалле оказывается высокой, что позволяет создавать микропроцессоры с длиной слова в восемь, двенадцать и шестнадцать битов и с быстродействием до 500 тысяч операций в секунду. Дальнейшее и резкое повышение быстродействия микропроцессорных средств достигается применением биполярной технологии. Однако вследствие уменьшения степени интеграции микросхем, увеличения потребляемой энергии и рассеиваемой мощности микропроцессоры, выполненные по биполярной технологии, организуются из нескольких кристаллов. Эти микропроцессоры имеют наращиваемую длину разрядной сетки, вычислительный процесс в них реализуется под управлением микропрограмм.

Микропрограммный принцип управления основан на использовании внешнего микропрограммирования и реализуется одним из трех способов: с использованием постоянного набора команд; с использованием изменяемого набора команд; с использованием постоянного набора команд, который можно дополнять другими "заказными" командами.

Микропрограммное управление с постоянным набором команд применяют, как правило, в серийных микроЭВМ, например, в микроЭВМ LSI-11. С точки зрения потребителя системы команд микроЭВМ, выполненных на базе микропроцессора с аппаратным принципом управления, и микроЭВМ, выполненных на основе микропрограммного управления с постоянным набором команд,

являются функционально равнозначными, так как образованы совокупностями базовых команд.

В настоящее время микроЭВМ с изменяемыми наборами команд являются преимущественно специализированными. МикроЭВМ с дополняемым набором команд выпускаются серийно (например, микроЭВМ семейства "Электроника С5"). В этих микроЭВМ разработчиками может быть реализована любая система команд. Для ее изменения обычно достаточно перепрограммировать управляющую память. Однако в специализированных микроЭВМ система команд может зависеть от их структуры. Это значит, что изменение набора команд может быть сопряжено с необходимостью внесения изменений и в структуру микроЭВМ.

МикроЭВМ с микропрограммным принципом управления реализуются на основе комплектов БИС серий К581, К584, К589 и др. [61]. Они обладают высоким быстродействием, имеют наращиваемую длину разрядной сетки, в них может быть реализован изменяемый набор команд. В этом их достоинства. Недостатки связаны с относительно высокой стоимостью, сложностью схемотехнического проектирования и разработки программного обеспечения.

Рассмотрим факторы, влияющие на выбор микропроцессорных средств, более подробно. Одним из важнейших является соответствие между требованиями проектной задачи и возможностями, предоставляемыми микропроцессорными средствами. При этом быстродействие и длина разрядной сетки процессора микроЭВМ играют решающую роль.

Быстродействие зависит от тактовой частоты, числа тактов, необходимых для выполнения различных команд, удельного веса этих команд в тех задачах, которые необходимо решать процессору. В зависимости от длины разрядной сетки, структуры микропроцессора и структуры системной шины микроЭВМ, схем ввода-вывода, организации прерываний и т.д. одна и та же программа, составленная для различных микроЭВМ одного класса, займет различное число ячеек памяти и потребует для выполнения разное время. Поэтому правильность выбора микропроцессорных средств может быть достоверно установлена только после завершения схемотехнического проектирования и программирования микроЭВМ, когда имеется готовая программа и возможна точная оценка времени ее выполнения.

Необходимо учитывать также следующее обстоятельство. Программирование выполняется для разных микропроцессорных средств по-разному. Например, однокристалльные микропроцессоры с аппаратным принципом управления программируются на основе списка базовых команд, а программирование многокристалльных микропроцессоров с изменяемым набором команд производится

на уровне микрокоманд и с учетом структуры микроЭВМ, которая к началу программирования должна быть уже разработана.

Изложенные соображения показывают, насколько существенными могут оказаться потери, вызванные неправильным первоначальным выбором микропроцессорных средств.

На практике для обеспечения гарантированного успеха при выборе микропроцессорных средств поступают следующим образом. Ориентировочно оценивают число и суммарную длительность некоторых типовых операций, содержащихся в алгоритме, например, операций типа "умножение" в алгоритме быстрого преобразования Фурье. Оценивают суммарное число и длительность нетиповых операций, таких, например, как операции ввода-вывода, используемые однократно, и т.п. С учетом полученных данных вычисляют итоговую длительность цикла выполнения программы которую затем умножают на коэффициент запаса, равный 1,5–2.

Полученная длительность цикла выполнения программы имеет предварительный характер. Если ее величина удовлетворяет необходимой скорости обработки информации микроЭВМ, то результат расчета может служить основанием для перехода к детальному

Т а б л и ц а 2.1

Наименование микропроцессорных средств	Тип модели микроЭВМ или цифрового устройства	Принцип управления	Возможность изменения набора команд	Базовый микропроцессорный комплект БИС (серия)	Технология	Быстродействие микроЭВМ, тыс. опер/с
1. Универсальные серийные микроЭВМ	Электроника С5-21	Микропрограммное	есть	К586	л-МОП	200
	Электроника НЦ-80-01	Микропрограммное	есть	—	КМОП	550/250
	Электроника 60	Микропрограммное	нет	К581	л-МОП	250
	Электроника К1-20	Аппаратное	нет	К580	л-МОП	500
2. Проблемно ориентированные микроЭВМ	К145ИК19	Аппаратное	нет	К145	р-МОП	—
	К145ИК18	Аппаратное	нет	К145	р-МОП	—
	К145ВГ1	Аппаратное	нет	К145	р-МОП	—
3. Специализированные микроЭВМ	Определяется разработчиком	Аппаратное	нет	К580	л-МОП	500
		Микропрограммное	есть	К584	И ² Л	до 200
		Микропрограммное	есть	К589	ТТЛШ	до 2000

Примечание. В табл. приняты следующие сокращения: ЦПЭ — центральный процессорный элемент; 1К = 2¹⁰ (в графе "Емкость адресуемой памяти").

схемотехническому проектированию и разработке программного обеспечения.

Расчет длительности цикла выполнения программы основывается на использовании данных, характеризующих быстродействие микропроцессорных средств. Обычно в справочниках приводятся длительность такта или быстродействие. Длительность такта в 4–6 раз меньше величины, обратной быстродействию. Это объясняется тем, что быстродействие измеряется числом элементарных операций (операций типа регистр – регистр), совершаемых в 1 с. Каждая элементарная операция выполняется в среднем в течение 4–6 тактов.

В табл. 2.1 приведены основные характеристики некоторых типов микропроцессорных средств, которые необходимо учитывать при их выборе в качестве элементов САУ. В основу построения табл. 2.1 положен признак, характеризующий функциональные возможности микроЭВМ. Здесь приведены основные параметры как некоторых серийных микроЭВМ, так и потенциальные характеристики специализированных микроЭВМ, проектирование которых выполняется на серийных микропроцессорных комплектах БИС. Важное значение имеет технология изготовления микропроцессо-

Разрядность (наращиваемая), бит	Емкость адре- суемой памяти, бит	Потребляемая мощность, Вт	Габаритные разме- ры, мм	Масса, кг
16	32К × 16	20	302 × 252 × 29	1,2
16/32	16К × 16	10	180 × 390 × 20	0,3
16	32К × 16	≤ 25	483 × 550 × 132	20
8	64К × 8	–	Плата: 220 × 320	–
8	–	–	–	–
8	–	–	–	–
8	4000 × 8	–	–	–
8	64К × 8	0,75 (ЦПЭ)	–	–
(4)	0,5К слов	–	–	–
(2)	0,5К слов	0,9 (ЦПЭ)	–	–

ров. Различные технологические процессы позволяют создавать микропроцессоры с характеристиками, изменяющимися в широком диапазоне значений.

Например, наибольшее быстродействие обеспечивает технология ТТЛШ (транзисторно-транзисторная логика с диодами Шоттки) — длительность такта составляет 0,1 мкс. Однако эта технология не позволяет создавать однокристалльный микропроцессор с длиной разрядной сетки в 8 и более разрядов. Поэтому в однокристалльных микропроцессорах используется *n*-МОП-технология, позволяющая получить длительность такта около 0,5 мкс.

В табл. 2.1 приведены характеристики универсальных серийных микроЭВМ "Электроника С5-21", "Электроника НЦ-80-01", "Электроника К1-20" и "Электроника 60" [30, 38, 48, 94].

МикроЭВМ "Электроника С5-21" является представителем семейства микроЭВМ "Электроника-С5". В состав семейства входят многоплатная микроЭВМ "Электроника С5-02", поставляемая в конструктивно и функционально законченном виде, одноплатные (С5-12, С5-21, С5-41), предназначенные для встраивания в системы с целью реализации функций контроля, управления, обработки данных, и однокристалльная микроЭВМ "Электроника С5-31" [48].

Особенностью организации микроЭВМ этого семейства является развитая система ввода-вывода, позволяющая организовать работу в селекторном и мультиплексном режимах. Селекторный режим с прямым доступом к памяти служит для подключения быстродействующих устройств, а мультиплексный — для подключения большого числа низкоскоростных устройств. МикроЭВМ комплектуются специальными функциональными модулями (аналого-цифровой преобразователь — АЦП, модуль цифрового ввода-вывода — ЦВВ, модули оперативной и перепрограммируемой памяти — ОЗУ и ППЗУ, модуль пульта управления и т.п.).

Все машины семейства программно совместимы. В состав резидентного программного обеспечения входят загрузчик, автокод, программы автономного обмена с внешними устройствами, библиотека стандартных программ, контрольно-профилактические тесты и транслятор с языка БЕЙСИК. Кроссовые средства предназначены для ЭВМ БЭСМ-6; ЕС ЭВМ, М-4030, М-220.

Система команд микроЭВМ семейства является достаточно развитой. Используются команды трех форматов. Исполнительные адреса формируются с использованием непосредственной, прямой и косвенной адресации относительно счетчика команд, а также индексации и автоиндексации. Адреса каналов ввода-вывода введены в общую память, что позволяет обращаться с каналами ввода-вывода как с ячейками памяти.

МикроЭВМ "Электроника НЦ-80-01" является одноплатной машиной семейства "Электроника НЦ". Это семейство включает также многоплатные микроЭВМ (НЦ-ОЗТ, НЦ-ОЗД, НЦ-31, НЦ-О4Т) и однокристальную микроЭВМ "Электроника НЦ-80" [38, 44]. Это семейство характеризуется совместимостью моделей ряда по схемотехнике и программному обеспечению, модульностью средств и наличием внешнего микропрограммного управления. Каждая микроЭВМ ряда строится как совокупность модулей, объединенных системной шиной.

Программное обеспечение включает кроссовые и резидентные средства. Резидентные средства состоят из трех систем: дисковой операционной, диалоговой системы БЕЙСИК с компилятором и интерпретатором и перфоленточной, состоящей из операционной системы однозадачного режима, ассемблера, редактора текста, системы отладки и библиотеки стандартных программ. Кроссовые средства предназначены для ЭВМ БЭСМ-6 и ЭВМ единой серии.

МикроЭВМ семейства "Электроника 60" являются многоплатными моделями, поставляемыми в конструктивно и функционально законченном виде. Эти микроЭВМ характеризуются программной совместимостью с миниЭВМ (М-400, СМ-3, СМ-4, "Электроника-100/25", "Электроника-100/16И") и возможностью использования программного обеспечения известных и доступных миниЭВМ [30, 31].

Все функциональные блоки ЭВМ (центральный процессор, память, модули ввода-вывода) выполнены в виде конструктивно законченных устройств, связь между которыми осуществляется через единый канал. В состав микроЭВМ входят модули центрального процессора М1 или М2 (в модуле М2 дополнительно реализованы команды расширенной арифметики и операции с плавающей запятой), модуль динамического ОЗУ емкостью 4К × 16 битов, модуль динамического ОЗУ емкостью 16К × 16 битов, модуль постоянной памяти (ПЗУ) емкостью 2К × 16 битов, модуль ППЗУ емкостью 4К × 16 битов, модули В1 и В2 для связи с электрической пишущей машинкой типа "Консул-260", считывателем с перфоленты FS-1501 и перфоратором типа ПЛ-150М (В2). Параллельный интерфейс представлен модулем И1. Имеется контроллер прямого доступа к памяти и интерфейс пользователя.

Внешние устройства включают модуль АЦП на 8 дифференциальных или 16 однопроводных каналов с разрешающей способностью 10 двоичных разрядов и модуль ЦАП. Период дискретизации аналогового сигнала (время преобразования на один канал) 100 мкс, время установления ЦАП 10 мкс при разрешающей способности также 10 битов.

Программное обеспечение включает: перфоленточную операционную систему, состоящую из абсолютного загрузчика, редактора текста, перемещающего ассемблера, отладчика, компоновщика, супервизора ввода-вывода, программы печати, пакета стандартных программ; трансляторы языков Бейсик, Фортран и Квейсик: полный набор диагностических программ для проверки функциональных модулей и дисковую операционную систему для работы в реальном масштабе времени [14].

Отметим, что зарубежным аналогом микроЭВМ "Электроника 60" является микрокомпьютер LSI-11 (миникомпьютер PDP-11) корпорации Digital Equipment. В работе [34] отмечается, что, архитектура PDP-11 оказывает сильное влияние на новейшие разработки цифровых машин, в частности микрокомпьютеров. В результате сравнения ряда получивших широкое признание архитектур цифровых машин, выполненного объединенным комитетом армии и ВМФ США по выбору архитектуры для машин военного применения, предпочтение было отдано архитектуре семейства PDP-11.

По указанным причинам в монографии основное внимание уделено вопросам применения в системах автоматического управления универсальных серийных микроЭВМ семейства "Электроника 60", с которыми программно совместима микроЭВМ "Электроника 81Б", имеющая быстродействие до 1 млн. операций в секунду, и микроЭВМ "Электроника НМС 11100.1" [61].

МикроЭВМ "Электроника К1-20" относится к семейству микроЭВМ и контроллеров серии "Электроника К" и предназначена для использования в системах управления и контроля, а также для отладки программного обеспечения на уровне машинных команд и в качестве лабораторной микроЭВМ [94].

По конструктивному исполнению и назначению эта микроЭВМ является одноплатной. Она может работать как с пультом, так и автономно. МикроЭВМ укомплектована стандартными программами, записанными в ПЗУ или ППЗУ. Среди них — программа монитор, обеспечивающая управление взаимодействием оператора с микроЭВМ через пульт в режиме реального масштаба времени, тест-программа самоконтроля, библиотека подпрограмм вычислений с плавающей запятой, включающая подпрограммы для вычисления четырех арифметических действий, прямых и обратных тригонометрических функций и операций $\sqrt{\quad}$, $x!$, e^x , $\ln x$, a^x .

Структура и состав микроЭВМ "Электроника К1-20" позволяют создавать экономичные системы управления, обладающие возможностями, характерными для систем управления на базе мини- и микроЭВМ, такими как управление в режиме реального масштаба времени, быстрая реакция на внешние изменения, сбор и обработка цифровой и число-импульсной информации, прямое цифровое

программное управление, возможность наращивания системы при помощи дополнительных плат сопряжения с устройствами связи с объектами и устройствами памяти, возможность организации систем на базе мини- и микроЭВМ.

Система программирования микроЭВМ включает наряду с резидентными также кросс-средства, реализованные на ЭВМ единой серии и БЭСМ-6. МикроЭВМ "Электроника К1-10" выполнена на микропроцессорном комплекте БИС серии К580. Этот комплект включает однокристалльный 8-разрядный процессор (КР580ИК80А), программируемые последовательный и параллельный интерфейсы (КР580ИК51 и КР580ИК55), программируемый таймер (КР580ВН53), контроллер прямого доступа к памяти (КР580ИК57), программируемый контроллер прерываний (КР580ВН59) и тактовый генератор (КР580ГФ24).

Комплект предназначен для построения широкого класса средств вычислительной техники и радиоэлектронной аппаратуры. Он характеризуется архитектурным единством, заключающимся в единстве интерфейса всех БИС, программируемости, универсальности применения и в возможности наращивания вычислительной мощности [94]. Комплект БИС серии К580 является одним из самых популярных, так как наличие у микропроцессора этой серии фиксированного базового набора команд и развитого математического обеспечения [65] облегчает составление прикладных программ.

Микропроцессорный комплект БИС серии К584 выполнен по интегральной инжекционной технологии (И²Л), обеспечивающей быстроедействие до 0,5 млн. операций в секунду. Эта технология позволяет по сравнению с технологией ТТЛШ снизить на порядок рассеиваемую мощность и вдвое увеличить степень интеграции.

Центральный процессорный элемент (ЦПЭ) К584ИК1 этой серии, основанный на микропрограммном управлении, позволяет проектировать процессоры с разряднонаращиваемой структурой. Длина разрядной сетки кратна четырем.

Функциональные возможности ЦПЭ характеризуются его способностью выполнить за один временной такт, равный 2,0 мкс, любую из стандартных операций модификации операндов или их комбинаций в восьми арифметических или в восьми булевских функциях, обычных циклических, арифметических и логических сдвигов операндов двойной длины, комбинации операций и сдвигов для упрощения итеративных алгоритмов умножения и деления, специальные операции пересылки. В ЦПЭ возможно выполнение операций с двойной точностью.

Центральный процессорный элемент К584ИК1 программируется на основе внешнего микропрограммирования. Это значит, что каждая команда представляется в виде микропрограммы, составлен-

ной из микроинструкций. Совокупность микропрограмм образует рабочую программу.

В состав микропроцессорного комплекта серии К584 входят центральный процессорный элемент (К584ВУ1), блок микропрограммного управления (К584УМ1), контроллер состояния (К584УМ2), магистральный приемопередатчик (К584КП1) [7].

Микропроцессор может быть образован с использованием БИС ЦПЭ, блока микропрограммного управления и памяти. В списке имеются 52 микрокоманды, образующие группы арифметико-логических, арифметических операций, операций сдвига и пересылки данных.

Микропроцессорный комплект серии К589 обладает высоким быстродействием. Он позволяет создавать различные системы команд на основе набора микрокоманд, а также изменять структуру микропроцессорной системы и длину разрядной сетки, приспособивая их под алгоритм управления и заданную точность. Этим объясняется широкое распространение комплекта серии К589 в различных сферах, основными из которых являются системы числового программного управления станками, высокопроизводительные системы обработки данных, быстродействующие автоматические системы и др.

В состав комплекта входят следующие БИС [16]: блок микропрограммного управления (БМУ) К589ИК01, центральный процессорный элемент (ЦПЭ) К589ИК02, схема ускоренного переноса (СУП) К589ИК03, многорежимный буферный регистр (МБР) К589ИР12, блок приоритетных прерываний (БПП) К589ИК14, шинный формирователь (ШФ) К589АП16 и шинный формирователь с инверсией (ШФИ) К589АП26.

Простейший микропроцессор в данном случае может быть образован совокупностью БИС ЦПЭ, БМУ и памяти (её называют управляющей памятью (УП)). Длина разрядной сетки определяется числом используемых кристаллов ЦПЭ и кратна двум.

Для микропроцессора этой серии используется внешнее микропрограммирование. Формат микроинструкций превышает 20 битов и достигает 48 и более битов. В списке имеется 46 микрокоманд, образующих группы микрокоманд пересылок, арифметических, логических микрокоманд и микрокоманд управления и сдвига. Список микрокоманд является достаточно совершенным, что позволяет создавать системы команд, эмулирующие команды серийных ЭВМ или эффективно приспособленные для конкретной задачи.

Микропрограммы для микропроцессоров на комплекте БИС серии К589 разрабатываются так же, как и для микропроцессоров на комплекте БИС серии К584. Однако система микрокоманд микропроцессора, выполненного на БИС серии К584, является более мощной.

В табл. 2.1 приведены основные характеристики однокристалльного микропроцессора серии K580 и микропроцессорных комплектов БИС серий K584 и K589, которые можно рекомендовать для применения в специальных микроЭВМ. Следует, однако, иметь в виду, что отечественная промышленность выпускает более десяти различных микропроцессорных комплектов БИС, выполненных по разным технологиям и позволяющих удовлетворить потребности широкого круга потребителей. Например, микропроцессорный комплект серии K1800, выполненный по технологии ЭСЛ (эмиттерно-связанная логика), имеет цикл 0,07 мкс, наращиваемую разрядность, кратную четырем.

Примером однокристалльного 16-разрядного микропроцессора с быстродействием около 500 тыс. операций в секунду является микропроцессор серии K1801 [85]. Этот микропроцессор имеет емкость ПЗУ 16К битов.

По технологии ТТЛШ, обеспечивающей длительность цикла выполнения микрокоманды, равную 0,12 мкс, выполнены микропроцессорные комплекты серий КР1802 и КР1804.

Эти комплекты позволяют создавать микропроцессоры с наращиваемой разрядностью, кратной соответственно восьми и четырем [19, 61].

Среди однокристалльных микроЭВМ особое место занимают модели, проблемно ориентированные на решение задач управления внешними объектами [93]. По степени универсальности использования их подразделяют на специализированные, работающие по жесткой программе, и модели широкого применения, программа действия которых заносится во внешнее ЗУ и может изменяться самим пользователем или по картам-заказам, составленным пользователем.

Перепрограммируемые БИС микроЭВМ двух семейств серии K145 (табл. 2.1) представляют собой однокристалльные микроЭВМ последовательного действия с использованием принципа многоуровневого программирования. МикроЭВМ адаптируются к внешним устройствам как по формату управляющих команд, так и по временным характеристикам.

Для реализации множества задач управления в микроЭВМ используется специальная система команд, обеспечивающая управление внешними устройствами и выполнение программы. Система команд микроЭВМ позволяет организовать как разомкнутую систему управления объектами по жесткой программе, так и замкнутую с большой сетью внутривнутрипрограммных ветвлений в соответствии с условиями, задаваемыми по времени и состоянию датчиков.

Проблемно ориентированные микроЭВМ серии K145 обладают относительно невысоким быстродействием, достаточным, однако, для решения многих задач управления.

Микропроцессор с аппаратным принципом управления. В микроЭВМ применяются микропроцессоры с двумя принципами управления: аппаратным и микропрограммным.

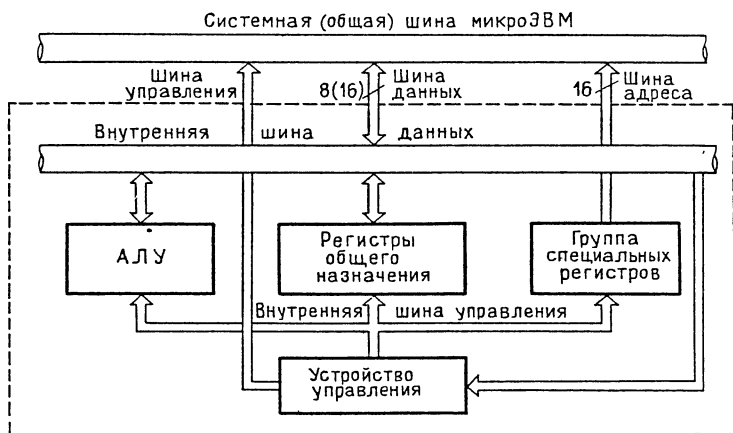
Аппаратное управление используется в большинстве однокристалльных микропроцессоров. Оно основано на внутреннем микропрограммировании. При аппаратном управлении система команд микропроцессора является фиксированной, она реализована во внутренних жестких электрических связях в кристалле микропроцессора и не может быть изменена разработчиком САУ.

Микропроцессоры с аппаратным управлением включают в свой состав (рис. 2.3) арифметико-логическое устройство (АЛУ), группу регистров общего назначения (РОН), группу специальных регистров, устройство управления и, как правило, внутренние шины данных и управления, внешние шины адреса, управления, данных.

Назначением АЛУ является обработка информации, совершаемая в форме некоторых операций над данными. Последовательность и характер операций, производимых над данными, задаются устройством управления.

Устройство управления оперирует с управляющими сигналами, которые можно разделить на две группы: внутренние сигналы управления и внешние сигналы управления. Внешние сигналы управления состоят из сигналов, формируемых устройством управления, и сигналов, поступающих в микропроцессор с системной шины микроЭВМ.

Внутренние сигналы управления формируются устройством управления в составе и последовательности, определяемыми результатом декодирования команды дешифратором команд, вхо-



Р и с. 2.3. Схема микропроцессора с аппаратным принципом управления

дящим в устройство управления. На характер этих сигналов оказывают влияние также внешние сигналы, поступающие в микропроцессор. Состав внутренних сигналов и их временная последовательность являются строго определенными. Так проявляется сущность внутреннего микропрограммирования. Под действием внутренних сигналов управления осуществляются все необходимые пересылки данных, а также операции над данными, обеспечивается адресация регистров внешних устройств и ячеек памяти.

Внешние сигналы управления, поступающие из микропроцессора на шину управления, используются для управления памятью и устройствами ввода-вывода.

Группа регистров общего назначения образует сверхоперативную память микропроцессора. Эти регистры используются для временного хранения операндов. Они могут быть использованы также в качестве указателей адресов, по которым хранятся операнды, или как индексные регистры.

С количеством регистров общего назначения часто связывают вычислительные возможности микропроцессоров. В настоящее время существует тенденция увеличения числа регистров общего назначения [175]. Обычно в микропроцессоре число регистров общего назначения не превышает 16. Однако их количество может достигать 64. Все регистры общего назначения являются программно доступными.

Среди регистров общего назначения часто выделяют один или несколько операционных или накопительных регистров, команды с участием которых совершаются наиболее быстро. Этим регистрам присваивают имена, такие, например, как аккумулятор, или вводят специальные обозначения: А, Т и другие. В некоторых случаях накопительные регистры выводят из групп регистров общего назначения и выделяют в особую группу. Так сделано, например, в микропроцессоре КР580ИК80.

Среди специальных регистров отметим счетчик команд, регистр-указатель стека и регистр признаков результата операции, или регистр состояния.

Счетчик команд содержит адрес команды, выбираемой из запоминающего устройства и выполняемой в текущем цикле. Содержимое счетчика команд увеличивается после выполнения команды на число, зависящее от формата этой команды и способа организации памяти. Например, в микроЭВМ "Электроника К1-20" однобайтовым командам соответствует увеличение содержимого счетчика команд после их выполнения на единицу, двухбайтовым командам — на два и трехбайтовым командам — на три.

Регистр-указатель стека содержит адрес первой свободной ячейки в стеке. Стек — это память магазинного типа, в которой выполняется принцип "первый вошел — вышел последний". Это значит,

что в ячейки стека информация заносится последовательно и извлекается в порядке, обратном порядку занесения. Стек входит чаще всего либо в состав основной оперативной памяти, либо выполняется на совокупности внутренних регистров. В последнем случае регистр-указатель стека не требуется.

Стековая организация памяти позволяет повысить вычислительную мощность микроЭВМ при обработке прерываний, работе с внутренними программами, выполнении программ ввода-вывода. Использование стека является эффективным средством обработки взаимосвязанных структур данных и данных с многоуровневым вложением.

Проиллюстрируем работу стека при обращении к подпрограмме. Это обращение требует прекращения выполнения основной программы. Перед тем как приступить к выполнению подпрограммы, микропроцессор должен сохранить адрес очередной команды основной программы, чтобы после выполнения подпрограммы продолжить выполнение основной программы. Для сохранения адреса очередной команды микропроцессор переписывает содержимое счетчика команд по команде ПЕРЕЙТИ К ПОДПРОГРАММЕ в стек, запоминая в регистре-указателе стека адрес очередной не занятой ячейки стека. Это позволяет микропроцессору при завершении подпрограммы по команде ВЫЙТИ ИЗ ПОДПРОГРАММЫ восстановить содержимое счетчика команд, определяющее адрес очередной команды основной программы.

Регистр признаков результата операции или регистр состояния фиксирует состояние микропроцессора после выполнения каждой команды. Его содержимое (часто называемое флагами или флажками) используется для организации внутрипрограммных переходов по заданным признакам и условиям. Если некоторое условие выполнено, то соответствующий триггер регистра устанавливается в устойчивое состояние, соответствующее 1 или 0. В результате проверки с помощью специальных команд этого состояния в программе образуются ветвления и циклы.

Признаками результата выполненной операции являются "нуль", "отрицательный результат", "перенос" и другие. Если, например, признаком результата является "нуль" и в ходе операции получен нуль, то соответствующий триггер регистра состояния установится в единичное состояние, которое можно использовать для организации соответствующего условного перехода.

Регистр состояния является необходимым элементом структуры микропроцессора в составе микроЭВМ фон-неймановского типа.

Внутренняя шина данных микропроцессора представляет собой многопроводную шину (магистраль), связывающую АЛУ, регистры общего назначения, специальные регистры и другие узлы внутри БИС микропроцессора. Обычно внутренняя шина данных является

двунаправленной и ее разрядность совпадает с разрядностью внешней шины данных. По внутренней шине данных передаются команды и данные.

Из-за ограниченного числа внешних выводов шина данных подключается к узлам БИС микропроцессора через специальную схему мультиплексора, работающего в режиме двунаправленной передачи данных. В этом случае передача и прием данных осуществляются последовательно во времени, что снижает быстродействие микропроцессора. Внутренняя шина данных через буферный регистр соединяется с внешней шиной данных, которая используется для организации обмена данными между микропроцессором, памятью и БИС интерфейса. Внешняя шина данных чаще всего 8-разрядная.

Шина адреса служит для адресации памяти и регистров внешних устройств, входящих в БИС интерфейса. На этой шине в начале каждого цикла чтения команды, чтения или вывода данных формируется двоичный код, определяющий адрес, по которому хранятся код команды, данные или адрес регистра внешнего устройства.

Шина адреса микропроцессора чаще всего 16-разрядная. Этого достаточно для прямой адресации памяти емкостью $2^{16}=64\text{К}$ слов и регистров внешних устройств. Если длина слова равна 8 разрядам, то в этом случае емкость адресуемой памяти равна 64К байтов.

Типичным примером микропроцессора со структурой, аналогичной структуре, приведенной на рис. 2.3, является микропроцессор типа КР580ИК80. Принцип действия микропроцессора упрощенно рассмотрим на следующем примере.

Пусть требуется вычислить $x = (5 + 35) \cdot 10$. Предположим, что программа выполнения арифметических операций находится в ячейках памяти с номерами 0201–0208, а данные хранятся в ячейках памяти 0301–0304 и расположены следующим образом:

Адрес (номер ячейки памяти)	Мнемоника команды	Содержимое Н, L
0201	MOV A, M	0301
0202	INR L	0302
0203	ADD M	0302
0204	CALL MUL	0303
0205		
0206		
0207	INR L	0304
0208	MOV M, A	0304
0301	5	
0302	35	
0303	10	
0304	400	

Программа содержит следующие команды микропроцессора КР580ИК80: MOV A, M – загрузить аккумулятор из ячейки памяти

с адресом, хранящимся в регистрах общего назначения Н и L (предполагаем, что в этих регистрах записан заблаговременно код 0301); INR L — увеличить на единицу содержимое регистра L; ADD M — сложить содержимое аккумулятора и ячейки памяти с адресом, записанным в регистрах Н и L; CALL MUL — безусловный переход к подпрограмме умножения MUL; MOV M, A — записать содержимое аккумулятора в ячейку памяти. Очевидно, что команды и данные представлены в двоичной системе счисления. Для последовательного выполнения каждой команды устройство управления должно обеспечить реализацию следующих функций: считывание команды, интерпретацию кода команды, установку адреса операнда, исполнение команды, установку счетчика команд для выборки очередной команды.

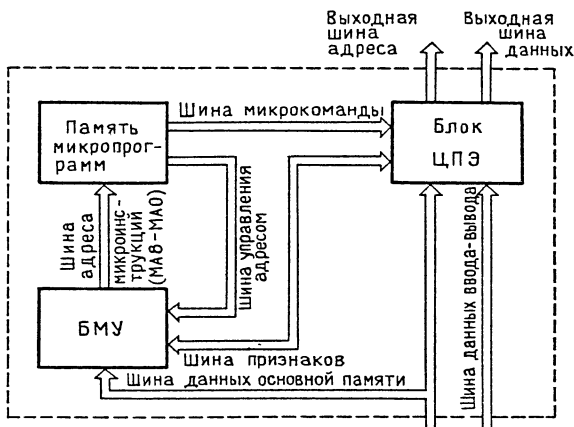
При исполнении программы прежде всего производится загрузка счетчика команд, в который заносится адрес 0201. По сигналу ПУСК устройство управления передает адрес 0201 по шине адреса микроЭВМ в память. Командное слово, хранимое по этому адресу, пересылается в дешифратор команд устройства управления. Код команды MOV A, M обеспечивает выработку соответствующих управляющих сигналов для загрузки в аккумулятор содержимого ячейки памяти с адресом 0301. После выполнения первой команды содержимое счетчика команд увеличивается на единицу, т.е. становится равным 0202, и рассмотренная последовательность операций в микропроцессоре повторяется по командному слову, хранимому в ячейке памяти с адресом 0202.

В программе используется подпрограмма MUL, обращение к которой осуществляется по команде CALL, занимающей три ячейки памяти с адресами 0204, 0205 и 0206. Последняя команда программы обеспечивает загрузку результата в ячейку памяти с адресом 0304.

Микропроцессор с микропрограммным управлением. Микропрограммное управление применяется в микропроцессорах с наращиваемой длиной разрядной сетки. Оно основано на внешнем микропрограммировании. При микропрограммном управлении набор команд может быть нефиксированным и изменяться разработчиком САУ.

Простейшие микропроцессоры с микропрограммным управлением обычно содержат несколько кристаллов БИС, включающих (рис. 2.4) блок центральных процессорных элементов (ЦПЭ), блок микропрограммного управления (БМУ) и память микропрограмм.

Блок ЦПЭ предназначен для обработки информации, поступающей из основной памяти или от устройств ввода-вывода. Он образован из некоторого количества совершенно одинаковых модулей малой разрядности (2- или 4-разрядные, но могут быть и 8-разряд-



Р и с. 2.4. Схема микропроцессора с микропрограммным принципом управления

ные модули). Эти модули выполнены в виде отдельных БИС и имеют структуру узкоформатного полного процессора. Для образования ЦПЭ заданной разрядности несколько модулей ЦПЭ объединяют общей шиной микрокоманд и соединяют цепи межразрядных переносов. Важным элементом структуры многокристального микропроцессора является память микропрограмм (управляющая память).

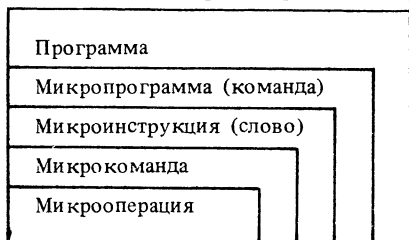
Память микропрограмм в каждой ячейке содержит слово, являющееся микроинструкцией. Формат микроинструкции зависит от структуры микропроцессора и микроЭВМ. В нем учитываются также особенности конкретного микропроцессорного комплекта БИС.

В общем случае формат микроинструкции обязательно включает разряды поля микрокоманды и поля управления адресом следующей микрокоманды. Наряду с указанными полями в формат микроинструкции могут входить поле кода управления логикой флажков, поле адресации и управления интерфейсом, стеком и др.

Обычно микроинструкция занимает несколько десятков разрядов. Ее длина определяет емкость в битах каждой ячейки памяти, которая оказывается значительной в сравнении с емкостью каждой ячейки памяти однокристалльных микропроцессоров. Например, минимальная емкость ячейки памяти микропрограмм микропроцессора на комплекте БИС серии К589 равна 21 разряду, а для микропроцессора КР580ИК80 она всегда равна 8 разрядам.

Выполнение микропроцессором с микропрограммным управлением одной микроинструкции не соответствует выполнению

команды в том смысле, в каком мы рассматривали понятие команды применительно к микропроцессору с аппаратным принципом управления. Для выполнения одной такой команды микропроцессору с микропрограммным управлением необходимо выполнить несколько микроинструкций, образующих микропрограмму:

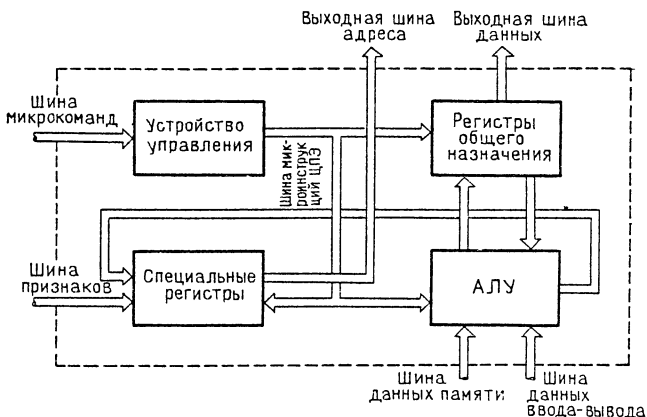


Каждая микроинструкция включает микрокоманду и микрооперации, коды которых поступают в блок ЦПЭ.

Очевидно, что микроинструкция содержит также код, определяющий адрес следующей микроинструкции, которую должен выполнить микропроцессор. Этот код подается в блок микропрограммного управления.

Блок микропрограммного управления предназначен для адресации памяти микропрограмм на основе таблицы функций управления адресом, задаваемой в БМУ аппаратными средствами. Блок микропрограммного управления позволяет наряду с безусловными организовывать условные переходы. Для этого в нем имеется регистр состояния, информация в который заносится из управляющей памяти или из блока ЦПЭ. Таким образом, для выполнения последовательности микрокоманд в микропроцессоре с микропрограммным управлением на каждом шаге микропрограммы должны быть выполнены адресация ячейки памяти микропрограмм, выборка микроинструкции, загрузка кода микрокоманды в блок ЦПЭ, а кода адреса следующей микроинструкции в БМУ.

Рассмотрим состав одного модуля центрального процессорного элемента, назначение и принцип действия его элементов. В состав ЦПЭ входят арифметико-логическое устройство, группа регистров общего назначения с разрядностью, равной ширине шины данных, группа специальных регистров, устройство управления и совокупность шин, показанных на рис. 2.5. АЛУ реализует некоторый набор операций в зависимости от кодов микрокоманд. Код микрокоманд, содержащий обычно код микрооперации и код маски, дешифрируется в устройстве управления, которое формирует совокупность сигналов, управляющих работой АЛУ и регистров и обеспечивающих выполнение микрооперации. Код маски используется для расширения списка микрокоманд.



Р и с. 2.5. Схема модуля центрального процессорного элемента

Программа работы микропроцессора хранится в основной постоянной памяти, откуда в начале цикла выполнения команды извлекается ее код и поступает в блок микропрограммного управления (рис. 2.4). БМУ преобразует этот код в адрес первой микроинструкции, которая извлекается из памяти микропрограмм и поступает на шину микрокоманд и шину управления адресом следующей микроинструкции. Код микрокоманды преобразуется в управляющие сигналы модулей ЦПЭ, в результате чего выполняется микрооперация, а код шины управления адресом следующей микроинструкции преобразуется в соответствующий адрес ячейки памяти микропрограмм, где хранится очередная микроинструкция. Так последовательно, шаг за шагом, выполняются микроинструкции, в результате чего оказывается выполненной некоторая команда.

Микропроцессор приступает к выполнению очередной команды программы. Для обеспечения этого каждая микропрограмма составляется таким образом, чтобы в результате выполнения некоторого количества последних микроинструкций на шине адреса блока ЦПЭ появился адрес основной-памяти, по которому записан код очередной команды.

Входящие в состав модулей ЦПЭ регистры общего назначения и специальные регистры (счетчик команд и буферные регистры) обычно не отличаются по своему назначению от соответствующих регистров однокристалльного микропроцессора с аппаратным управлением.

Обычно память микропрограмм выполняется в виде ПЗУ. При необходимости модификации алгоритма или набора команд ПЗУ микропрограмм может быть извлечено из системы и заменено новым.

Микропроцессоры с микропрограммным управлением в отличие от однокристальных микропроцессоров выполняются не только с разделенными входными и выходными шинами данных, но и имеют несколько разделенных входных шин. Это позволяет создавать микропроцессорные системы с высокоэффективной разветвленной организацией межмодульных связей, учитывающей особенности реализуемого алгоритма обработки данных и повышающих производительность системы.

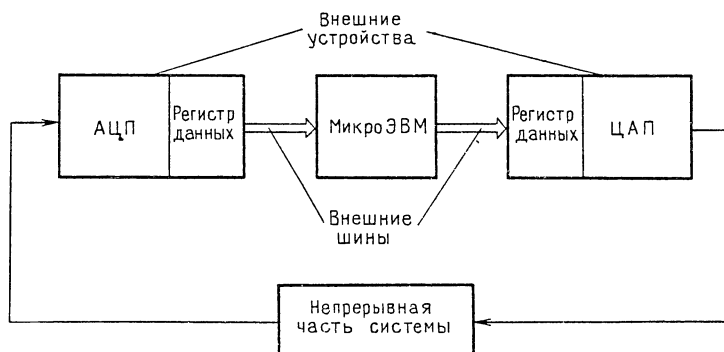
В заключение отметим, что микропрограммный принцип управления широко используется в серийных микроЭВМ (табл. 2.1).

§ 2.2. ВВОД-ВЫВОД МИКРОЭВМ В СТРУКТУРЕ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

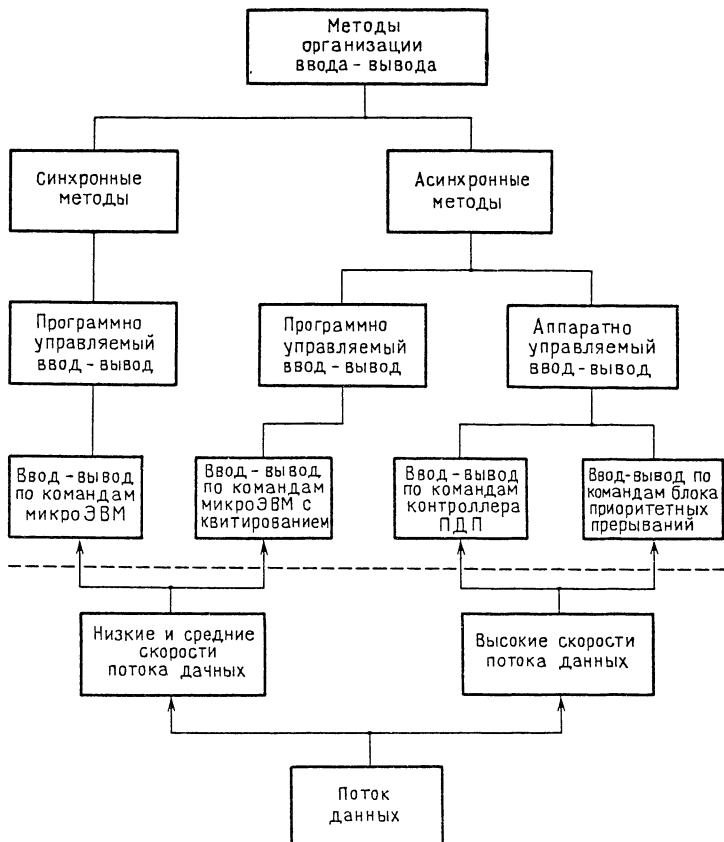
Предварительные замечания. Эффективность использования микроЭВМ зависит от ее способности взаимодействовать с непрерывной частью автоматической системы. Функции элементов сопряжения микроЭВМ с непрерывной частью автоматической системы могут выполнять аналого-цифровой и цифро-аналоговый преобразователи (АЦП и ЦАП соответственно), которые в этом случае называются внешними устройствами ВУ (рис. 2.6).

К микроЭВМ подключаются также и другие самые разнообразные внешние устройства, например, индикаторы, цифровые датчики, накопители на магнитной ленте, принтеры (печатающие устройства) и графопостроители, счетчики, исполнительные устройства и т.д.

Внешние устройства отличаются характеристиками, режимами передачи и приема данных. Они могут иметь малую скорость передачи данных (< 10 К байт/с) или, наоборот, обладать высокой скоростью передачи (порядка нескольких М байт/с. Длина слов данных также быть различна — от 1 до 32 битов. Передача данных может осуществляться последовательным или параллельным кодом и т.д.



Р и с. 2.6. Внешние устройства микроЭВМ



Р и с. 2.7. Методы организации ввода-вывода

Для организации обмена данными между внешними устройствами и микропроцессором и памятью микроЭВМ используется интерфейс ввода-вывода, являющийся составной частью микроЭВМ.

Интерфейсом ввода-вывода называется стандартное сопряжение устройств управления внешними устройствами и каналов ввода-вывода.

В понятие интерфейса ввода-вывода входят совокупность интерфейсных схем, шин и правил обмена данными.

Интерфейс ввода-вывода, являющийся буфером между системной шиной микроЭВМ и внешними устройствами, обеспечивает возможность проектирования внешних устройств независимо от структуры шины, с которой они будут применяться. Для подклю-

чения одного и того же внешнего устройства к двум шинам с различной структурой требуется изменить только интерфейс.

Интерфейс ввода-вывода зависит от характеристик потока данных внешних устройств, структуры системной шины, режима передачи данных (последовательного или параллельного) и метода организации ввода-вывода. В общем случае интерфейс может быть спроектирован с использованием синхронного и асинхронного методов организации ввода-вывода (рис. 2.7).

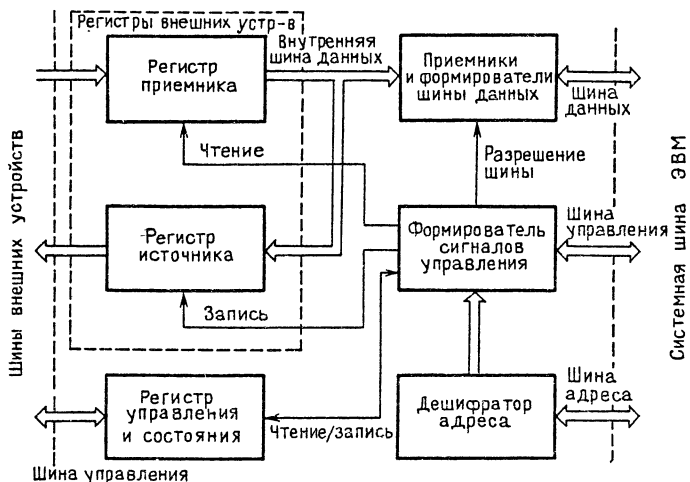
При синхронном обмене данными информация поступает непрерывной последовательностью синхронно с задающей частотой, сопровождающей данные, или в определенной временной последовательности. С начала обмена данными внешнее устройство должно обязательно обслуживаться процессором, иначе информация будет потеряна.

В тех случаях, когда скорость передачи данных низкая, применяют программно управляемый обмен. Если требуемая скорость передачи пакета данных высока, то тогда необходимо использовать аппаратно управляемые методы организации ввода-вывода, в частности, прямой доступ к памяти (ПДП). Синхронная передача данных обычно производится при работе с электромеханическими и другими низко- и среднескоростными внешними устройствами.

Асинхронные методы организации ввода-вывода связаны с прекращением выполнения микроЭВМ текущей или невыполнением очередной команды рабочей программы.

При программно управляемом асинхронном обмене микроЭВМ может находиться в режиме ожидания, в который она попадает по командам ВВОД или ВЫВОД рабочей программы. Это состояние продолжается до поступления сигнала ГОТОВНОСТЬ от внешних устройств. Такой способ организации взаимодействия микроЭВМ с внешними устройствами, основанный на взаимном обмене информацией о состоянии (статусе), называют квитированием. Квитированием достигается синхронизация различных по быстродействию микропроцессорных средств и внешних устройств. Однако ожидание момента готовности внешних устройств может приводить к потерям машинного времени.

От этого недостатка свободен асинхронный ввод-вывод с прерыванием программы по запросу внешнего устройства. В этом случае при поступлении сигнала ЗАПРОС ПРЕРЫВАНИЯ микроЭВМ после выполнения текущей команды выдает сигнал РАЗРЕШЕНИЕ ПРЕРЫВАНИЯ и далее реализует подпрограмму обслуживания прерывания, связанную, например, с вводом или выводом данных. Дальнейшее повышение скорости обмена данными между внешними устройствами и памятью микропроцессорных средств достигается использованием режима прямого доступа к памяти.



Р и с. 2.8. Типовая структура БИС интерфейса

Известны различные методы организации прямого доступа к памяти, простейший и наиболее распространенный из которых приводит микропроцессор или микроЭВМ в состояние ОСТАНОВ, что не всегда допустимо. Поэтому разработаны также другие методы, среди которых можно отметить прямой доступ с пропуском цикла и прямой доступ к памяти с параллельной работой микропроцессора [128].

Типовая схема БИС интерфейса. Структурная схема БИС интерфейса показана на рис. 2.8. В интерфейсе регистры данных соединены шиной внешних устройств с соответствующим внешним устройством, являющимся источником или приемником данных. Эти регистры служат для временного хранения передаваемых данных. Их называют регистром-приемником и регистром-источником. В интерфейсе, служащем для ввода и вывода данных, регистр-приемник и регистр-источник имеют общий адрес. Этот адрес поступает по системной шине от микропроцессора в дешифратор адреса, который совместно с формирователем сигналов управления выбирает соответствующий регистр данных.

Регистр управления и состояния хранит информацию, поступающую от микропроцессора по шине управления к внешнему устройству, или хранит текущее состояние внешнего устройства (например, готовность вводимых данных, обнаруженные ошибки и т.д.).

При выполнении операции ввода данные принимаются от внешнего устройства и запоминаются в регистре-приемнике. Затем проверяется слово состояния в регистре управления и состояния,

и после этого осуществляется программный ввод данных в микропроцессор. Вывод данных выполняется аналогично, но в обратной последовательности.

В интерфейс может быть введена логика генерирования прерывания по готовности данных. Тогда будет реализован асинхронный ввод данных с прерыванием.

Передача данных между микропроцессором и интерфейсом ввода-вывода в синхронном режиме опирается на управляющие сигналы ЧТЕНИЕ ВУ, ЗАПИСЬ ВУ и ПОДТВЕРЖДЕНИЕ. Микропроцессор совместно со специальными логическими схемами управления системной шиной выдает запрос ЧТЕНИЕ ВУ или ЗАПИСЬ ВУ, а БИС интерфейса реагирует подтверждением, что данные приняты с шины данных ВУ в регистр-приемник или находятся в регистре-источнике и помещены на шину данных ВУ.

При генерировании сигнала ЧТЕНИЕ ВУ дешифратор адреса и формирователь сигналов управления сигналом ЧТЕНИЕ выбирают регистр управления и состояния и регистр-приемник. Формирователь сигналов управления выдает также сигналы РАЗРЕШЕНИЕ ШИНЫ и ПОДТВЕРЖДЕНИЕ. В результате данные из регистра-приемника поступают на шину данных системной шины микроЭВМ и вводятся в микропроцессор или память.

Сигнал ЗАПИСЬ ВУ означает вывод данных. По этому сигналу формирователь сигналов управления вырабатывает сигнал ЗАПИСЬ, разрешающий ввод данных с системной шины микроЭВМ в регистр-источник и управляющий регистром управления и состояния.

Формирователь сигналов управления генерирует также с некоторой задержкой сигнал ПОДТВЕРЖДЕНИЕ. Когда микропроцессор принимает сигнал подтверждения, он сбрасывает сигнал ЧТЕНИЕ ВУ или ЗАПИСЬ ВУ и снимает с шины адреса адрес ВУ и/или данные с шины данных. Это приводит к снятию схемой интерфейса сигнала ПОДТВЕРЖДЕНИЕ. На этом цикл обмена данными завершается.

Синхронный обмен данными. Программно управляемый обмен основан на том, что системы команд многих микроЭВМ включают специальные команды ввода и вывода. Так, например, микроЭВМ на микропроцессоре КР580ИК80 имеет команды IN (ВВЕСТИ) и OUT (ВЫВЕСТИ), которые обеспечивают ввод данных с шины данных в аккумулятор и соответственно вывод данных из аккумулятора на шину данных. В микроЭВМ семейства LSI-11 эта же задача решается одной командой MOV (ПЕРЕСЛАТЬ). Соответствующей записью операндов решается задача адресации регистра внешнего устройства и регистра общего назначения микропроцессора, выступающих в команде в роли источника (обозначается src) и приемника (dst) данных.

В микроЭВМ семейства LSI-11 устройства ввода-вывода выбираются адресацией регистров внешних устройств, к которым микроЭВМ обращается как к ячейкам памяти. Это несколько удлиняет операции обмена, но техническая реализация оказывается более экономичной: в этом случае дешифрация адреса регистра внешнего устройства производится уже имеющимся дешифратором адреса запоминающего устройства, а не специальным блоком.

Организация программно-управляемого синхронного обмена данными целиком ложится на программиста. Он должен учитывать, что при программировании могут возникнуть трудности из-за разницы в быстродействии микроЭВМ и внешних устройств. Например, при выполнении операций вывода данных может случиться, что микроЭВМ уже начала выдачу данных, а в это время внешнее устройство еще занято обработкой предыдущей порции информации. Во время операции ввода может возникнуть ситуация, когда микроЭВМ несколько раз подряд считывает и обрабатывает одни и те же данные из-за того, что внешнее устройство еще не успело подготовить очередную порцию данных и т.д.

Рассмотрим, например, организацию синхронного обмена данными между микроЭВМ семейства LSI-11 и внешними устройствами.

Пусть микроЭВМ должна собирать данные о результатах измерений, выполненных тремя внешними устройствами. Необходимо спроектировать интерфейс ввода-вывода и написать программу для передачи данных. Обратимся к рис. 2.8. Предположим, что каждое внешнее устройство подключено к своему регистру-приемнику.

Данные из внешних устройств должны передаваться в память микроЭВМ и запоминаться в трех последовательных ячейках. Так как в данном случае можно применить простейший интерфейс, то будем полагать, что постоянно имеется сигнал ЧТЕНИЕ на входе регистра управления и состояния.

Каждое внешнее устройство должно иметь свой адрес. Предположим, что эти адреса READ1=160000, READ2=160004, READ3=160008. МикроЭВМ должна послать три команды ввода с указанными адресами. Допустим далее, что адреса ячеек памяти соответственно DEV1=020000, DEV2=020004, DEV3=020008.

Команда ввода реализуется с использованием регистров общего назначения, которым присвоим следующие имена: READ1=%0; READ2=%1; READ3=%2; DEV1=%3; DEV2=%4; DEV3=%5. Здесь знаки % указывают на операцию присвоения регистрам 0, 1, 2, ... соответствующих имен.

Команда ввода для данного примера имеет вид MOV @(READ), @(DEV). Перед обращением к командам ввода следует во все шесть регистров общего назначения занести соответствующие адреса внешних устройств и ячеек памяти.

Т а б л и ц а 2.2

Команда	Комментарий
MOV # 160000, READ1	Запись адреса в РОИ
MOV # 160004, READ2	То же
MOV # 160002, READ3	То же
MOV # 020000, DEV1	То же
MOV # 020004, DEV2	То же
MOV # 020002, DEV3	То же
MOV @ (READ1), @ (DEV1)	Ввод данных из ВУ в память.
MOV @ (READ2), @ (DEV2)	То же
MOV @ (READ3), @ (DEV3)	То же

Программа ввода данных в память микроЭВМ в этом случае будет иметь вид, показанный в табл. 2.2.

Программа состоит из двух частей, Первые шесть команд обеспечивают запись в регистры общего назначения адресов внешних устройств и ячеек памяти. Последние три команды реализуют пересылку данных из внешних устройств в ячейки памяти.

Ввод данных осуществляется следующим образом. МикроЭВМ формирует при выполнении каждой из последних трех команд адрес соответствующего внешнего устройства, выдаваемый на шину адреса, и сигнал ЧТЕНИЕ ВУ, поступающий на шину управления. Дешифратор адреса и формирователь сигналов управления интерфейса соответствующего внешнего устройства выбирают регистр-приемник, откуда данные через приемники и формирователи шины данных поступают по системной шине в заданные ячейки памяти. Очевидно, что в рассматриваемом случае следует применить три идентичных схемы интерфейса (по числу внешних устройств), отличающихся только настройкой дешифратора адреса.

Асинхронный программно управляемый обмен данными (ввод-вывод с квити́рованием). Надежная синхронизация работы внешних устройств и микроЭВМ при вводе-выводе данных может быть достигнута с помощью тестовых проверок.

При программно управляемом асинхронном обмене ввод и вывод данных также осуществляются по командам ВВЕСТИ и ВЫВЕСТИ. Однако микроЭВМ в асинхронном режиме в отличие от синхронного прекращает выполнение команды и находится в режиме ожидания готовности данных внешних устройств до тех пор, пока не поступит от внешних устройств сигнал ГОТОВНОСТЬ ВУ, преобразуемый БИС интерфейса в сигнал ПОДТВЕРЖДЕНИЕ. При обмене информацией с большинством внешних устройств вопросы и ответы микроЭВМ и внешних устройств даются в следующем порядке.

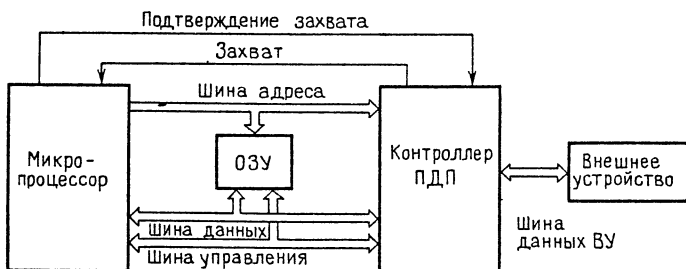
Микропроцессор и специальные логические схемы микроЭВМ формируют, например, по команде ВВЕСТИ сигнал запроса ЧТЕНИЕ ВУ, поступающий на БИС интерфейса. МикроЭВМ тем самым указывает на готовность к приему следующей порции данных. По сигналу ГОТОВНОСТЬ ВУ (часто этот сигнал называют СТРОБОМ), поступающему от внешнего устройства, интерфейс вырабатывает сигнал ПОДТВЕРЖДЕНИЕ. Данные заносятся в микропроцессор.

На рис. 2.9 изображена структурная схема алгоритма, в соответствии с которой микроЭВМ реализует описанный процесс ввода данных с применением квитирования. Блок 1 соответствует запросу сигналом ЧТЕНИЕ ВУ данных для ввода в процессор микроЭВМ. Блоки 2 и 3 образуют цикл, который повторяется до тех пор, пока не будет установлен сигнал ПОДТВЕРЖДЕНИЕ. В течение этого времени микроЭВМ находится в режиме ожидания. При поступлении сигнала ПОДТВЕРЖДЕНИЕ цикл ожидания прекращается, и выполняется ввод данных в процессор микроЭВМ (для микроЭВМ LSI-11 возможен ввод в память или в другой регистр внешнего устройства). После завершения операций блока 4 микроЭВМ снимает сигнал запроса ЧТЕНИЕ ВУ (блок 5) и ждет сброса сигнала ПОДТВЕРЖДЕНИЕ (блок 6). Организуется новый цикл ожидания того момента времени, когда сигналы ГОТОВНОСТЬ ВУ и, следовательно, ПОДТВЕРЖДЕНИЕ будут сняты. Когда этот момент наступает, микроЭВМ переходит к выполнению очередной команды рабочей программы. БИС интерфейса в этом режиме может быть организована в соответствии со схемой, показанной на рис. 2.8.

В заключение отметим, что асинхронный программируемый ввод-вывод позволяет надежно синхронизировать работу микроЭВМ и внешних устройств. Однако рассмотрение этого режима пока-



Рис. 2.9. Схема алгоритма ввода данных с квитированием



Р и с. 2.10. Схема включения контроллера прямого доступа к памяти

зывает, что в процессе обмена действия микроЭВМ сводятся к реализации только операций ввода или вывода, при выполнении которых микроЭВМ ожидает сигналов готовности от внешних устройств, не выполняя никакой другой полезной работы.

Такое ограничение функций микроЭВМ выполнением только операций ввода или вывода заметно снижает производительность всей системы.

Решение проблемы состоит в использовании методов обмена информацией, основанных на режиме прямого доступа к памяти и на использовании сигналов прерываний от внешних устройств.

Асинхронный аппаратно управляемый ввод-вывод. Режим прямого доступа к памяти. Часто возникает необходимость обмена большими массивами данных — пакетами данных между внешними устройствами и микроЭВМ. Принципиально эта задача может быть решена с использованием асинхронного программно управляемого обмена данными. Однако в этом случае скорость обмена данными будет невысока из-за потерь времени, связанных с многократным выполнением команд ввода-вывода.

Эти непроизводительные потери исключают, вводя в состав микроЭВМ специальное управляющее устройство — контроллер прямого доступа к памяти. Контроллер ПДП обеспечивает асинхронный аппаратно управляемый обмен данными с квитированием между памятью микроЭВМ и внешними устройствами. В течение всего промежутка времени, пока реализуется режим ПДП, микропроцессор может выполнять рабочую программу, не связанную с обращением к системной шине. На это время он отключается от шин с помощью буферных усилителей с тремя состояниями. Однако в простейшем случае при реализации ПДП микропроцессор находится в состоянии ОСТАНОВ.

На время обмена данными контроллер ПДП подключается непосредственно к шинам микроЭВМ. В результате этого внешние устройства получают возможность прямого доступа к памяти.

Адреса, по которым помещается массив данных, формируются контроллером ПДП на основе информации, полученной от микроЭВМ.

Режим ПДП реализуется следующим образом (рис. 2.10). В начале обмена данными микроЭВМ по шинам данных и управления передает контроллеру ПДП следующую информацию: число слов, передаваемых при обмене; адрес первого слова в памяти; управляющее слово, указывающее на вид обмена данными — чтение или запись. После получения контроллером этой информации он посылает в микропроцессор запрос доступа к системной шине, называемый сигналом ЗАХВАТ. В ответ микропроцессор после завершения текущей команды переводит буферные усилители в высокоимпедансное состояние и посылает контроллеру ПДП сигнал ПОДТВЕРЖДЕНИЕ ЗАХВАТА. После этого начинается передача данных от ВУ в память. По окончании передачи всего пакета данных контроллер ПДП снимает сигнал ЗАХВАТ со входа процессора, который снова подключается к системной шине.

Подчеркнем, что прямой доступ к памяти позволяет получить наибольшую скорость обмена данными, но требует в сравнении с программно управляемым обменом более сложного интерфейса, в состав которого включается контроллер ПДП. Вся логика управления в режиме ПДП содержится обычно в контроллере, а БИС интерфейса ввода-вывода служит только для буферирования данных.

Контроллер включает следующие элементы.

1. Регистр управления и состояния, который хранит команду процессора на разрешение режима ПДП, коды типа передачи (чтение или запись), состояния прерываний (разрешены или запрещены) и состояния контроллера.

2. Счетчик слов, в который до начала режима ПДП загружается число передаваемых слов. После каждого цикла передачи содержимое счетчика уменьшается на единицу. Передача прекращается, когда в счетчике остается нуль.

3. Счетчик-указатель адреса. В этот счетчик до начала передачи данных заносится адрес ячейки памяти, начиная с которой будет помещен пакет данных. После передачи каждого слова данных производится увеличение содержимого счетчика для формирования следующего адреса ячейки памяти.

Проектирование контроллеров ПДП является довольно сложной задачей. Но промышленностью выпускаются БИС контроллеров ПДП, которые упрощают реализацию режима передачи блоков данных [163].

Обмен по прерыванию. Система прерывания позволяет с помощью определенного сигнала внешнего устройства прервать выполняемую программу и вызвать для исполнения программу с

более высоким приоритетом. После окончания программы с более высоким приоритетом микроЭВМ автоматически возвращается к прерванной программе.

Прерывания при вводе-выводе обычно вызваны следующими условиями: а) готовность внешних устройств по вводу или выводу данных; б) появление ошибок при вводе-выводе; в) возникновение особого условия во внешних устройствах, например аварийной ситуации. Возможны также другие причины возникновения прерываний [157].

При наличии режима прерывания микроЭВМ не ожидает готовности внешних устройств к обмену данными, а выполняет основную программу до тех пор, пока от внешнего устройства не поступит запрос на прерывание. В результате повышается производительность микроЭВМ.

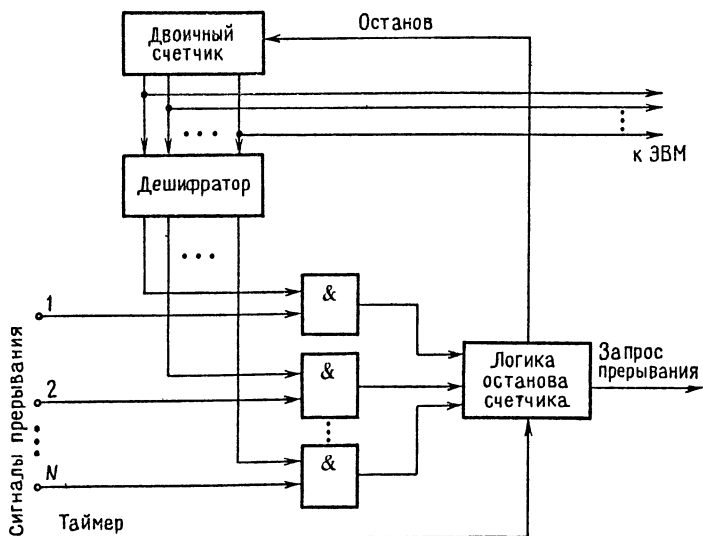
При переходе к обработке прерываний микропроцессор обычно записывает в стековую память содержимое счетчика команд и регистров общего назначения. Это делается для того, чтобы после выполнения подпрограммы обслуживания прерываний микроЭВМ смогла продолжить выполнение основной программы.

Прерывания подразделяют на одноуровневые и многоуровневые. При одноуровневом прерывании обслуживание запроса на новое прерывание возможно только после завершения обслуживания выполняемого прерывания. При многоуровневом прерывании возможно прерывание подпрограммы, обслуживающей ранее поступивший запрос на прерывание. В этом случае запросы на прерывание должны отличаться приоритетом. Тогда поступление запроса с большим приоритетом приводит к обслуживанию этого прерывания.

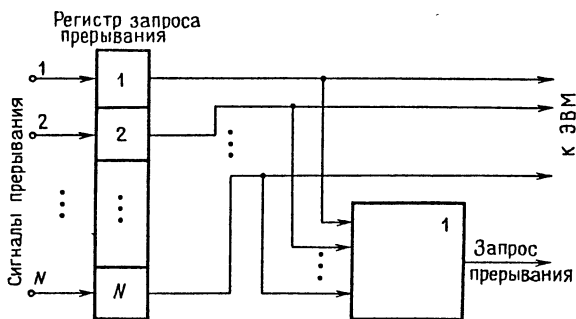
Системы прерывания реализуют с использованием методов циклического опроса, регистра прерывания, запоминания состояний и векторного прерывания [197].

В системе с циклическим опросом сигналы прерывания поступают на двухвходовые логические схемы И, на вторые входы которых последовательно заводятся сигналы с дешифратора состояния двоичного счетчика (рис. 2.11). Этим проверяется наличие сигналов прерывания. Работа счетчика прекращается по сигналу ОСТАНОВ, формируемому логикой останова счетчика. Двоичный параллельный код, снимаемый с выходов счетчика, несет в состоянии останова счетчика информацию о номере канала, по которому поступил запрос прерывания. Данная система прерывания является одноуровневой и имеет большое время реакции на запрос прерывания, так как в ней осуществляется последовательный опрос линий запроса прерываний.

Система с использованием регистра прерывания также является одноуровневой. В этой системе отдельные сигналы прерывания



Р и с. 2.11. Система прерывания с циклическим опросом



Р и с. 2.12. Система прерывания с регистром прерывания

подводятся к соответствующим разрядам параллельного регистра запроса прерывания (рис. 2.12). Далее они все объединяются логическим элементом ИЛИ, формирующим сигнал ЗАПРОС ПРЕРЫВАНИЯ. По этому сигналу подпрограмма прерывания проверяет каждый разряд регистра прерывания, выбирая активную линию с наивысшим приоритетом и вызывая соответствующую подпрограмму обслуживания прерывания.

К регистру запроса прерывания можно обращаться как к порту ввода-вывода. Тогда алгоритм, реализованный в схеме на рис. 2.12, может быть выполнен в форме подпрограммы микро-

ЭВМ, реализующей последовательный опрос (полинг) состояний разрядов регистра запроса прерывания.

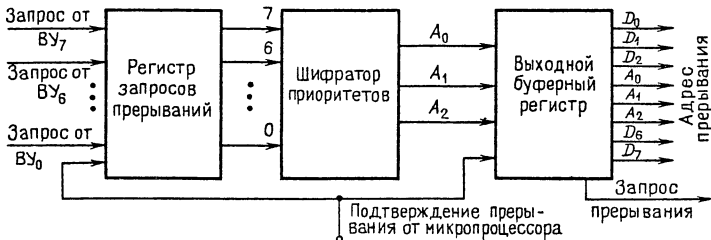
Принципиально в этой схеме возможна организация многоуровневого прерывания. Для этого подпрограмма обслуживания прерывания должна при сохранении или повторном появлении активного состояния линии ЗАПРОС ПРЕРЫВАНИЯ регулярно или повторно обращаться к подпрограмме опроса каждого разряда регистра прерывания для выявления запроса с более высоким приоритетом [53]. Такое решение задачи организации многоуровневых прерываний является, однако, малоэффективным.

Чаще используется векторное прерывание, при котором сокращается время реакции на запрос прерывания. Эта система прерывания является многоуровневой. Векторное прерывание вызывает подпрограмму, соответствующую тому внешнему устройству, которое запросило прерывание. Адрес подпрограммы может быть постоянно размещен в блоке прерываний, в микропроцессоре или поступать из внешнего устройства.

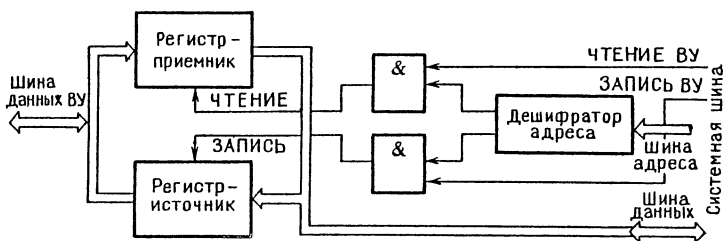
Схема управления векторными 8-уровневыми приоритетными прерываниями приведена на рис. 2.13.

Работает схема следующим образом. Каждое из восьми внешних устройств может в любой момент времени запросить прерывание, формируя сигнал на своей линии запроса. Запрос запоминается в регистре запросов прерывания. Запрос от внешнего устройства, подключенного к седьмой линии, имеет наивысший приоритет. Шифратор приоритетов выбирает запрос с наибольшим приоритетом (если поступило более одного запроса) и преобразует его в 3-битный вектор прерывания $A_2 A_1 A_0$, который запоминается в выходном буферном регистре.

Когда микропроцессор воспринимает сигнал ЗАПРОС ПРЕРЫВАНИЯ, его логика управления шиной генерирует сигнал ПОДТВЕРЖДЕНИЕ ПРЕРЫВАНИЯ, указывающий на то, что микропроцессор готов приступить к обслуживанию прерывания. Тогда содержимое выходного буферного регистра поступает на линию данных, и микропроцессор начинает обслуживать прерывание с



Р и с. 2.13. Схема организации 8-уровневых приоритетных прерываний



Р и с. 2.14. Простой интерфейс для синхронного ввода-вывода

ячейки памяти 0000 0000 00A₂ A₁ A₀ 000₍₂₎. Схема приоритетных 8-уровневых прерываний может быть организована на БИС К589ИК14 и К589ИР12 [16].

Организация схем ввода-вывода. Синхронный программно управляемый ввод можно осуществить с помощью простого интерфейса, показанного на рис. 2.14. Линии данных внешнего устройства присоединены к регистру-приемнику и регистру-источнику. Регистры имеют на выходе схемы с тремя состояниями.

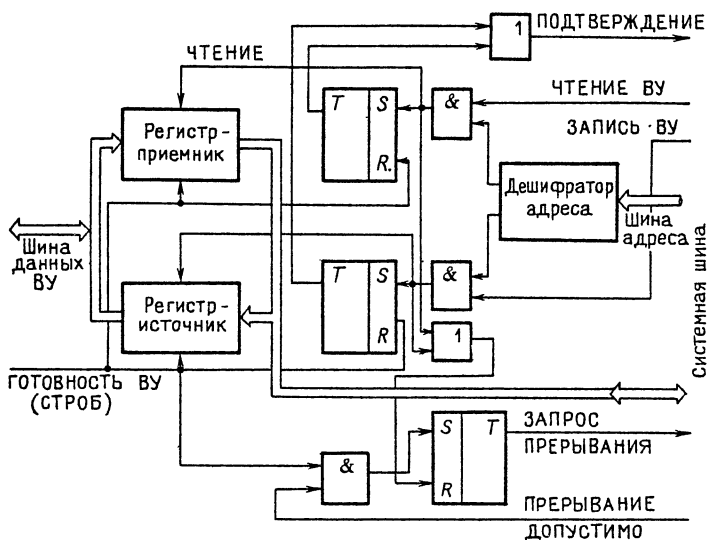
Если адрес на адресной шине, формируемый микропроцессором по командам ВВЕСТИ или ВЫВЕСТИ, соответствует адресу интерфейса, к которому подключено внешнее устройство, то выход дешифратора адреса активен. Если одновременно имеется управляющий сигнал на линии ЧТЕНИЕ ВУ, то на выходе схемы совпадения формируется сигнал ЧТЕНИЕ, и данные из регистра-приемника коммутируются на линии данных системной шины.

При выводе регистр-источник заполняется данными с системной шины в тот момент, когда одновременно активны выход дешифратора адреса и линия ЗАПИСЬ ВУ.

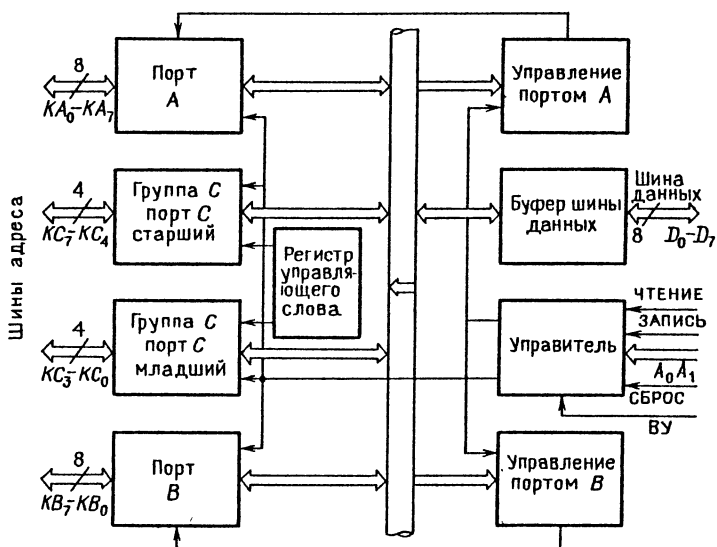
Приведенный интерфейс синхронного ввода-вывода может быть легко реализован на базе стандартных интегральных микросхем с низкой и средней степенями интеграции. Он имеется, как будет показано далее, в программируемых интерфейсах.

Асинхронный программно управляемый интерфейс может быть организован по схеме, показанной на рис. 2.15. Внешнее устройство подключено к двунаправленной шине данных ВУ и взаимодействует с интерфейсом с использованием сигнала ГОТОВНОСТЬ ВУ. При вводе входной стробирующий сигнал ГОТОВНОСТЬ ВУ обеспечивает запись входных данных в регистр-приемник, имеющий на выходе схему с тремя состояниями.

После возбуждения процессором линии ЧТЕНИЕ ВУ и выдачи адреса внешнего устройства триггер *T* переводится в единичное состояние, обеспечивая формирование сигнала ПОДТВЕРЖДЕНИЕ. Одновременно формируется сигнал ЧТЕНИЕ, под действием



Р и с. 2.15. Интерфейс асинхронного программно-управляемого обмена и обмена по прерыванию



Р и с. 2.16. Структура БИС программируемого интерфейса

которого данные из регистра-приемника коммутируются на линии данных системной шины.

При выводе данных процессор формирует сигнал ЗАПИСЬ ВУ и обеспечивает адресацию интерфейса. Данные заносятся в регистр-источник и хранятся в нем до поступления сигнала ГОТОВНОСТЬ ВУ, которым в данном случае внешнее устройство сообщает о своей готовности к приему данных. При поступлении этого сигнала данные из регистра-источника коммутируются на шину данных ВУ, а интерфейс формирует сигнал подтверждения приема данных. Схема, показанная на рис. 2.15, может быть использована в режиме прерывания. Если сигнал микропроцессора ПРЕРЫВАНИЕ ДОПУСТИМО активен в течение времени действия сигнала ГОТОВНОСТЬ ВУ, то триггер запроса прерывания устанавливается в единичное состояние, формируя сигнал ЗАПРОС ПРЕРЫВАНИЯ. Триггер запроса прерывания возвращается в нулевое состояние в момент передачи входных данных на линию данных системной шины или шины данных ВУ.

Интерфейс часто выполняется на базе серийных БИС. Программируемый интерфейс (рис. 2.16) представляет собой универсальную схему ввода-вывода. Он имеет 24 линии ввода-вывода, которые могут работать в режимах 0, 1, 2. Исходным является режим высокоимпедансного состояния всех 24 информационных линий. В этот режим программируемый интерфейс вводится под действием системного сигнала СБРОС.

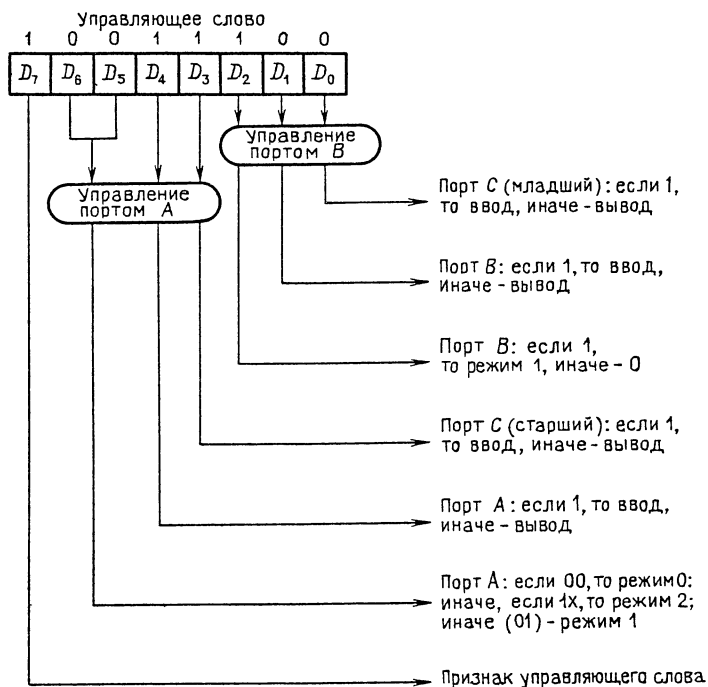
Первый режим является базовым. Он обозначается цифрой 0. В этом режиме все 24 линии ввода-вывода работают как три 8-разрядных порта (канала): *A*, *B* и *C*. Каждый порт может быть либо портом ввода, либо портом вывода. Выбор конкретной БИС интерфейса и вывод ее буферов из высокоимпедансного состояния производятся подачей на эту схему сигналов выбора кристалла ВУ, ЧТЕНИЕ или ЗАПИСЬ и управляющего слова. Управляющее слово выводится из микропроцессора по команде ВЫВЕСТИ по адресу порта *C* выбранной БИС интерфейса. Для этого используется шина адреса A_0A_1 , на которой устанавливается код 11.

В общем случае направление передачи информации в зависимости от состояний шины A_0A_1 и линий сигналов ВУ, ЧТЕНИЕ и ЗАПИСЬ определяется по правилам, показанным в табл. 2.3.

Режим 0 соответствует синхронному программно управляемому обмену данными. Он осуществляется по инициативе микропроцессора выдачей на шину данных необходимого управляющего слова, которое заносится в регистр управляющего слова (РУС), если $D7 = 1$. Формат управляющего слова, задающего режим работы микросхемы КР580ИК55, показан на рис. 2.17. Так, например,

Т а б л и ц а 2.3

Направление передачи информации	A_0	A_1	ВУ	ЗАПИСЬ	ЧТЕНИЕ
1. С шины данных					
- в регистр управляющего слова	1	1	0	0	1
- на шину порта <i>A</i>	0	0	0	0	1
- на шину порта <i>B</i>	1	0	0	0	1
- на шину порта <i>C</i>	0	1	0	0	1
2. На шину данных:					
- с шины порта <i>A</i>	0	0	0	1	0
- с шины порта <i>B</i>	1	0	0	1	0
- с шины порта <i>C</i>	0	1	0	1	0
3. Высокоимпедансное состояние	любой	любой	1	любой	любой



Р и с. 2.17. Формат управляющего слова

управляющее слово 10011100 реализует следующий режим работы БИС интерфейса:

Порт *A* – режим 0, ввод; порт *B* – режим 1, вывод; порт *C* (старший) – режим 0, ввод; порт *C* (младший) – режим 1, вывод.

Принцип действия интегральной схемы интерфейса в этом режиме рассмотрим на примере организации управления 8-разрядным индикатором.

Пусть в каждый разряд индикатора последовательно вводится 4-разрядное число, преобразуемое в символ шестнадцатеричной системы счисления. Тогда управление индикатором можно осуществить по программе, структурная схема которой показана на рис. 2.18. Работает программа следующим образом.

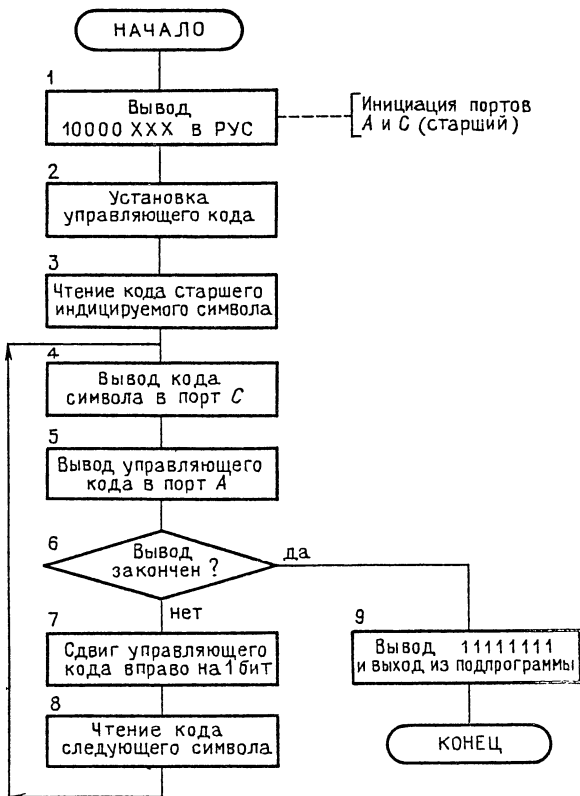
По команде ВЫВЕСТИ (OUT) кодом 1000XXX БИС интерфейса переводится в режим 0. При этом порты *A* и *C* (старшие разряды) работают на вывод данных.

Подпрограмма индикации содержит блоки, обозначенные на рис. 2.18 цифрами 2 – 9. Блок 2 соответствует установке управляющего кода 01111111. В старшем разряде этого кода записан 0. Это соответствует низкому уровню сигнала на входе РАЗРЕШЕНИЕ ИНДИКАТОРА, который управляет регистром-зашелкой. При низком уровне сигнала разрешается запись выводимого из микропроцессора кода в регистр-зашелку; при высоком уровне сигнала на этом входе разрешение на запись кода в регистр-зашелку не выдается.

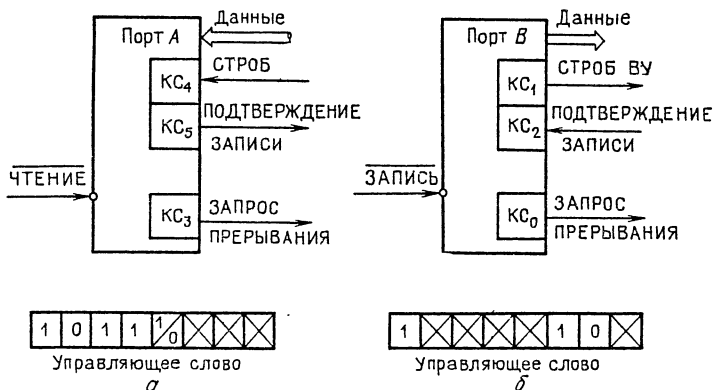
Далее из памяти извлекается код старшего индицируемого символа шестнадцатеричной системы счисления (блок 3) и двумя командами ВЫВЕСТИ осуществляется вывод кода символа в порт *C*, а также управляющего кода 01111111 в порт *A* (блоки 4 и 5). Если выведенная цифра не является восьмой, то по команде микропроцессора осуществляется сдвиг управляющего кода вправо на один разряд и считывается код следующего символа (блоки 7 и 8). В подпрограмме используется циклический сдвиг управляющего кода. Следовательно, разрешенным оказывается регистр-зашелка следующего разряда индикатора, в котором индицируется код следующего символа.

По завершении вывода в порт *C* восьми символов шестнадцатеричной системы счисления осуществляется вывод в порт *A* управляющего кода 11111111 и реализуется выход из подпрограммы индикации.

Рассмотрим далее режим 1. Он соответствует стробируемому вводу-выводу (вводу-выводу с квитинованием). Управляющие слова для этого режима показаны на рис. 2.19, *a* и *б*. Управляющее слово, показанное на рис. 2.19, *a*, переводит порт *A* в состояние ввода данных от внешнего устройства к микропроцессору.



Р и с. 2.18. Структурная схема подпрограммы индикации



Р и с. 2.19. Сигналы управления БИС интерфейса в режиме 1

Сигнал СТРОБ, подаваемый от внешнего устройства на один из входов порта *C* (например, на вход КС4 для порта *A*) инициирует ввод данных в буферный регистр порта *A*. Одновременно появляется сигнал ПОДТВЕРЖДЕНИЕ ЗАПИСИ (выход КС5), подаваемый во внешнее устройство. В результате сигнал СТРОБ сбрасывается.

Сигнал СТРОБ задним фронтом при единичном состоянии триггера разрешения прерывания инициирует появление на выходе порта сигнала ЗАПРОС ПРЕРЫВАНИЯ, который поступает в микропроцессор. Через отрезок времени, необходимый для обслуживания прерывания, данные из буферного регистра порта *A* переписываются по сигналу ЧТЕНИЕ в процессор. Сигнал ЗАПРОС ПРЕРЫВАНИЯ сбрасывается передним фронтом сигнала ЧТЕНИЕ. Задним фронтом сигнала ЧТЕНИЕ сбрасывается сигнал ПОДТВЕРЖДЕНИЕ ЗАПИСИ, что свидетельствует о возможности ввода нового слова.

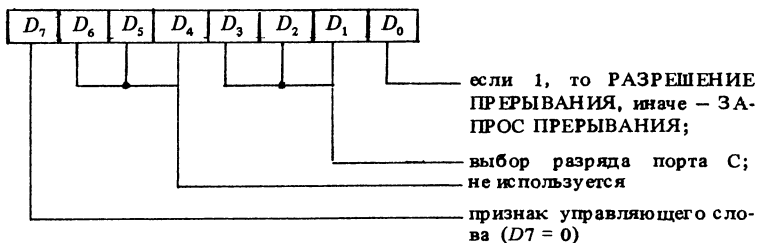
На рис. 2.19,б приведена схема вывода данных через порт *B*. Данные переписываются в буферный регистр порта *B* с шины данных микропроцессора по сигналу ЗАПИСЬ. Этот сигнал инициирует появление сигнала СТРОБ ВУ, которым внешнее устройство извещается о том, что микропроцессор заполнил данными выходной буфер. В момент времени, когда внешнее устройство прочитало данные с буфера порта *B*, оно генерирует сигнал ПОДТВЕРЖДЕНИЕ ЗАПИСИ. По заднему фронту сигнала ПОДТВЕРЖДЕНИЕ ЗАПИСИ происходит сброс сигнала СТРОБ ВУ, что может быть использовано системой прерывания для извещения о том, что внешнее устройство готово к приему нового слова данных.

В режиме 2 обмен данными происходит по сигналам, аналогичным управляющим сигналам режима 1. Отличие заключается в том, что режим 2 осуществляется одной двунаправленной 8-разрядной шиной порта *A* и программируются два триггера разрешения прерываний, связанных соответственно с операциями ввода и вывода [3].

Режимы 1 и 2 работы программируемого интерфейса могут быть отнесены, как следует из изложенного, к асинхронным методам организации ввода-вывода. Обмен данными управляется программно-аппаратными средствами.

Порт *C* может быть использован наряду с указанными возможностями также в качестве буферного регистра. Любой из его восьми разрядов может быть установлен в 1 или 0 при записи в разряде *D7* нуля. Любой из разрядов порта *C* может быть выбран также в качестве триггера запроса прерывания или триггера разрешения прерывания. Программный доступ к ним позволяет программисту разработать различные процедуры обработки прерываний, приспособив их к структуре системы. Управляющее слово

для этого случая имеет вид:



В заключение отметим, что многие интерфейсы ввода-вывода, реализованные в виде отдельной БИС, ориентированы на определенный тип внешних устройств. Примером может служить БИС последовательного интерфейса КР580ИК51, преобразующая данные из параллельной формы в последовательную, БИС интерфейса для печатающих устройств, с программируемым хронированием и др. [7].

Быстродействующие преобразователи аналоговых сигналов. Сопряжение с микроЭВМ. Значительное число систем управления является аналоговым. Для преобразования аналоговых сигналов в цифровую форму и обработки их в реальном времени требуются быстродействующие аналого-цифровые преобразователи (АЦП) с разрешением по амплитуде, равным 8 и более разрядам двоичного числа.

Основными требованиями, предъявляемыми к этим АЦП, являются высокие разрешающая способность и быстродействие.

Разрешающая способность измеряется числом разрядов преобразователя N и определяет способность преобразователя реализовать 2^N уровней квантования.

В процессе преобразования непрерывного сигнала в двоично-кодированный выходной сигнал происходит его квантование, т.е. разбиение на конечное число дискретных уровней, отличающихся между собой на величину наименьшего значащего разряда. Этим объясняется появление конечной разрешающей способности АЦП.

Быстродействие — это время, необходимое для завершения преобразования аналогового сигнала в цифровой. Требуемое быстродействие (частота дискретизации) определяется по теореме Котельникова (частота дискретизации должна быть по крайней мере в два раза выше максимальной частоты в спектре преобразуемого сигнала) или из некоторых других условий, рассмотренных в § 4.1.

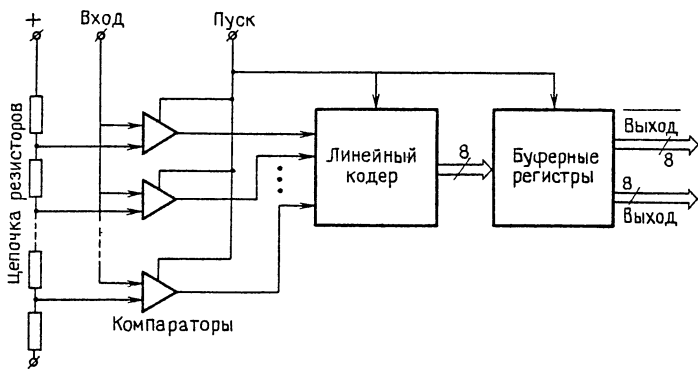
Быстродействие и разрешающая способность АЦП зависят от технологии изготовления больших интегральных схем и метода преобразования.

Широкое распространение при изготовлении АЦП, работающих на частотах до 100 МГц, получила биполярная технология. Разрешение этих преобразователей ограничено, однако, 4–6 разрядами. Это связано с тем, что плотность упаковки элементов в кристаллах БИС, выполненных по биполярной технологии, относительно невысока. Использование высокоскоростного параллельного принципа преобразования приводит в этом случае при увеличении числа разрядов к экспоненциальному росту размеров кристалла. Более прогрессивной для 8-разрядных АЦП является модифицированная 1-мкм биполярная технология на основе эмиттерно-связанной логики (ЭСЛ).

В параллельном АЦП (рис. 2.20), выполненном по технологии 1-мкм ЭСЛ, высокая скорость преобразования достигается за счет одновременной обработки аналогового входного сигнала дифференциальными амплитудными компараторами. В преобразователе участвуют в анализе сразу все $2^8 - 1 = 255$ компараторов величины входного сигнала.

На вторые входы компараторов подаются точные опорные напряжения с цепочки резисторов. Каждое опорное напряжение отличается от следующего на единицу младшего разряда. Компараторы, порог которых оказывается ниже напряжения преобразуемого сигнала, срабатывают. Остальные компараторы остаются в выключенном состоянии. Полученный в результате код преобразуется далее линейным кодером в двоичную форму.

Скорость такого преобразования сигналов зависит от паразитной емкости матрицы компараторов. Использование модифицированной 1-мкм технологии позволяет существенно уменьшить входную и выходную емкости матрицы компараторов, что обеспечивает преобразование до 75 млн. отсчетов в секунду [136].



Р и с. 2.20. Быстродействующий параллельный АЦП

Т а б л и ц а 2.4

Характеристики	СА 3300	SDA 5010
Технология	4-мкм К/МОП	3-мкм ЭСЛ
Количество разрядов	6	6
Максимальная частота дискретизации, МГц	15	100
Потребляемая мощность, мВт	140	450
Погрешность, \pm ед. младшего разряда	0,5	0,25
Площадь кристалла, мм ²	7,74	10,32

Более плотная упаковка кристаллов интегральных АЦП достигается по технологии комплементарных МОП-схем (К/МОП-схем) на сапфире. Так, 8-разрядный АЦП, выполненный по 4-мкм К/МОП-технологии, занимает около 2/3 площади кристалла, необходимой для такого АЦП, изготовленного по модифицированной 1-мкм биполярной технологии.

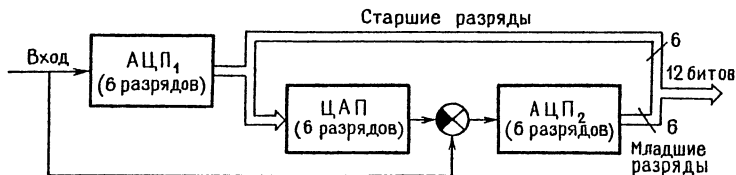
К/МОП-приборы энергетически выгоднее преобразователей, выполненных по биполярной ЭСЛ-технологии, но имеют примерно на порядок меньшую частоту дискретизации.

Характеристики некоторых параллельных АЦП даны в табл. 2.4 [90, 117].

Быстродействующий 12-разрядный АЦП можно создать на основе цифро-аналогового преобразователя с разрешением 6 битов и двух 6-разрядных АЦП, используя принцип двухступенчатого преобразования.

В двухступенчатых преобразователях аналоговый сигнал обрабатывается в двух амплитудных диапазонах: сначала осуществляется грубое преобразование, например, 6-разрядным АЦП; затем следует точное преобразование, позволяющее повысить разрешающую способность преобразователя.

В начале каждого цикла преобразования (рис. 2.21) первый АЦП₁ преобразует сигнал с разрешением 6 разрядов. ЦАП превращает полученный цифровой код в аналоговый сигнал, который вычитают из исходного сигнала. Разностный сигнал обрабатывается



Р и с. 2.21. Двухступенчатый преобразователь непрерывного сигнала

TDC1014	CA3308	TDC1007	TDC1025	K1107ПВ1
2-мкм ТТЛ	4-мкм К/МОП	2-мкм ТТЛ	1-мкм ЭСЛ	—
6	8	8	8	6
30	15	30	75	20
750	200	2000	2000	—
0,25	1	0,5	0,5	—
14,19	16,13	44,52	20,65	—

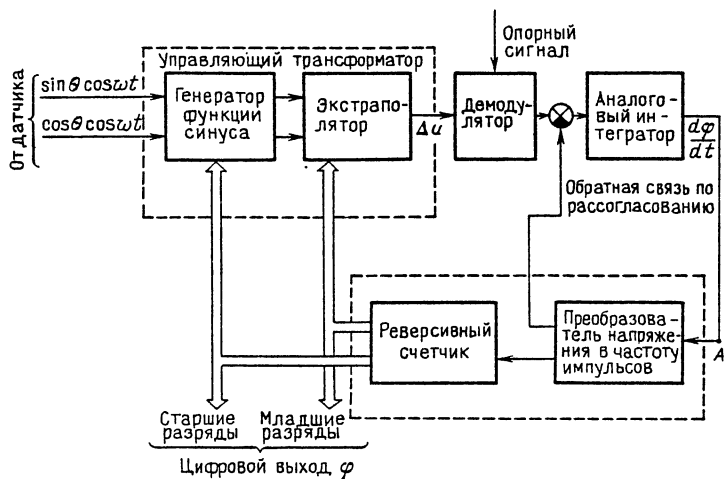
ступенью точного АЦП₂, образующего младшие разряды цифрового кода. Двухступенчатый АЦП позволяет реализовать преобразование с частотой дискретизации до 4 МГц [120].

Высокоточный 14-разрядный преобразователь угол-код разработан фирмой ILC/Data-Device Corp [141]. Основу преобразователя составляет система автоматического регулирования, показанная на рис. 2.22. Выходом системы является цифровой угол φ , содержащийся в реверсивном счетчике. Информация об угле θ сельсина или синусно-косинусного датчика заключается в двух составляющих: $\sin \theta \cos \omega t$ и $\cos \theta \cos \omega t$.

В узле суммирования основного контура регулирования находится управляющий трансформатор, сигнал с выхода которого $\Delta u(t, \theta, \varphi) = u_0 \sin(\theta - \varphi) \cos \omega t$. Полагая $u_0 = 1$, сигнал рассогласования получаем в виде $\Delta u(\theta - \varphi, t) = \sin(\theta - \varphi) \cos \omega t$. Несущая частота ω подавляется демодулятором. Система автоматического регулирования содержит два последовательно включенных каскада интегрирования: аналоговый и цифровой. Функции цифрового интегратора выполняет преобразователь напряжения в частоту и реверсивный счетчик. Это связано с тем, что цифровой угол, являющийся интегралом напряжения в точке А, равен произведению частоты на время. Система является астатической второго порядка.

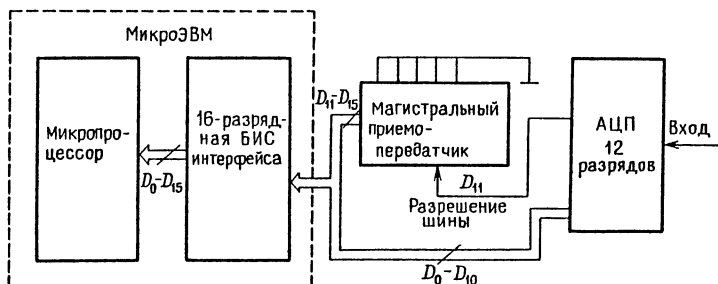
Основными характеристиками преобразователя являются следующие: точность 2,6'; мертвая зона составляет $\pm 1,2'$; дифференциальная нелинейность не превышает $\pm 0,25$ ед. младшего разряда; выходная реакция φ свободна от дрожания и не рыскает; напряжение питания 15В; потребляемый ток 30 мА.

Самостоятельной и далеко не тривиальной задачей является сопряжение АЦП с микроЭВМ. Это связано с тем, что сигнал с выхода АЦП представлен не в двоичном дополнительном коде, а знаковый разряд должен модифицироваться. Кроме того, длина разрядной сетки АЦП, как правило, не совпадает с соответствующей характеристикой микроЭВМ.



Р и с. 2.22. Преобразователь угол-код

Сопряжение АЦП с микроЭВМ может быть выполнено по схеме, показанной на рис. 2.23. Сигнал старшего значащего разряда АЦП (12-го разряда) используется для управления магистральным приемопередатчиком по входу РАЗРЕШЕНИЕ ШИНЫ. Когда входной сигнал преобразователя положителен, сигнал старшего разряда равен нулю и приемопередатчик закрыт. В результате пять старших разрядов D11–D15 кода, поступающего в микроЭВМ, также равны нулю. С другой стороны, когда входной сигнал отрицателен, а передатчик работает в режиме с тремя устойчивыми состояниями, в старшем разряде преобразователя – единица, и в пяти старших разрядах кода на входе микроЭВМ также появляются единицы. Управление схемой сопряжения осуществляется в режиме прерывания [92] или на основе организации обмена с квитированием.



Р и с. 2.23. Схема сопряжения АЦП с микроЭВМ

Быстродействующие преобразователи цифровых сигналов. Цифро-аналоговое преобразование сигналов применяется для сопряжения встраиваемой управляющей микроЭВМ с аналоговым приемником сигналов в системе управления. Кроме того, на основе ЦАП могут быть построены аналого-цифровые преобразователи (рис. 2.21).

Основными параметрами ЦАП являются разрешающая способность, точность и время установления.

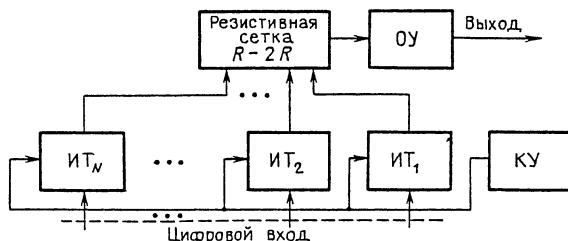
Разрешающая способность характеризует число уровней аналогового сигнала и выражается в виде общего числа двоичных разрядов входного цифрового сигнала преобразователя. При разрешающей способности N двоичных разрядов преобразователь должен обеспечивать формирование выходного аналогового сигнала, имеющего 2^N дискретных уровней.

Точность характеризуется величиной отклонения аналогового сигнала от расчетного значения и выражается в процентах от полного диапазона изменения выходного сигнала.

Временем установления называется интервал времени от момента скачкообразного изменения входного цифрового сигнала до момента времени, когда выходной аналоговый сигнал достигает нового установившегося значения с наименьшей погрешностью. Время установления характеризует быстродействие ЦАП.

В высокоскоростных ЦАП не должно быть случайных ошибок (глитчеров), причиной появления которых является быстрое изменение цифровых сигналов на входе преобразователей и неодинаковое время включения и выключения токовых ключей. Поэтому в состав ЦАП вводят схемы деглитчирования.

Наиболее распространенными высокоскоростными ЦАП являются параллельные преобразователи на коммутаторах тока с резистивной сеткой R-2R [213]. Схема такого преобразователя показана на рис. 2.24. Преобразователь содержит блок источников равных токов (ИТ₁, ..., ИТ_N), резистивную сетку R-2R, компенсационный (КУ) и выходной операционный (ОУ) усилители. В рассматриваемом ЦАП допустимы небольшие отклонения сопротивлений резис-



Р и с. 2.24. Схема ЦАП параллельного типа

Таблица 2.5

Характеристики	TDC— 1016J— 8	TDC— 1016J— 9	TDC— 1016J— 10	MP8308 ECL	5609
Разрешающая способность, бит	8	9	10	8	8
Время установления, мкс	0,03	0,03	0,03	0,0075	0,045
Линейность, ± 1 ед. младшего разряда	—	—	—	± 1/2	± 1/2
Потребляемая мощность, мВт	600	600	600	1100	—

торов от их номиналов, так как они могут быть скомпенсированы соответствующей регулировкой источников токов в разрядах преобразователя.

Параллельные ЦАП могут быть реализованы без резистивной сетки на интегральных делителях (генераторах) опорных токов. Отсутствие прецизионных резисторов и, вследствие этого, их высокая технологичность позволяют изготавливать такие преобразователи в виде монолитных интегральных микросхем [3]. В табл. 2.5 приведены параметры некоторых ЦАП [115].

Вопросы повышения точности ЦАП рассмотрены в работе [124].

§ 2.3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ МИКРОЭВМ

Представление информации в микроЭВМ. В микроЭВМ производится обработка информации, представляемой в различной, но чаще всего в числовой двоичной форме. Информацией внутри микроЭВМ являются команды и данные.

В большинстве микроЭВМ данные и команды обрабатываются по словам, длину которых измеряют в байтах или в битах (числе разрядов). Один байт равен восьми битам. Слова хранятся в ячейках памяти. Обычно считается, что одно слово хранится в одной ячейке памяти.

Представление информации в микромашинах связано с понятием "формат слова". Формат слова отображает организацию (назначение) отдельных разрядов в слове. Данные и команды могут быть представлены либо одним словом, либо несколькими. Во втором случае имеются два типа слов: слова команд (команды) и слова данных, которые распознаются только по их расположению в памяти.

Код операции представляет собой закодированное название операции, которую должна выполнить микроЭВМ. Совокупность кодов операций образует набор команд микроЭВМ. Число разря-

дов, выделенных для кода операции, ограничивает набор команд. Современные микроЭВМ выполняют от нескольких десятков до нескольких сотен команд.

Код операнда задает правило, по которому вычисляется адрес данных, участвующих в операции. Данные могут храниться в регистрах общего назначения микропроцессора, в ячейках памяти, в регистрах устройства ввода-вывода. Данные в микроЭВМ могут быть представлены в некотором диапазоне, который определяется длиной слова данных и используемым способом кодирования. Например, в микроЭВМ на базе микропроцессора серии КР580 для слова данных могут быть выделены восемь или шестнадцать разрядов и один знаковый разряд в регистре состояния. Следовательно, диапазон чисел при таком представлении равен 2^8 или 2^{16} . Обычно диапазон чисел связывают с теми экстремальными числами, которые способны обрабатывать микроЭВМ. Так, если необходимо обработать числа в диапазоне от -128 до 127 , то слово данных должно иметь диапазон 256 .

В микроЭВМ для операций с отрицательными числами широко используется дополнительный код двоичных чисел. Дополнительный код получается достаточно просто по следующему правилу: в двоичной записи числа необходимо заменить все единицы на нули, а нули на единицы и прибавить к получившемуся числу единицу в младшем разряде.

Получим для примера дополнительный код числа 77_{10} . Для этого сначала запишем 77 в двоичной форме, затем заменим единицы на нули, а нули на единицы и прибавим единицу:

$$\begin{array}{r}
 77 = 01001101 \\
 \quad 10110010 \text{ — дополнение до единицы} \\
 \quad + \quad \quad 1 \\
 \hline
 -77 = 10110011 \text{ — дополнительный код}
 \end{array}$$

Восемь двоичных разрядов позволяют представить числа в диапазоне от $10000000 = -128$ до $01111111 = +127$ (табл. 2.6).

Т а б л и ц а 2.6

Число		Число	
двоичное	десятичное	двоичное	десятичное
10000000	-128	00000001	1
10000001	-127	00000010	2
10000010	-126	00000011	3
...
11111110	-2	01111101	+125
11111111	-1	01111110	+126
00000000	0	01111111	+127

Программа работы микроЭВМ может быть написана в машинных кодах. Машинный код — это упорядоченная совокупность двоичных чисел, содержащих информацию о коде операции и операндах. Машинный код — ”язык”, понятный микроЭВМ. МикроЭВМ расшифровывает код операции как ответ на вопрос: ”Что нужно сделать микроЭВМ по данной команде?”. Операнд содержит информацию, определяющую адрес данных, над которыми совершаются действия, предусмотренные в команде. Это могут быть содержимое регистров общего назначения, ячеек памяти, данные внешних устройств и т.д.

Например, команда пересылки данных из регистра R1 в регистр R5 микроЭВМ LSI-11 имеет следующий машинный код

Машинный код команды	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1
----------------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Номер разряда	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
---------------	----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

В примере показана команда, состоящая из одного машинного слова длиной в шестнадцать битов. Здесь шесть младших разрядов слова с кодом 000101 задают вторую часть операнда — содержимое регистра R5. Шесть следующих разрядов слова задают первую часть операнда — содержимое регистра R1. Разряды 12 — 14 содержат код операции пересылки, а в пятнадцатом разряде записан признак байтовой обработки. Если содержимым пятнадцатого разряда является нуль, то микроЭВМ обрабатывает слова длиной в шестнадцать битов (два байта). Запись в пятнадцатом разряде единицы является указанием микроЭВМ обрабатывать слова длиной в восемь битов (один байт).

В списке команд микроЭВМ семейства LSI-11 имеются команды длиной в два 16-разрядных слова. Ниже приведен пример такой команды.

Код команды пересылки числа в регистр общего назначения R5:

```
0 001 010 111 000 101
0 111 100 000 111 111
```

Здесь второе слово является непосредственным операндом.

Из сказанного видно, что составление программы в двоичном коде является трудоемкой задачей, решение которой сопряжено с большой вероятностью появления ошибок.

Несколько лучшие возможности предоставляет восьмеричный код, который вводится по правилам, приведенным в табл. 2.7.

Таблица 2.7

Десятичный код	Двоичный код	Восьмеричный код	Десятичный код	Двоичный код	Восьмеричный код
0	0000	00	6	0110	06
1	0001	01	7	0111	07
2	0010	02	8	1000	10
3	0011	03	9	1001	11
4	0100	04	10	1010	12
5	0101	05			

Для записи команды в восьмеричном коде каждое слово условно представляется состоящим из элементарных конструкций длиной в три бита. Счет ведется с младшего разряда. Каждая элементарная трехбитовая конструкция заменяется одноразрядным числом восьмеричной системы счисления. Очевидно, что одно машинное 16-разрядное двоичное слово представляется 6-разрядным восьмеричным кодом.

Ниже следует пример перевода машинного слова, представленного в двоичной и в восьмеричной системах счисления:

Машинный двоичный код команды 0 001 000 001 000 101

Восьмеричный код команды 0 1 0 1 0 5

Приведенная ранее команда из двух машинных слов в восьмеричном коде записывается следующим образом:

012705
074077.

Запись команд в восьмеричной системе счисления при программировании САУ с микроЭВМ является широко распространенной, и мы также будем ею пользоваться в дальнейшем.

Очевидно, что отмеченные ранее недостатки программирования в двоичном коде сохраняются, однако, и при программировании в восьмеричном коде.

Поэтому большее распространение получило программирование на языке ассемблера. Ассемблер — это язык символического программирования. В нем каждому машинному слову поставлено в соответствие понятное человеку мнемоническое обозначение. Как правило, это обозначение образуется в форме аббревиатуры от английских или русских слов, определяющих смысл команд. Например, мнемоника ASR расшифровывается предложением "Arithmetic Shift Right" (арифметический сдвиг вправо).

Достоинствами программирования на языке ассемблер по сравнению с программированием в машинных кодах являются простота запоминания человеком списка команд, лучшее восприятие им программы, меньшая вероятность ошибок. Важным является также то, что программы, составленные опытными программистами на языке ассемблер, требуют минимальной емкости памяти и обеспечивают наивысшую производительность микроЭВМ.

Поэтому в настоящее время язык ассемблер является наиболее распространенным языком программирования микроЭВМ, входящих в системы автоматического управления.

Известно большое количество версий языка ассемблер. Каждая микроЭВМ снабжена своим языком. Однако все версии этого языка имеют, несмотря на внешнее различие, много общих черт. Считается, что специалисту, работающему с микропроцессорами и микроЭВМ, достаточно однажды усвоить любую версию языка ассемблер для того, чтобы при необходимости быстро приспособиться к любой другой его версии. Одним из типичных языков низкого уровня является ассемблер микроЭВМ семейства LSI-11, изучение которого полезно специалистам, разрабатывающим программное обеспечение САУ с системой команд микроЭВМ семейства "Электроника 60".

Методы адресации. В микроЭВМ LSI-11 реализованы четыре основных метода адресации ячеек памяти, регистров общего назначения или внешних устройств. Эти методы сведены в табл. 2.8.

В таблице приняты следующие обозначения: *rs*, *rd* – имена регистров общего назначения, являющихся соответственно в зависимости от метода адресации источником (*src*) и приемником (*dst*) операндов или участвующих в формировании адресов источника и приемника; *rsd* – имя регистра общего назначения, являющегося одновременно источником и приемником операндов или участвующего в формировании адреса источника-приемника; $[\cdot]$ – содержимое регистра общего назначения; $[(\cdot)]$ – содержимое ячейки памяти; * – символ некоторой операции; (\rightarrow) – оператор пересылки данных; x , x_1 , x_2 – смещения; *m* – код метода адресации.

Группа операндов SD применяется в двухадресных командах. В каждой команде этой группы операнд состоит из двух частей, определяющих исполнительные адреса по правилам, приведенным во втором столбце табл. 2.8.

Группа операндов D используется в одноадресных командах. Характерным для этих команд является то, что источник и приемник данных здесь один и тот же. Его исполнительный адрес также определяется по правилам, приведенным в табл. 2.8.

Т а б л и ц а 2.8

Условное обозначение операнда	Правила адресации источника (<i>src</i>) и приемника (<i>dst</i>) команд и данных	Прямые методы адресации		Косвенные методы адресации	
		наименование	код (<i>m</i>)	наименование	код (<i>m</i>)
Группа SD-операндов					
1.1. <i>rs, rd</i>	$[rs] * [rd] \rightarrow [rd]$	прямая регистровая адресация	0	косвенная регистровая адресация	1
2.1. $(rs) +, rd$	$[(rs)] * [rd] \rightarrow [rd],$ $[rs] + 2 \rightarrow [rs]$	автоинкрементная адресация	2	косвенная автоинкрементная адресация	3
2.2. <i>rs, (rd) +</i>	$[rs] * [(rd)] \rightarrow [(rd)],$ $[rd] + 2 \rightarrow [rd]$	то же	2	то же	3
2.3. $(rs) +, (rd) +$	$[(rs)] * [(rd)] \rightarrow [(rd)],$ $[rs] + 2 \rightarrow [rs];$ $[rd] + 2 \rightarrow [rd]$	то же	2	то же	3
3.1. $-(rs), rd$	$[rs] - 2 \rightarrow [rs];$ $[(rs)] * [rd] \rightarrow [rd]$	автодекрементная адресация	4	косвенная автодекрементная адресация	5

3.2. $rs, -(rd)$	$[rd] - 2 \rightarrow [rd];$ $[rs] * [(rd)] \rightarrow [(rd)]$	то же	4	то же	5
3.3. $-(rs), -(rd)$	$[rs] - 2 \rightarrow [rs];$ $[rd] - 2 \rightarrow [rd];$ $[(rs)] * [(rd)] \rightarrow [(rd)]$	то же	4	то же	5
4.1. $x(rs), rd$	$[(rs + x)] * [rd] \rightarrow [rd]$	индексная адресация	6	косвенная индексная адресация	7
4.2. $rs, x(rd)$	$[rs] * [(rd + x)] \rightarrow [(rd + x)]$		6		7
4.3. $x_1(rs), x_2(rd)$	$[(rs + x_1)] * [(rd + x_2)] \rightarrow$ $\rightarrow [(rd + x_2)]$		6		7
Группа D-операндов					
5.1. rsd	$* [rsd] \rightarrow [rsd]$	прямая регистровая адресация	0	косвенная регистровая адресация	1
5.2. $(rsd) +$	$* [(rsd)] \rightarrow [(rsd)];$ $[rsd] + 2 \rightarrow [rsd]$	автоинкрементная адресация	2	косвенная автоинкрементная адресация	3
5.3. $-(rsd)$	$[rsd] - 2 \rightarrow [rsd];$ $* [(rsd)] \rightarrow [(rsd)]$	автодекрементная адресация	4	косвенная автодекрементная адресация	5
5.4. $x(rsd)$	$* [(rsd + x)] \rightarrow [(rsd + x)]$	индексная адресация	6	косвенная индексная адресация	7

Пр и м е ч а н и е: косвенная адресация вводится записью перед соответствующей частью операнда знака @; исполнительный адрес в этом случае определяется по правилам, приведенным на стр. 129

Прямая регистровая адресация. В операции *, показанной в пп. 1.1 и 5.1 табл. 2.8, участвуют данные, хранящиеся в регистрах общего назначения *rs* и *rd* или *rsd* соответственно для двухадресных и одноадресных команд. В командах номера регистров обязательно конкретизируются, а мнемоника выбирается из Приложения 1 по функции команды. Например, команда пересылки данных из регистра R1 в регистр R5 – MOV R1, R5, а команда увеличения на единицу содержимого регистра R1 – INC R1.

Автоинкрементная адресация (адресация с автоувеличением). В операции * (см. пп. 2.1 – 2.2. и 5.2 табл. 2.8) участвуют содержимое ячейки памяти, адрес которой хранится в регистре общего назначения, выделенном круглыми скобками, и содержимое регистра общего назначения. Содержимое регистра, выделенного в первом столбце табл. 2.8 круглыми скобками, увеличивается на два.

Например, команда пересылки данных из ячейки памяти, адрес которой хранится в регистре R1, в регистр R5 с увеличением содержимого регистра R1 на два – MOV (R1)+, R5, а команда увеличения на единицу содержимого ячейки памяти с использованием этого метода адресации – INC (R1)+.

Автодекрементная адресация (адресация с автоуменьшением). Данный способ адресации аналогичен автоинкрементной адресации. Различие заключается в том, что содержимое регистра общего назначения, выделенного в первом столбце табл. 2.8 круглыми скобками, уменьшается на два до начала выполнения операции.

Например, команда пересылки данных из ячейки памяти, адрес которой на два меньше содержимого регистра R1, в регистр R5 – MOV –(R1), R5, а команда увеличения на единицу содержимого ячейки памяти, адрес которой меньше содержимого регистра R1 на два, – INC – (R1).

Индексная адресация (адресация с адресной константой). В операции * (см. пп. 4.1 – 4.2, 5.4 табл. 2.8) участвуют содержимое регистра общего назначения и содержимое ячейки памяти, адресом которой является сумма смещения *x* с содержимым регистра общего назначения, выделенного в команде круглыми скобками.

Например, команда пересылки данных из ячейки памяти, адрес которой равен содержимому регистра R1, сложенному со смещением $x = 100$, в регистр R5 – MOV 100(R1), R5, а команда увеличения на единицу содержимого ячейки памяти, адрес которой равен содержимому регистра R1, сложенному со смещением $x = 100$, – INC 100(R1).

Непосредственная адресация. Этот способ адресации используется для включения в команды непосредственных операндов-констант. Непосредственная адресация является частным случаем автоинкрементной адресации с использованием в качестве источника данных ячейки памяти, адресуемой счетчиком команд R7.

Эта особенность непосредственной адресации в мнемонике языка ассемблер не проявляется. В микроЭВМ адресные выражения вида $\#n$, где n – константа, воспринимаются как непосредственные операнды. Команды с непосредственным способом адресации состоят из двух машинных слов, причем второе слово содержит операнд n .

Примером команды с непосредственной адресацией является пересылка числа 100 в регистр R5: MOV # 100, R5.

Относительная адресация Частный случай индексной адресации с использованием в операндах $x(rs)$, $x(rd)$ и $x(r)$ табл. 2.8 счетчика команд – регистра R7.

Методы косвенной адресации. В микроЭВМ имеется возможность организации косвенной адресации.

В мнемонике языка ассемблер признаком косвенной адресации является знак @ ("a" коммерческое), который ставится перед соответствующей частью операнда в первом столбце табл. 2.8 и, следовательно, в команде. Включение знака @ в команду эквивалентно введению пары круглых скобок при обеспечении адресации источника и приемника по правилам, приведенным во втором столбце табл. 2.8.

Например, команда пересылки данных из регистра R1 в ячейку памяти, адрес которой хранится в регистре R5, – MOV R1, @(R5), команда пересылки данных в регистр R5 из ячейки памяти, адрес которой хранится в другой ячейке памяти, адресуемой содержимым регистра общего назначения R1, с увеличением содержимого R1 на два – MOV @(R1)+, R5, а команда увеличения на единицу содержимого ячейки памяти, адрес которой хранится в регистре R1, – INC @R1.

Общие представления машинных кодов команд. Формат одноадресных команд имеет следующий вид:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

Код признака байтовой обработки

Код операции

Код способа адресации

Код регистра общего назначения

Если обозначить буквами b , k , m , d соответственно восьмеричные коды признака байтовой обработки, операции, метода

адресации и регистра общего назначения, то формат одноадресных команд можно представить в виде

$$\boxed{b \quad k \quad m \quad d} \quad (2.1)$$

Заметим, что код операции k образован трехразрядным числом восьмеричной системы счисления, а код признака байтовой обработки b может быть равен только нулю или единице.

Представление (2.1) является машинным кодом одноадресных команд, записанным в общем виде.

Например, машинный код команды INC D, представленный в общем виде, будет $b052 md$. Здесь D – один из операндов соответствующей группы табл. 2.8.

Формат двухадресных команд имеет следующий вид:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Код признака байтовой обработки	Код операции			Код способа адресации регистра-источника		Код регистра источника	Код способа адресации регистра-приемника			Код регистра-приемника					

Если обозначить буквами s , d соответственно восьмеричные коды регистров общего назначения, являющихся в зависимости от метода адресации источником и приемником операндов или участвующих в формировании адресов источника и приемника, то формат двухадресных команд имеет вид

$$\boxed{b \quad k \quad m \quad s \quad m \quad d} \quad (2.2)$$

В данном случае код операции является одноразрядным восьмеричным числом. Представление (2.2) соответствует машинному коду двухадресных команд, представленному в общем виде.

Например, машинный код команды MOVSD в общем виде: $b1msmd$. Здесь SD – один из операндов соответствующей группы табл. 2.8.

Представления (2.1) и (2.2) используются при записи машинных кодов команд (см. Приложение 1).

С л о в о с о с т о я н и я п р о ц е с с о р а. Это слово используется при организации в программах циклов и ветвлений по результатам выполнения операций. Результаты операций называются в этом случае условиями.

Условия, которые могут быть проверены в микроЭВМ семейства LSI-11 – это нуль, знак, перенос и арифметическое переполнение. Эти условия называют признаками результата операции и обозначаются следующим образом: Z – признак нуля; N – признак

нак знака; С — признак переноса; V — признак арифметического переполнения.

Признак (соответствующий триггер признака результата операции) считается установленным, если определенное этим признаком условие выполнено. Так, например, если результат операции равен нулю, то признак Z устанавливается в единичное состояние.

Совокупность состояний, в которых находятся признаки результата операции, образует слово состояния процессора.

Слово состояния процессора для каждой команды микроЭВМ семейства LSI-11 приведено в Приложении 1.

Функции команд микроЭВМ. Базовая система машинных команд микроЭВМ семейства LSI-11 насчитывает свыше 80 команд, основные из которых приведены в Приложении 1.

Команды классифицированы в соответствии с действиями, которые по ним выполняются, т.е. в соответствии с функциями команд. Таким способом команды разделены на пять основных категорий.

Команды пересылок: по командам этого типа производится, например, пересылка данных из одной области памяти в другую, в регистр общего назначения или в регистр внешнего устройства.

Арифметические и логические команды: именно эти команды действительно преобразуют информацию. К этому типу относятся операции сложения, вычитания, логические операции И, ИЛИ и некоторые другие команды.

Команды переходов: эти команды используются для изменения обычного порядка выполнения команд в программе. Существуют команды условного и безусловного перехода. Например, можно использовать команду условного перехода для выбора следующей команды из некоторой заданной ячейки памяти, если результат выполнения некоторой предшествующей команды окажется равным нулю.

Специальные команды: эти команды имеют различное назначение. Можно, например, по команде WAIT перевести процессор в состояние ожидания сигнала прерывания, а по команде HALT остановить выполнение программы.

Команды битовых операций: часто требуется устанавливать или сбрасывать комбинации кодов слова состояния процессора, эту задачу решают команды данного типа.

Команды сдвига: эти команды вызывают одновременное перемещение границ расположения некоторой информации. Например, сдвиг слова влево на один бит означает одновременное смещение всех его битов влево на одну позицию.

В микроЭВМ семейства LSI-11 возможны два типа сдвигов: арифметические и логические. Разница между ними заключается в обращении с содержимым знакового разряда. При арифметическом сдвиге содержимое знакового разряда не меняется, остальные биты слова сдвигаются. При логическом сдвиге знаковый разряд ничем не отличается от остальных.

Все команды могут выполняться над словами длиной в два байта и в один байт. В последнем случае к мнемонике операции добавляется без пробела буква В. Например, мнемоника команды вычитания байтов имеет вид SUBB. В старшем разряде машинного кода в этом случае записывается единица.

Функция команды и метод адресации позволяют определить группу, из которой следует выбирать команду, наиболее подходящую для выполнения заданных действий.

Функции каждой команды микроЭВМ семейства LSI-11 приведены в Приложении 1. Рассмотрим некоторые примеры.

Команды пересылки имеют мнемоническое обозначение MOV (move). К мнемонике MOV через пробел при записи команды на языке ассемблер присоединяется по правилам табл. 2.8 операнд группы D или операнд группы SD. Если программисту требуется воспользоваться косвенной адресацией, то перед соответствующей частью операнда следует поставить знак @.

В командах MOV в качестве источника (*src*) и приемника (*dst*) операндов могут использоваться соответственно: регистр — регистр, ячейка памяти — регистр, регистр — ячейка памяти, ячейка памяти — ячейка памяти.

Так как в микроЭВМ каждому устройству, включая ячейки памяти, регистры общего назначения, регистры внешних устройств, присваивается адрес на общей шине, то командой MOV можно обращаться к регистрам внешних устройств как к ячейкам ОЗУ процессора.

Рассмотрим, например, команду MOV R1, R5. Символами R1 и R5 обозначены первый и пятый регистры общего назначения. Такая запись команды на основании правил адресации операндов, приведенных в п. 1 табл. 2.8, означает, что содержимое регистра R1 должно быть помещено в регистр R5, т.е. $R1 \rightarrow R5$. Прежнее содержимое регистра R5 пропадает. После выполнения команды содержимое регистра R1 не изменяется и представляет собой копию содержимого регистра R5. Машинный код команды при $b = 0$ будет 010105. Другой пример команды пересылки MOV R1, (R5)+. Такая запись команды на основании правила 2.2. табл. 2.8 означает, что содержимое регистра R1 помещается в ячейку памяти, адрес которой хранится в регистре R5, т.е. $[R1] \rightarrow [(R5)]$. Содержимое регистра R5 увеличивается на два. Машинный код команды имеет вид 010125, $b = 0$.

И еще один пример: MOV R1, -(R5). В этом случае применяем правило 3.2 табл. 2.8, из которого следует, что результатом выполнения команды является загрузка содержимого регистра R1 в ячейку памяти, адрес которой определяется содержимым регистра R5, уменьшенным на два. Машинный код команды 010145, $b = 0$.

В двух последних примерах происходит автоматическое изменение адреса. Этим обеспечивается, например, перемещение указателя номера ячеек памяти по последовательным элементам таблицы операндов.

Команда MOV 100 (R1), 50 (R5) в соответствии с правилом 4.3 табл. 2.8 расшифровывается следующим образом: поместить содержимое ячейки памяти, адрес которой равен сумме содержимого регистра R1 со смещением 100 в ячейку памяти, адрес которой равен содержимому регистра R5, сложенному со смещением 50. Машинный код команды 016165.

Такая команда обеспечивает в общем случае произвольный доступ к данным в таблицах или стеках.

Группа команд арифметических операций объединяет двенадцать команд. Команда сложения ADD SD имеет следующее описание: содержимое источника (*src*), складывается с содержимым приемника (*dst*) и результат записывается по адресу приемника. Содержимое источника (*src*) не изменяется. Сложение выполняется в двоичном дополнительном коде.

По команде вычитания SUB SD содержимое источника вычитается из содержимого приемника и результат записывается по адресу приемника.

Команда COM SD служит для замены содержимого приемника (*dst*) его обратным кодом.

Например, COM (R0)+. До операции $[(R0)] = 013333$. После операции $[(R0)] = 164444$. Кроме того, $[R0] + 2 \rightarrow [R0]$.

Если требуется установить триггеры Z и N в состояния, зависящие от содержимого приемника, то в этом случае применяется команда TST D.

Например, TST @R1. До операции $[(R1)] = 012300$, триггеры N и Z находятся в единичном состоянии. После операции $[(R1)] = 012300$, а триггеры N и Z обнуляются.

Команда умножения MUL S, *rd* выполняется следующим образом: перемножаются содержимое источника (*src*) и приемника. Приемником является один из регистров общего назначения. Результат перемножения помещается в регистр, используемый в качестве приемника, и в следующий за ним регистр, если регистр приемника имеет четный номер. Если регистр приемника имеет нечетный номер, то сохраняется только младшая часть результата. Операнды записываются в двоичном дополнительном коде.

Команда деления $DIV S, rd$ также выполняется над данными, представленными в двоичном дополнительном коде. Регистр rd выбирается четным, в нем и в следующем за ним регистре помещается делимое. Частное заносится в регистр общего назначения rd , а остаток — в следующий регистр.

В группе логических операций команда $BIT SD$ используется для проверки состояния разрядов источника (src), для которых установлены соответствующие разряды в приемнике (dst). Результат проверки фиксируется триггерами N и Z . Например, команда $BIT 30, @R3$ расшифровывается так: до операции $[(R3)] = 30, N = 1, Z = 1$. После операции $[(R3)] = 30, N = 0, Z = 0$.

Команда $CMP SD$ сравнивает содержимое источника и приемника и изменяет признаки. Содержимое источника и приемника сохраняется.

В группе специальных команд рассмотрим подробнее команду $JSR rsd, D$. По этой команде содержимое регистра общего назначения rsd , который в данном случае называется регистром связи, заносится в ячейку памяти стекового типа, адрес которой хранится в регистре-указателе стека. Содержимое счетчика команд переписывается в регистр связи, а в счетчик команд заносится новое число, которое определяет начальный адрес подпрограммы, к которой осуществляется переход по команде $JSR rsd, D$.

Функции остальных команд очевидны из пояснений, даваемых в Приложении 1.

Программирование микроЭВМ с изменяемым набором команд. Для микроЭВМ на базе микропроцессоров серий $K589, K582, K584, K1800, K1802, K1804$ и др. используется внешнее микропрограммирование. Как отмечено выше, это значит, что каждая команда представляется в виде микроинструкций. Совокупность микропрограмм образует программу работы микроЭВМ.

Поскольку основные принципы микропрограммирования являются общими, рассмотрим их на примере программирования микроЭВМ на комплекте БИС серии $K589$.

Каждая микроинструкция имеет в этом случае формат, включающий несколько полей:

F6	F5	F4	F3	F2	F1	F0	K1	K0	YA6	YA5	YA4	YA3	YA2	YA1	YA0
----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----

Код микрокоманды

Код маски

Код управления адресом микрокоманды

УФ3	УФ2	УФ1	УФ0	ЗМ
-----	-----	-----	-----	----

Код управления логикой флажков

Код управления загрузкой

Эти поля необходимы для решения следующих задач: предоставление информации блоку центральных процессорных элементов (ЦПЭ) о коде микрокоманды, включающем код микрооперации и операнда (соответствующие разряды кода обозначены F0 – F6) и коде маски (K0, K1), необходимой для расширения функциональных возможностей ЦПЭ; управление адресом следующей микрокоманды (разряды УА0 – УА6) и логикой флажков (разряды УФ0 – УФ3); управление загрузкой в блок микропрограммного управления начального адреса микропрограммы из основной памяти (разряд ЗМ).

Остановимся на принципах формирования указанных кодов.

Код F0 – F6 зависит от содержания микрооперации, которую в ходе выполнения микроинструкции должен осуществить блок ЦПЭ. Для уяснения принципов формирования кода F0 – F6 следует обратиться к списку микроопераций, приведенному в табл. 2.9.

Каждой микрооперации соответствует определенный код так называемых Ф- и Р-полей. Эти коды приведены в табл. 2.10.

Запишем для примера код микрооперации LMI с участием аккумулятора А.

Обозначение кода Ф-поля для данной микрооперации Ф1, а кода Р-поля – Р1. Следовательно, кодом операции является 001, кото-

Т а б л и ц а 2.9

Мнемоническое обозначение микрооперации	Обозначение кода		Код маски K0, K1	Содержание микрооперации
	Ф-поле	Р-поле		
1	2	3	4	5
1. Микрооперации пересылки				
LMI	Ф1	Р1	00	$R \rightarrow PA; R + C1 \rightarrow R$
LMM	Ф1	Р2	00	$M \rightarrow PA; M + C1 \rightarrow AT$
SDA	Ф2	Р2	11	$A - 1 + C1 \rightarrow AT$
LDI	Ф2	Р3	11	$B - 1 + C1 \rightarrow AT$
ACM	Ф3	Р2	00	$M + C1 \rightarrow AT$
LTM	Ф5	Р2	11	$C1 \vee M \rightarrow C0; M \rightarrow AT$
LMF	Ф6	Р2	00	$C1 \rightarrow C0; \overline{M} \rightarrow AT$
CMR	Ф7	Р1	00	$C1 \rightarrow C0; \overline{R} \rightarrow R$
LCM	Ф7	Р2	00	$C1 \rightarrow C0; \overline{M} \rightarrow AT$
CMA	Ф7	Р3	00	$C1 \rightarrow C0; \overline{AT} \rightarrow AT$
2. Арифметические микрооперации				
ILR	Ф0	Р1	00	$R + C1 \rightarrow R, A$
ALR	Ф0	Р1	11	$A + R + C1 \rightarrow R, A$
ACM	Ф0	Р2	00	$M + C1 \rightarrow AT$

Таблица 2.9 (окончание)

1	2	3	4	5
2. Арифметические микрооперации				
AMA	Ф0	P2	11	$M + A + C1 \rightarrow AT$
DSM	Ф1	P1	11	$11 \rightarrow PA; R - 1 + C1 \rightarrow R$
LDM	Ф1	P2	11	$11 \rightarrow PA; M - 1 + C1 \rightarrow AT$
CIA	Ф1	P3	00	$\overline{AT} + C1 \rightarrow AT$
DCA	Ф1	P3	11	$AT - 1 + C1 \rightarrow AT$
SDR	Ф2	P1	11	$A - 1 + C1 \rightarrow R$
INR	Ф3	P1	00	$R + C1 \rightarrow R$
ADR	Ф3	P1	11	$A + R + C1 \rightarrow R$
INA	Ф3	P3	00	$AT + C1 \rightarrow AT$
AIA	Ф3	P3	11	$B + AT + C1 \rightarrow AT$
3. Микрооперации управления				
DSM	Ф1	P1	11	$11 \rightarrow PA; R - 1 + C1 \rightarrow R$
LDM	Ф1	P2	11	$11 \rightarrow PA; M - 1 + C1 \rightarrow AT$
CLR	Ф4	P1	00	$C1 \rightarrow C0; 00 \rightarrow R$
CLA	Ф4	P2	00	$C1 \rightarrow C0; 00 \rightarrow AT$
NOP	Ф6	P1	00	$C1 \rightarrow C0; R \rightarrow R$
CSR	Ф2	P1	00	$C1 - 1 \rightarrow R$
CSA	Ф2	P2	00	$C1 - 1 \rightarrow AT$
4. Логические микрооперации				
ANR	Ф4	P1	11	$C1 \vee (R \wedge A) \rightarrow C0; R \wedge A \rightarrow R$
ANM	Ф4	P2	11	$C1 \vee (M \wedge A) \rightarrow C0; M \wedge A \rightarrow AT$
ANI	Ф4	P3	11	$C1 \vee (AT \wedge B) \rightarrow C0; AT \wedge B \rightarrow AT$
TZR	Ф5	P1	11	$C1 \vee R \rightarrow C0; R \rightarrow R$
LTM	Ф5	P2	11	$C1 \vee M \rightarrow C0; M \rightarrow AT$
TZA	Ф5	P3	11	$C1 \vee AT \rightarrow C0; AT \rightarrow AT$
ORR	Ф6	P1	11	$C1 \vee A \rightarrow C0; R \vee A \rightarrow R$
ORM	Ф6	P2	11	$C1 \vee A \rightarrow C0; M \vee A \rightarrow AT$
ORI	Ф6	P3	11	$C1 \vee B \rightarrow C0; B \vee AT \rightarrow AT$
XNR	Ф7	P1	11	$C1 \vee (R \wedge A) \rightarrow C0; R \overline{\vee} A \rightarrow R$
XNM	Ф7	P2	11	$C1 \vee (M \wedge A) \rightarrow C0; M \overline{\vee} AT$
XNL	Ф7	P3	11	$C1 \vee (AT \wedge B) \rightarrow C0; B \overline{\vee} AT \rightarrow AT$
5. Микрооперации сдвига				
SRA	Ф0	P3	00	$(AT)_0 \rightarrow СП0; (AT)_1 \rightarrow (AT)_0;$ $СП1 \rightarrow (AT)_1$
ALR	Ф0	P1	11	$A + R + C1 \rightarrow R, A$

Примечание: R — один из регистров общего назначения R0 — R9; PA — регистр адреса; C1 — вход переноса; M — вход шины данных; A — аккумулятор; C0 — выход переноса; T — регистр общего назначения; СП1 — вход сдвига вправо; СП0 — выход сдвига вправо; B — вход шины данных внешнего устройства; AT — результат помещается в A или T; R, A — результат помещается в R и в A; $\overline{\vee}$ — исключающее ИЛИ; $(AT)_0$, $(AT)_1$ — младший и старший разряды соответственно.

Т а б л и ц а 2.10

Обозначение кода Ф- или Р-поля	Регистр, участ- вующий в микрокоманде	Коды						
		Ф-поле			Р-поле			
		F6	F5	F4	F3	F2	F1	F0
Ф0	—	0	0	0	—	—	—	—
Ф1	—	0	0	1	—	—	—	—
Ф2	—	0	1	0	—	—	—	—
Ф3	—	0	1	1	—	—	—	—
Ф4	—	1	0	0	—	—	—	—
Ф5	—	1	0	1	—	—	—	—
Ф6	—	1	1	0	—	—	—	—
Ф7	—	1	1	1	—	—	—	—
P1	R0	—	—	—	0	0	0	0
	R1	—	—	—	0	0	0	1
	R2	—	—	—	0	0	1	0
	R3	—	—	—	0	0	1	1
	R4	—	—	—	0	1	0	0
	R5	—	—	—	0	1	0	1
	R6	—	—	—	0	1	1	0
	R7	—	—	—	0	1	1	1
	R8	—	—	—	1	0	0	0
	R9	—	—	—	1	0	0	1
P2	T	—	—	—	1	0	1	0
	A	—	—	—	1	0	1	1
P3	T	—	—	—	1	1	1	0
	A	—	—	—	1	1	1	1

рый должен быть занесен в разряды F6, F5, F4. Код операнда, которым в примере является содержимое аккумулятора, равен 1101. Этот код заносится соответственно в разряды F3, F2, F1, F0. Таким образом, кодом микрокоманды LMI(A) является код 001 1101, который при программировании микропроцессорного устройства заносится в разряды F6–F0 ячейки управляющей памяти.

Особое место в списке микроопераций занимают микрооперация ANR и ей подобные. Смысл этой микрооперации заключается в следующем. Данные со входа переноса логически складываются с 1 или 0, образуящимися в результате операции И над содержи-

Таблица 2.11

Название	Обозначение	Код управления адресом						
		YA6	YA5	YA4	YA3	YA2	YA1	YA0
Переход в текущей колонке	JCC	0	0	Y4	Y3	Y2	Y1	Y0
Переход в нулевую строку	JZR	0	1	0	Y3	Y2	Y1	Y0
Переход в текущей строке	JCR	0	1	1	Y3	Y2	Y1	Y0
Переход в текущей колонке в группе адресов строк	JCE	1	1	1	0	Y2	Y1	Y0
Переход по содержанию триггера	JFL	1	0	0	Y3	Y2	Y1	Y0
Переход по содержанию триггера	JCF	1	0	1	0	Y2	Y1	Y0
Переход по содержанию триггера	JZF	1	0	1	1	Y2	Y1	Y0
Переход по содержанию регистра команд	JPR	1	1	0	0	Y2	Y1	Y0
Переход по левым разрядам регистра команд	JLL	1	1	0	1	Y2	Y1	Y0
Переход по правым разрядам регистра команд	JRL	1	1	1	1	1	Y1	Y0
Переход по разрядам команды K4–K7	JPX	1	1	1	1	0	Y1	Y0

мым регистра R и аккумулятора по правилу: если при выполнении операции И над содержимым регистра R и аккумулятора A в любом из разрядов результата получается 1, то и результатом операции $(R \wedge A)$ является 1; результатом операции $(R \wedge A)$ является 0, если в обоих разрядах логического И результатом являются нули.

Далее результат логического сложения $C1 \vee (R \wedge A)$ помещается на выход переноса. Результат логического умножения содержимого регистра R с содержимым аккумулятора помещается также в регистр R. Напомним, что R — это один из регистров общего назначения. В операции номер или обозначение регистра обязательно конкретизируется. Например, в программе, составленной в мнемонике языка, который называется в этом случае микроассемблером, должно быть записано ALR (R9, A), но не ALR (R, A).

Адрес следующей строки					Адрес следующей колонки			
МА8	МА7	МА6	МА5	МА4	МА3	МА2	МА1	МА0
У4	У3	У2	У1	У0	М3	М2	М1	М0
0	0	0	0	0	У3	У2	У1	У0
М8	М7	М6	М5	М4	У3	У2	У1	У0
М8	М7	У2	У1	У0	М3	М2	М1	М0
М8	У3	У2	У1	У0	М3	0	1	Ф
М8	М7	У2	У1	У0	М3	0	1	С
М8	М7	У2	У1	У0	М3	0	1	Z
М8	М7	У2	У1	У0	PK3	PK2	PK1	PK0
М8	М7	У2	У1	У0	0	1	PK3	PK2
М8	М7	1	У1	У0	1	1	PK1	PK0
М8	М7	М6	У1	У0	К7	К6	К5	К4

В списке микроопераций для каждой микрокоманды указан код маскирования К0, К1. Поэтому заполнение соответствующих разрядов микроинструкции не представляет каких-либо трудностей.

Несколько сложнее заполняются разряды кода управления адресом следующей микрокоманды. Для овладения принципом формирования соответствующих кодов необходимо ознакомиться с особенностями адресации управляющей памяти с использованием блока микропрограммного управления (см. рис. 2.4).

Блок микропрограммного управления формирует в цикле выполнения текущей микрокоманды на выходах МА8–МА0 код адреса ячейки управляющей памяти, где хранится код следующей микроинструкции. Управляющая память емкостью в 512 микроинструкций организуется в виде матрицы, состоящей из 32 строк и 16 колонок.

Для адресации строк используется 5-разрядный код МА8—МА4 блока микропрограммного управления. 4-разрядный код МА3—МА0 на выходе этого же блока обеспечивает адресацию столбцов. Адрес следующей микроинструкции полностью определяется кодом на шине управления адресом микрокоманды.

Связь между кодом на шине управления и адресом следующей микроинструкции называется функцией управления адресом.

Функции управления адресом приведены в табл. 2.11 [3]. В таблице приняты следующие буквенно-цифровые обозначения: Y_i , $i = 0, 4$, — данные в i -м разряде шины управления адресом следующей микрокоманды; M_i , $i = 0, 8$, — данные в i -м разряде шины адреса текущей микроинструкции; Φ , C , Z — соответственно содержимое триггеров Φ , C и Z ; PK_i , $i = 0, 3$, — данные в i -м разряде регистра кода команды в блоке микропрограммного управления; K_i , $i = 4, 7$, — данные в i -м разряде шины K .

Как следует из табл. 2.11, блок микропрограммного управления обеспечивает формирование переходов к следующей микроинструкции в соответствии с одиннадцатью функциями переходов, из которых три (JFL, JCF, JZF) являются условными. Адрес следующей ячейки управляющей памяти зависит от кода, поступающего по шине УА6—УА0. Отдельные разряды этого кода являются фиксированными и не могут быть изменены (например, УА6 и УА5 в функции JCC), другие разряды могут быть использованы для записи конкретного кода, обеспечивающего заданный переход. Так, выбором функции JCC обеспечивается переход к одной из любых 32 строк управляющей памяти (число комбинаций 5-разрядного двоичного кода У4—У0 равно 32). Однако адрес колонки следующей микроинструкции не изменяется (код М3—М0 является адресом колонки текущей микроинструкции). Аналогично следует понимать и остальные десять функций управления адресом.

В качестве еще одного примера рассмотрим функцию JFL. Допустим, что код управления адресом УА6—УА0 равен 1001111. Младшие четыре разряда 1111 сформированы программистом. Предположим также, что в разряде М8 записана 1, а в разряде М3 — 0. Определим адрес следующей микроинструкции для $\Phi = 1$ и для $\Phi = 0$. Очевидно, что адрес следующей строки соответствует коду $11111_2 = 31_{10}$. Адрес следующей колонки $0011_2 = 3_{10}$ или $0010 = 2_{10}$ в зависимости от содержимого триггера Φ . Таким образом, содержимое триггера Φ обеспечивает в рассмотренном примере переход к ячейке управляющей памяти, находящейся в третьем или втором столбце тридцать первой строки.

Содержимое триггеров C и Z может быть задано программно с использованием функций управления логикой флажков. Функции управления флажками приведены в табл. 2.12 [53, 197].

Т а б л и ц а 2.12

Название	Обозначение	Код управления флажками по входу		Код управления флажками по выводу	
		3	2	1	0
Установить С- и Z-флажки по состоянию входной линии Φ_{Π}	SCZ	—	—	0	0
Установить Z-флажок по состоянию входной линии Φ_{Π}	STZ	—	—	0	1
Установить С-флажок по состоянию входной линии Φ_{Π}	STC	—	—	1	0
Сохранить С- и Z-флажки	HCZ	—	—	1	1
Установить на выходной линии $\Phi_{\text{В}}$ уровень логического нуля	FFO	0	0	—	—
Выдать на выходную линию $\Phi_{\text{В}}$ содержимое триггера С	FFC	0	1	—	—
Выдать на выходную линию $\Phi_{\text{В}}$ содержимое триггера Z	FFZ	1	0	—	—
Установить на выходной линии $\Phi_{\text{В}}$ уровень логической единицы	FF1	1	1	—	—

Число функций равно восьми. Три функции используются для управления флажками С и Z, две функции служат для выдачи состояния триггеров-флажков С и Z. Три функции обеспечивают сохранение состояния триггеров С и Z, а также выдачу из блока микропрограммного управления сигналов, соответствующих логическим единице и нулю. Состояние триггера Φ изменяется не программно, а в результате выполнения операции, когда, например, имеется изменение данных на линии СПО-СО.

Таким образом, программирование устройств на многокристальных микропроцессорах заключается в разработке программы из микропрограмм. Микропрограммы состояются из микроинструкций. Каждая микроинструкция в случае минимального формата состоит из пяти полей — микрокоманды, маскирования, управления адресом, управления логикой флажков и загрузкой, в каждом из которых должен быть помещен соответствующий код фиксированной разрядности. Минимальная длина микроинструкций равна 21 биту. Очевидно, что программирование многокристальных микропроцессоров в машинных кодах, как и в случае

применения однокристалльных микропроцессоров, не может быть признано рациональным. Определенные удобства при программировании обеспечивает язык программирования микроассемблер, рассматриваемый в следующем подразделе.

Структура команд языков Ассемблер и Микроассемблер. Команды программы, написанные на языке ассемблер, перед загрузкой в память микроЭВМ должны быть преобразованы в команды в машинных кодах. Этот процесс преобразования называется трансляцией или ассемблированием. Трансляция осуществляется с использованием специальной программы-ассемблера. Программа ассемблирования предъявляет определенные требования к конструкции записи (называемой также синтаксисом) команд ассемблера. Эта конструкция одинакова для всех языков ассемблерного уровня.

Команда, написанная на языке ассемблер, может быть условно разделена на четыре части, каждая из которых записывается в своем поле (см. табл. 2.13).

При записи команд на языке ассемблер мнемонику операции отделяют от операнда пробелом. Между операндом и комментарием ставят обычно точку с запятой, а после метки в большинстве ассемблеров ставят двоеточие, но иногда оставляют пробел.

Назначение операций и операндов в командах языка ассемблер дано на стр. 123. Назначение метки поясним следующим примером. Предположим, что имеется часть программы, состоящей из пяти команд микроЭВМ LSI-11.

	АДРЕС	ОПЕРАЦИЯ	ОПЕРАНД
1-я команда	00 02	MOV	# 100, R2
2-я команда	04 06	MOV	# 30, R3
3-я команда	10	NEG	R2
4-я команда	12 14	ADD	20(R3), R2
5-я команда	16 20	BR	[операнд].

Требуется осуществить по команде BR безусловный переход к команде MOV # 30, R3. В команде BR операнд записан в общем

Т а б л и ц а 2.13

	Наименование полей			
	Поле метки	Поле операции	Поле операнда	Поле комментария
Части команды	Метка	Операция	Операнд	Комментарий

виде. Для его конкретизации можно поступить одним из двух способов: записать в явном виде или привести символическое имя адреса перехода.

В первом случае операнд является смещением, которое вычисляется программистом по формуле

$$\Delta = 0,5(\beta - (\alpha + 2)),$$

где α — адрес команды перехода; β — адрес команды, к которой выполняется переход. Адреса записываются в восьмеричном коде. При переходе назад смещение представляется в дополнительном коде.

Следовательно, для безусловного перехода с адреса 16 к адресу 04, по которому расположена команда `MOV # 30, R3`, необходимо в программе записать в явном виде смещение $\Delta = 0,5(04 - (16 + 2)) = 0,5(04 - 20) = -06$. Тогда 5-я команда будет иметь вид `BR -06`.

Вычисление смещения Δ может быть выполнено автоматически в процессе ассемблирования. В этом случае программа ассемблирования (называемая обычно, как и язык программирования, ассемблером) вычисляет адреса переходов и ставит им в соответствии последовательности команд. В связи с этим в программе можно приводить не смещения (или абсолютные адреса для команды `JMP`), а задавать адреса переходов с помощью символических имен-меток.

Таким образом, метка — не число, а имя. Метка ставится после мнемокода операции перехода и записывается в поле метки напротив той команды, к которой должен быть выполнен переход. Программа-ассемблер в процессе трансляции заменяет метку соответствующим абсолютным адресом. Например, в программе

```
MOV # 100, R2
MOV # 30, R3
FORWARD: NEG R2
          ADD 20(R3), R2
          BR FORWARD
```

метка `FORWARD` (ВПЕРЕД) связана с командой `MOV`. В команде `BR FORWARD` программа-ассемблер подставляет смещение Δ вместо символического имени `FORWARD`.

Поле комментария используется для пояснения назначения команды. Это поле используется только для удобства чтения программы и полностью игнорируется ассемблирующей программой при трансляции.

Версией языка ассемблер, предназначенной для микропроцессоров с микропрограммным принципом управления, является микроассемблер.

Программа на языке микроассемблер состоит из микроинструкций, которые могут занимать несколько полей, отведенных под метку, мнемонику операций, мнемонику операнда, код маски, мнемонику функции управления адресом, символическое имя адреса, обозначение функции управления логикой флажков, комментариев и др. [53]. Метка отделяется от мнемоники операции двоеточием. Комментарий начинается и заканчивается знаками /* и */ соответственно. Операнд помещается в круглые скобки. Микроинструкция заканчивается точкой с запятой. В приведенном описании правила соответствуют интерпретации языка XMAS системной программы CROMIS, разработанной для микроЭВМ семейства Intel 3000.

Для примера приведем фрагмент программы на языке микроассемблер:

```
LOOP: SRA (T) 00 JCF(EXIT, STAY) FFZ STZ;
```

/* сдвиг младшего бита множителя в Z, переход по содержимому триггера C к микроинструкции EXIT, если C = 1, иначе переход к микроинструкции STAY, выдача на выход Φ_v Z, установка Z по входу Φ_n */

```
STAY: LMI (R9) 00 JZF(ZERO, ONE) FFZ HCZ;
```

/* увеличение на единицу содержимого счетчика команд и переход после проверки младшего разряда */

```
ONE: ALR (R9, A) 00 JCC(ZERO) FFZ HCZ;
```

/* сложение содержимого R9 с содержимым A, переход к следующей микроинструкции */

```
ZERO: STA (T) 00 JZF(LOOP) FFZ HCZ;
```

/* сдвиг младшего бита множителя в Z, переход к циклу LOOP */.

Системное программное обеспечение. Программа, написанная на языке ассемблера, называется исходной программой или исходным модулем (рис. 2.25). Исходная программа транслируется в объектную, которая затем корректируется и тестируется. Скорректированная и проверенная программа называется рабочей.

Чтобы пользователь имел возможность получить рабочую программу, изготовители серийных микроЭВМ поставляют набор готовых программ, которые образуют системное программное обеспечение. Системное программное обеспечение подразделяют на резидентное и кросс-программное. Резидентное программное обеспечение реализуется в той же микроЭВМ, которая будет использована для обработки прикладных программ. Кросс-програм-

Р и с. 2.25. Состав программного обеспечения пользователя

ное обеспечение содержит служебные программы, предназначенные для создания прикладных программ на ЭВМ другого типа.

Системное программное обеспечение по своему составу может быть различным. Для конкретности рассмотрим системное программное обеспечение микроЭВМ семейства "Электроника-60", используя терминологию работы [31].

В состав системного программного обеспечения, называемого операционной системой, входят программы абсолютного загрузчика, редактора текста, ассемблирования, отладчика, перфоратора и ряд других.

Программа абсолютного загрузчика предназначена для ввода в память микроЭВМ программ операционной системы, исходного, объектного и рабочего модулей, хранящихся на перфоленте, и проверки правильности загрузки программ в память с помощью контрольного суммирования.

Программа абсолютного загрузчика чаще всего размещается в постоянной памяти микроЭВМ. Но она может храниться также и на перфоленте. В этом случае в техническом описании микроЭВМ приводится порядок загрузки соответствующей программы с перфоленты в микроЭВМ.

В качестве примера рассмотрим загрузку абсолютного загрузчика в память микроЭВМ семейства "Электроника-60" (для комплекта в составе микроЭВМ, ЭПМ Консул-260 и фотосчитывающего устройства).

Для загрузки программы в память необходимо выполнить следующие действия:

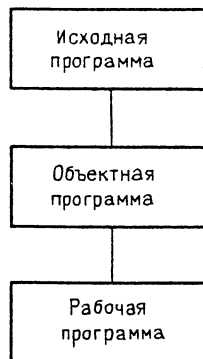
1) включить питание микроЭВМ (пишущая машинка напечатает содержимое ячейки 000024 и знак *);

2) клавишу ПРОГРАММА—ПУЛЬТ перевести в положение ПРОГРАММА;

3) установить перфоленту с программой абсолютного загрузчика кодом 351 на считывающий элемент фотосчитывающего устройства;

4) включить питание фотосчитывающего устройства;

5) напечатать адрес регистра устройства ввода и знак L (177550 L — для высокоскоростного устройства ввода; 177560 L — для низкоскоростного устройства ввода); при этом перфолента



Т а б л и ц а 2.14

Объем адресуемой памяти микроЭВМ	Значения старших разрядов	Объем адресуемой памяти микроЭВМ	Значения старших разрядов
8К	01	40К	11
16К	03	48К	13
24К	05	56К	15
32К	07		

приходит в движение; происходит загрузка программы; по окончании загрузки печатается пусковой адрес

* XX7500

*

где старшие разряды XX определяются из табл. 2.14;

б) напечатать, если требуется, повторный пуск программы абсолютного загрузчика, пусковой адрес * XX7500 и знак G.

Программа редактора текста предназначена для редактирования исходной программы, внесения в нее исправлений и вывода текста программы или его части на перфорацию или печать. Редактирование текста происходит в режиме диалога пользователя с ЭВМ.

Программа редактора текста может храниться в ПЗУ микроЭВМ или вводиться в память с перфоленты. Ввод программы с перфоленты осуществляется под управлением абсолютного загрузчика.

В процессе редактирования программист имеет возможность делать вставки, стирать знаки и строки, выполнять замену знаков, строк и т.д.

Режим редактирования начинается с начального диалога программиста с микроЭВМ, в результате которого в микроЭВМ вводится информация об используемых устройствах ввода-вывода, объеме памяти и т.д.

Первый вопрос *I, печатаемый под управлением редактора, требует ответа

H ↵ — устройство ввода высокоскоростное или

L ↵ — устройство ввода низкоскоростное.

Следующий вопрос *0 требует аналогичных ответов, характеризующих в этом случае скоростные свойства устройства вывода. Здесь символ ↵ означает операцию возврата каретки электрической пишущей машинки.

При использовании низкоскоростного перфоленточного устройства редактор задает вопрос: НПР ВЫКЛ?, который напоминает о необходимости выключить это устройство. После этого необходимо нажать клавишу ↵. Редактор напечатает *, что указывает на готовность редактора к приему команд.

При использовании высокоскоростного перфоленточного устройства вывода редактор сразу после ответа на второй вопрос печатает *.

Исправление ошибок при вводе программы в память осуществляется по специальным командам:

1) команда \backslash , формируемая нажатием клавиши ЗБ, стирает предыдущий знак текущей строки;

2) команда $\Leftarrow U$, образованная нажатием клавиш СКЛ и U, стирает все знаки текущей строки;

3) команда $\Leftarrow P^*$, получаемая нажатием клавиш СКЛ и P, полностью стирает текущую команду;

4) запуск редактора сначала производится нажатием два раза комбинации клавиш СКЛ и P.

Программа ассемблирования преобразует исходную программу в объектную. При трансляции (ассемблировании) исходной программы ассемблер может выявлять синтаксические ошибки, такие, например, как повторное использование метки, отсутствие операнда и т.п.

Трансляция производится за несколько проходов. При каждом проходе выполняется определенная операция: построение таблицы меток; преобразование меток в абсолютные адреса; замена мнемонических кодов двоичными кодами.

В первом проходе программа-ассемблер формирует список синтаксических ошибок, которые могут быть скорректированы с помощью редактора.

Если ошибки в процессе трансляции не обнаружены, то под действием команд программы-ассемблера может быть вызвана программа-перфоратор, которая управляет процессом выдачи объектной программы на перфоленту. После завершения вывода микроЭВМ переходит в режим ожидания.

Программа ассемблирования микроЭВМ семейства "Электроника-60" названа перемещающим ассемблером. Эта программа обеспечивает задание функций прохода и устройств ввода-вывода с помощью начального диалога, печатает сообщение о синтаксических ошибках и таблицу имен пользователя, создает перемещаемые объектные модули и обеспечивает связь между объектными модулями с помощью глобальных имен, выполняет секционирование программ и обрабатывает участки программы с помощью директив условной трансляции.

Вводу программы, записанной на языке Ассемблер, предшествует начальный диалог, в результате которого устанавливается, какие устройства используются для ввода-вывода. Вопросы перемещающего ассемблера и ответы пользователя приведены в табл. 2.15.

Т а б л и ц а 2.15

Символы, печатаемые ассемблером и пользователем	Функция символа
* <i>S</i>	Какое устройство используется для ввода исходной программы?
* <i>B</i>	Какое устройство используется для вывода объектной программы?
* <i>L</i>	Какое устройство используется для выдачи листинга?
* <i>T</i>	На какое устройство выдается таблица имен пользователя?
<i>T</i>	ЭПМ
<i>L</i>	Низкоскоростное перфоленточное устройство ввода-вывода
<i>H</i>	Высокоскоростное перфоленточное устройство ввода-вывода
<i>P</i>	Высокоскоростное печатающее устройство
/1	Первый проход
/2	Второй проход
/3	Третий проход
/E	Печать ошибок
←≠	Запрещение вывода или при-вводе — низкоскоростное перфоленточное устройство ввода.

Пр и м е ч а н и е: Вопросы программы ассемблирования начинаются со знака * .

Программа перемещающего ассемблера допускает изменение режима работы. Если, например, в ответ на вопрос **B* напечатать *H/3/E*, то ассемблер будет выводить объектный модуль и выдавать сообщения об ошибках при третьем проходе.

Программа ассемблирования в процессе трансляции берет на себя, как следует из изложенного, многие функции программиста, такие, например, как адресация, преобразование чисел и др. Программист, однако, должен давать программе-ассемблеру некоторые уточнения. Например, ассемблирующая программа не знает, в какую ячейку памяти должна быть помещена первая команда программы. Без специальных указаний эта программа не может также установить конец исходной программы.

Программист выдает такого рода информацию программе ассемблирования в директивах ассемблера.

Директива ассемблера не отображается машинным кодом в объектном модуле, а является указанием программе-ассемблеру. Каждая директива программы перемещающего ассемблера начи-

нается с точки и записывается в поле операции. Поле операнда может быть пустым или содержать один или несколько операндов. Перед директивой может стоять метка, а за директивой — следовать комментарий.

Ниже приводится пример нескольких директив ассемблера:

```
. TITLE CONS  
. END  
. SAM = 5  
.      = 410  
. BYTE 47, SAM
```

Директива `.TITLE CONS` присваивает объектному модулю наименование `CONS`. Директива `.END` указывает на логический и физический конец исходной программы. Директива `.BYTE` используется для записи операндов в виде последовательно расположенных байтов.

Точка в четвертой строке примера представляет собой символ счетчика адресов, а знак `=` является оператором прямого присваивания. Следовательно, по директиве `.BYTE 47, SAM` в ячейке памяти с адресом 410 запоминается восьмеричный код 047, а в ячейке 411 — код 005.

Для записи последовательно расположенных слов длиной в два байта используется директива `.WORD`. Язык перемещающего ассемблера микроЭВМ "Электроника-60" включает в себя язык ассемблера, дополненный, в частности, оператором прямого присваивания.

Формат оператора прямого присваивания следующий:

имя = выражение.

Например,

`A = 1`; `A` присваивается 1

`B = CONS`; `B` присваивается значение выражения `CONS`

`C : D = 3`

`E: MOV # 10, ABC`; `D` присваивается 3, меткам `C` и `E` присваивается адрес ячейки памяти, в которой записана команда `MOV`.

Обращение к регистру общего назначения осуществляется через его имя. Правила присвоения имени регистрам общего назначения поясняются следующим примером:

`R0 = %0`; регистру 0 присвоено имя `R0`

`R6 = R0 + 6`; регистру 6 присвоено имя `R6`

`TNE = %2`; регистру 2 присвоено имя `TNE`.

Знак `%` в операторе указывает на обращение к регистру. Если в операнде некоторое число помещено в круглые скобки, то это число определяет номер регистра общего назначения.

Программы редактора текста, ассемблирования позволяют получить объектную программу, не содержащую синтаксических ошибок. Однако в этой программе могут быть логические ошибки, вызванные использованием неправильных команд, нарушением порядка их следования, некорректным использованием меток и т.д.

Поэтому объектную программу следует отладить. Эта работа выполняется с помощью программы отладки.

Программа отладки представляет собой служебную программу, предназначенную для отладки программ пользователя в режиме диалога путем проверки этих программ на ожидаемые результаты. Допускается внесение небольших изменений в программу в процессе отладки.

Программа отладки обычно обеспечивает следующие возможности: 1) индикацию содержимого ячеек памяти; 2) изменение содержимого ячеек памяти; 3) индикацию содержимого регистров; 4) останов по адресу; 5) пошаговый режим.

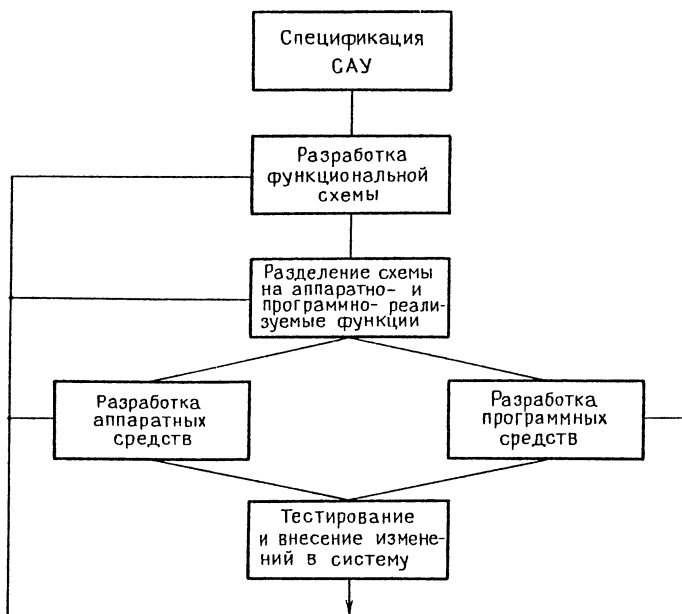
При останове по адресу программист, используя возможности индикации содержимого ячеек памяти и регистров, может проверить правильность работы части программы. Если возникают ошибки, то их можно исправить с помощью средств изменения содержимого памяти. После выполнения нужных операций программист может продолжить работу, выдав соответствующую команду.

В пошаговом режиме можно выполнять программу по одной команде. После выполнения каждой команды программист может проверить результат и внести изменение в программу. Разновидностью пошагового режима является *n*-тактный режим, при котором, задавая значение *n*, программист указывает, какое число последовательных команд должно быть выполнено.

Как только отладка программы полностью завершена, можно передать управление программе перфорации, которая обеспечит вывод рабочей программы на перфоленгу. На этом процесс разработки рабочей программы завершается.

§ 2.4. ЭТАПЫ РАЗРАБОТКИ И ОТЛАДКИ МИКРОЭВМ ДЛЯ САУ

Порядок разработки САУ с микроЭВМ. Разработка САУ с микроЭВМ производится в следующем порядке. Первоначально составляется спецификация системы (рис. 2.26), включающая перечень требований и основных тактико-технических данных разрабатываемой САУ. Спецификацию системы можно в определенной степени отождествить с техническим заданием на разработку. Однако спецификация системы является более узким понятием, так как в



Р и с. 2.26. Порядок разработки САУ с микроЭВМ

ней основная роль отводится алгоритму и вытекающим из него характеристикам и требованиям.

Спецификация системы является исходным материалом для разработки функциональной схемы, которую получают путем разбиения алгоритма функционирования системы на функциональные блоки. Разработке функциональной схемы может предшествовать синтез структурной схемы системы, который может быть выполнен инженерными или математическими методами. Отдельные функциональные блоки могут быть выделены на основе анализа алгоритма работы САУ с учетом возможности реализации операций, входящих в алгоритм, этими блоками.

Выделенные функциональные блоки могут реализовываться как аппаратными, так и программными, т.е. на основе микроЭВМ, средствами. Выбор способа реализации в каждом конкретном случае обосновывается технико-экономически. После выбора способа реализации всех функциональных блоков САУ приступают к разработке программных и аппаратных средств.

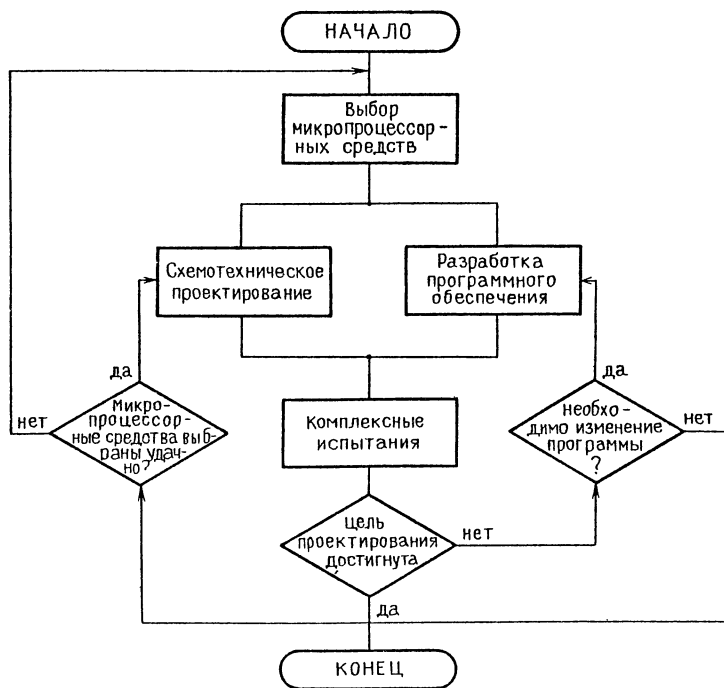
Разработка аппаратных средств связана с детальной проработкой структуры, конструкции функциональных блоков, вопросов их взаимодействия. Эта работа выполняется традиционными методами.

С введением микроЭВМ в контур САУ связана разработка программных средств реализации алгоритмов работы функциональных блоков. Программными средствами реализации алгоритмов являются микроЭВМ, которые с учетом сферы их применения называются управляющими. Разработка управляющих микроЭВМ может выполняться в общем случае параллельно с разработкой аппаратных средств САУ. Однако допустимым является их последовательное проектирование, так как функциональные блоки, реализуемые аппаратно и программно, связаны между собой только входными и выходными сигналами и временными соотношениями между ними.

После разработки аппаратных и программных средств осуществляются настройка и отладка всей системы в целом. При необходимости в систему вносятся изменения.

Обычно разработка САУ завершается разработкой документации, выпуском опытной партии и началом серийного производства.

Этапы разработки управляющих микроЭВМ. Основные этапы разработки управляющих микроЭВМ показаны на рис. 2.27. Этими этапами являются выбор микропроцессорных средств, схемотехническое



Р и с. 2.27. Схема алгоритма разработки управляющей микроЭВМ

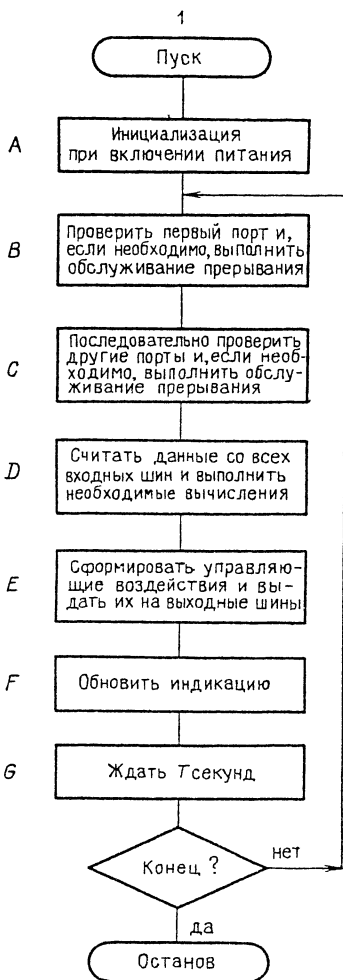
ческое проектирование, разработка прикладного программного обеспечения, комплексные испытания. Каждый из этих этапов может быть подвергнут дальнейшей детализации по содержанию работы, выполняемой на соответствующем этапе.

Например, выбор микропроцессорных средств связан с анализом требований, предъявляемых к проектируемому функциональному узлу САУ на базе микроЭВМ, формированием на этой основе оценочных значений для характеристик микроЭВМ, которая должна быть использована в САУ.

На этом этапе определяются основные ожидаемые технико-экономические характеристики САУ с микроЭВМ, формулируются конструкторско-технологические, климатические и др. требования и ограничения.

Глубокий и всесторонний анализ этих и многих других факторов, зависящих от конкретной области применения и назначения САУ с микроЭВМ, позволяет решить задачу обоснованного выбора микропроцессорных средств.

Следующие два этапа работ выполняются в тесной взаимосвязи. Это обусловлено тем, что показатели эффективности разрабатываемого программного обеспечения зависят от качества схемотехнического проектирования, так как структура микроЭВМ оказывает решающее влияние на ее производительность. И наоборот — система команд и требуемая вычислительная мощность микроЭВМ зачастую предъявляют требования к ее структуре. Наиболее наглядно эта взаимосвязь проявляется при выборе в качестве микропроцессорных средств микропроцессорных комплексов БИС с микропрограммным управлением.



Р и с. 2.28. Пример структурной схемы алгоритма

Схемотехническое проектирование как стадия разработки САУ с микроЭВМ является в большинстве случаев относительно простой задачей, решаемой для выбранных микропроцессорных средств по известным правилам, определенным поставщиками этих средств.

Существенно сложнее, но и, по мнению специалистов, более интересной, требующей напряжения творческих способностей программиста является стадия разработки прикладного программного обеспечения.

Процесс разработки прикладного программного обеспечения выполняется в следующей последовательности: 1) разработка схемы алгоритма; 2) написание текста программы; 3) трансляция (компиляция) программы в коды микропроцессора; 4) отладка программного обеспечения.

Разработка схемы алгоритма и написание текста программы выполняются вручную. Трансляция и отладка программы осуществляются с использованием систем автоматизированного проектирования (САПР).

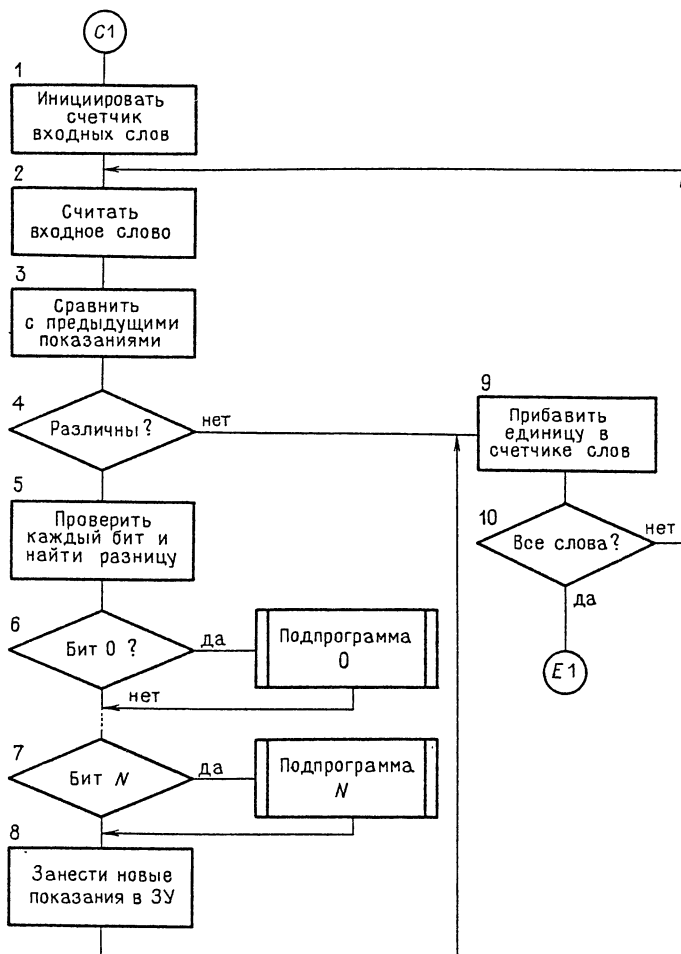
Разработка схем алгоритмов прикладных задач. Алгоритм работы микроЭВМ, реализующей прикладную задачу, обычно представляется в форме трех схем, которые назовем соответственно структурной, функциональной и машинно-ориентированной. Эти названия не являются общепризнанными, но отвечают существу требований, предъявляемых к соответствующим схемам.

Структурная схема представляет собой укрупненную схему алгоритма, указывающую на последовательность функций, которые должны выполняться с помощью программного обеспечения (рис. 2.28) [68–70]. Структурная схема обычно достаточно проста и содержит малое число условных переходов. Она определяет состав основных блоков программного обеспечения.

Функциональная схема содержит подробную схему алгоритма работы основных блоков. Она поясняет принцип их функционирования. При этом указываются конкретные операции, необходимые для достижения желаемого результата. На рис. 2.29 показана функциональная схема алгоритма для блока *D1* (см. рис. 2.28).

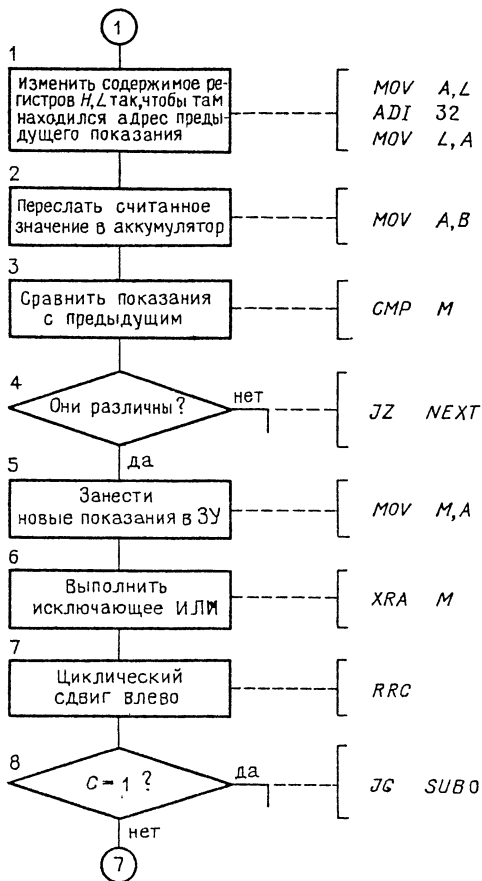
На этом этапе схема алгоритма должна быть представлена как можно более детально, но она еще не привязывается к системе команд микропроцессора. Принято считать, что схема должна содержать приблизительно в 20 раз больше блоков, чем структурная схема алгоритма.

Машинно-ориентированная схема алгоритма является результатом дальнейшей детализации алгоритма. В этой схеме находят отражение системы команд конкретных микропроцессорных средств. Схема алгоритма этого уровня содержит приблизительно в три раза больше блоков, чем функциональная схема. Схема ал-



Р и с. 2.29. Пример функциональной схемы алгоритма

горитма, учитывающая особенности микропроцессорных средств, является наиболее подробной. Она используется для написания программ как на ассемблере, так и на языках высокого уровня. Каждый блок этой схемы соответствует, как правило, не более, чем трем командам. На рис. 2.30 показана машинно-ориентированная схема алгоритма, учитывающая структуру и систему команд микропроцессора КР580ИК80, являющаяся детализацией блоков 2 – 6 схемы, приведенной на рис. 2.29. Соответствующий текст программы на языке ассемблер показан в правой части схемы.



Р и с. 2.30 Пример машино-ориентированной схемы алгоритма

Функции и классификация систем автоматизированного проектирования (САПР) управляющих микроЭВМ. Специализация, связанная с решением микроЭВМ конкретной задачи, осуществляется путем разработки прикладной программы, представляемой в системе команд микропроцессора, и записи этой программы в ПЗУ микроЭВМ.

Разработка прикладных программ является одной из наиболее ответственных и трудоемких операций в цикле проектирования САУ с микроЭВМ. Ее стоимость составляет 50–90 % всей стоимости проектирования, а стоимость одной отлаженной строки программного обеспечения приблизительно соответствует стоимости микропроцессора [97].

Для автоматизации процесса написания программ в комплексе с разработкой необходимого аппаратного обеспечения и их совместной отладки используются системы автоматизированного проектирования (САПР). Так как рассматриваемые в дальнейшем САПР предназначены для проектирования микроЭВМ, встраиваемых в системы автоматического управления, то соответствующие системы автоматизированного проектирования будем называть САПР управляющих микроЭВМ.

Функции и возможности САПР управляющих микроЭВМ определяются обычным порядком создания управляющих микроЭВМ (рис. 2.27). Они связаны применительно к этапу схемотехнического проектирования с тестированием микросхем микропроцессорных комплектов, использованием и тестированием стандартных функциональных плат и модулей.

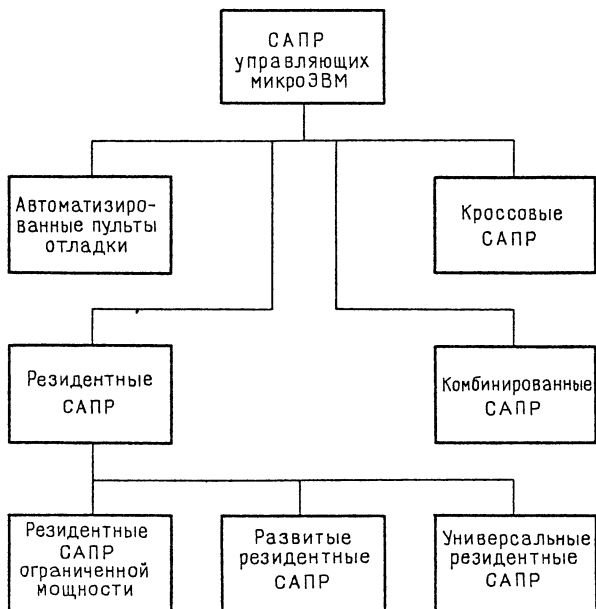
Более разнообразными являются функции САПР на этапе разработки программного обеспечения. Они определяют требования к САПР, связанные с возможностью создавать и редактировать исходные программы, выполнять ассемблирование, диз-ассемблирование, т.е. получение программы на языке ассемблера по машинному коду, и компиляцию, осуществлять интерпретацию программ с помощью эмулятора, моделировать работу внешних устройств, выполнять автономную отладку программного обеспечения и использовать стандартные программные модули.

Комплексные испытания управляющих микроЭВМ требуют реализации динамической (т.е. во времени) совместной отладки программного обеспечения и аппаратных средств, моделирования аппаратных функций микроЭВМ. Желательной на этой стадии является автоматизация программирования ПЗУ.

Полный комплекс работ, связанных с проектированием управляющих микроЭВМ, требует от соответствующих САПР автоматизации также процесса разработки документации, а на стадии опытного и серийного производства – тестирования микросхем, функциональных плат и модулей, проверки работоспособности всей системы в целом. Разработка документации связана с использованием библиотек стандартных модулей, получением листингов программ, распечаткой схемной и конструкторской документации.

Изложенные функции реализуются в различной степени в САПР управляющих микроЭВМ (рис. 2.31).

Наиболее простым путем автоматизации проектирования является создание пульта отладки, являющегося чисто аппаратным средством доступа к внутренним шинам микропроцессора. Пульт отладки обычно предоставляет разработчику возможность управления работой микроЭВМ в режимах пошагового выполнения программы, прерываний, останова по маске и др. Пульт снаб-



Р и с. 2.31. Классификация САПР управляющих микроЭВМ

жается техническими средствами для занесения программы в память в машинных кодах. Перевод программы в машинные коды микропроцессора производится вручную. Пульты отладки обладают ограниченными возможностями, но их достоинствами являются невысокая стоимость, простота и удобство в эксплуатации.

Второй путь автоматизации проектирования микропроцессорных систем связан с использованием кроссовых САПР. Кроссовые САПР предназначены для создания, моделирования и отладки прикладных программ, а также для моделирования и оценки алгоритмов в процессе разработки и эксплуатации систем на базе микроЭВМ.

Широкие программные возможности универсальных ЭВМ, на которых реализуются кроссовые САПР, позволяют создавать прикладные программы с помощью языков высокого уровня. Применение кроссовых средств снижает стоимость и время разработки программного обеспечения. Кроссовые САПР имеют, однако, ряд недостатков, ограничивающих диапазон их применения. Они связаны с трудностями моделирования вычислительной среды, в которой работает микропроцессорная система, с ограниченными возможностями отладки аппаратной части микроЭВМ.

В резидентных системах САПР управляющих микроЭВМ, определяющих третий возможный путь автоматизации проектирования, программное обеспечение САПР реализуется на микроЭВМ тех же типов, которые используются в САУ. В них выполнение разрабатываемой и отлаживаемой программы происходит одновременно с работой САПР. Таким образом, программное обеспечение разрабатывается в реальных или близких к реальным условиям. Однако резидентным САПР также свойственны недостатки, основными из которых являются необходимость использования развитого набора дорогостоящего периферийного оборудования, невозможность использования для прикладных программ любой области памяти микроЭВМ и всех каналов ввода-вывода, что связано с необходимостью поддержания работы САПР. На резидентной САПР сложно отлаживать многопроцессорные вычислительные системы.

Лучшими возможностями автоматизации проектирования микроЭВМ обладают комбинированные САПР. В комбинированных САПР объединяются системные возможности кроссовых САПР и возможности отладки аппаратной части микроЭВМ с помощью резидентных САПР. Совмещение происходит как функциональное, так и временное.

Проектирование с использованием пульта отладки. Системы проектирования этого класса представляют собой одноплатные микроЭВМ с минимальной конфигурацией аппаратного обеспечения. В качестве устройства ввода-вывода используется простейшая клавишная панель с индикаторами, а в ряде случаев — телетайп и дисплей.

Основой пульта отладки могут служить как серийные одноплатные микроЭВМ, так и микроЭВМ, специально спроектированные для этой цели.

Пульты отладки предназначаются для разработки прототипов несложных устройств на базе микроЭВМ с объемом программного обеспечения до одной тысячи слов.

Основными характеристиками таких отладочных средств являются следующие: емкость ПЗУ для хранения системных программ от 256 до 4К слов; емкость ОЗУ, в котором помещаются отлаживаемые программы и данные, от 128 до 4К слов. Как правило, на плате предусматривается интерфейс последовательного и параллельного асинхронного канала ввода-вывода для подключения телетайпа и дисплея, а также до 48 линий параллельного ввода-вывода для обмена информацией с управляемым объектом.

Шины данных и адресные шины микропроцессора, сигналы управления и синхронизации, а также шины интерфейса ввода-вывода выводятся на разъем печатной платы. Это дает возможность

разработчику осуществлять контроль работы микроЭВМ, увеличивать вычислительные ресурсы микроЭВМ.

При использовании систем данного класса программирование микроЭВМ осуществляется в машинных кодах, вручную. Используется шестнадцатеричная или восьмеричная клавиатура. Программное обеспечение пульта отладки ограничивается, как правило, монитором, размещенным в ПЗУ. Такой монитор позволяет реализовать пошаговый режим отладки программы, режим поблочного выполнения программы; вывод информации из регистров общего назначения и ОЗУ на индикацию; обеспечивает работу с клавишным устройством ввода, телетайпом и (или) дисплеем [97].

Работа пульта отладки осуществляется обычно следующим образом. При включении питания системой генерируется сигнал СБРОС, устанавливающий счетчик команд процессора в нулевое состояние. Пошаговый режим выполнения программы реализуется с помощью сигналов ГОТОВНОСТЬ и ОЖИДАНИЕ. Сигнал ГОТОВНОСТЬ является разрешающим для нажатия клавиши ШАГ и выполнения одной команды программы.

Клавиатура содержит наряду с клавишей ШАГ еще восемь командных клавиш: СБРОС, АДРЕС, ПАМЯТЬ, ПЕРЕДАЧА КОДА, РЕГИСТР, СЧЕТ, ОБНУЛЕНИЕ, ТОЧКА РАЗРЫВА. При нажатии клавиш пульта отладки программа-монитор выполняет следующие функции.

1. СБРОС. Выполнение программы прерывается. Счетчик команд обнуляется. В результате работы программы-монитора на индикаторе появляется начальный адрес области памяти для программы пользователя.

2. АДРЕС. После нажатия клавиши АДРЕС программа-монитор выводит на индикатор адрес ЗУ, набираемый на клавиатуре пульта.

3. ПАМЯТЬ. После нажатия клавиши ПАМЯТЬ программа выводит на индикатор информацию, набираемую на клавиатуре пульта.

4. ПЕРЕДАЧА КОДА. Набранная информация переносится в выбранную ячейку ОЗУ. Содержимое счетчика команд увеличивается на единицу. Повторяя операции 3 и 4, можно последовательно ввести в память всю программу.

5. РЕГИСТР. Программа выводит на индикацию номер и содержимое того регистра общего назначения, который выбирается нажатием соответствующей клавиши пульта.

6. ШАГ. Нажимая на эту клавишу, разработчик реализует пошаговый режим выполнения программы с выбранного адреса. Результат операции, выполняемой на каждом шаге, выносится на индикатор.

7. СЧЕТ. Система выполняет программу пользователя. Программа-монитор в этом режиме не используется (если не введена точка останова).

8. ОБНУЛЕНИЕ. При нажатии этой клавиши осуществляется сброс информации, отображенной на индикаторе адреса и данных (кода).

9. ТОЧКА РАЗРЫВА. Указывает адрес контрольного останова. Эта подпрограмма монитора используется в режиме СЧЕТ.

Кроссовые САПР. Кроссовые системы автоматизированного проектирования обычно реализуются на больших и малых ЭВМ, выполненных на процессоре, отличающемся от микропроцессора, для которого разрабатываются на этих ЭВМ программные и аппаратные средства.

Кросс-программное обеспечение САПР включает кросс-компилятор, кросс-ассемблер, кросс-диз-ассемблер, интерпретатор, эмулятор.

Кросс-компилятор — это служебная программа, позволяющая преобразовать исходную программу, написанную на языке высокого уровня, в объектную программу, представленную в машинных кодах микроЭВМ или на языке Ассемблер. Языками высокого уровня могут быть Алгол, Бейсик, Кобол, Фортран, Пл/м, Паскаль и др.

Кросс-ассемблер производит ассемблирование, т.е. преобразование исходной программы в объектную. Кросс-ассемблер делает по существу то же самое, что и кросс-компилятор. Но из-за того, что каждому оператору на языке высокого уровня соответствует большое число команд в объектной программе, кросс-компилятор является более экстенсивным. В процессе компиляции объектная программа всегда получается более длинной, чем в процессе ассемблирования.

Кросс-диз-ассемблер является служебной программой, преобразующей объектную программу из машинного кода в мнемонику языка ассемблер.

Служебной программой является также эмулятор. Эмулятор обеспечивает следующие функции: имитацию характеристик и системы команд конкретного типа микропроцессора; имитацию периферийных устройств той микроЭВМ, для которой создается рабочая программа; возможность отладки объектной программы в режиме реального масштаба времени.

На основании анализа перечня функций, возложенных на эмулятор, следует вывод, что эмулятор позволяет воспроизводить процесс обработки информации, который должен быть реализован в микроЭВМ, на ЭВМ другого типа, но приблизительно так же, как это делает микроЭВМ.

Таким образом, кросс-компилятор и кросс-ассемблер обеспечивают генерацию объектного кода программ, а эмулятор позволяет произвести их отладку.

В настоящее время все основные изготовители микроЭВМ предлагают кросс-системы автоматизированной разработки программного обеспечения. С целью обеспечения машинной независимости такие системы в большинстве случаев написаны на языке Фортран, хотя это не эффективно с точки зрения затрат машинного времени. Так, кросс-ассемблер для микропроцессора I8080, написанный на ассемблере ЭВМ, по времени счета в 20 раз эффективнее, чем кросс-ассемблер, написанный на языке Фортран [97].

Развитое системное программное обеспечение больших ЭВМ, наличие средств ведения библиотеки налаживаемых программ, а также широкий выбор устройств ввода-вывода предоставляют пользователям кросс-систем большие возможности. Однако стоимость эксплуатации кросс-систем на больших ЭВМ весьма высока, поэтому длительное использование этих систем неэкономично.

Малые ЭВМ обладают меньшими возможностями для организации кросс-систем, так как их ресурсы являются существенно более ограниченными. Но удобство и меньшая стоимость эксплуатации делают привлекательными разработку и использование на их основе кроссовых САПР.

Существенным недостатком кросс-систем является невозможность достоверной совместной отладки программного обеспечения и технических средств. Изложенные факторы делают предпочтительным использование САПР на основе резидентных микроЭВМ.

Резидентные системы автоматизированного проектирования. Резидентные системы обеспечивают параллельное проектирование и совместную отладку программного обеспечения и технических средств управляющих микроЭВМ.

Резидентные системы в отличие от систем отладки с использованием пульта имеют больший объем памяти, развитый интерфейс, позволяющий использовать более широкий диапазон устройств ввода-вывода, больший объем системного программного обеспечения.

Поставляются они в виде многоплатных микроЭВМ (например, "Электроника С5-02", "Электроника К1-10", "Электроника-60" и др.), имеющих пульт, электрическую пишущую машинку или дисплей, фотосчитывающее устройство, перфоратор и др. периферийное оборудование. Так же, как и в пульте отладки, ядро системы составляет микроЭВМ на основе комплекта БИС, который будет применяться в разрабатываемой управляющей микроЭВМ.

Запоминающее устройство состоит из ПЗУ емкостью 1—4К слов с возможностью расширения до 16К слов и ОЗУ объемом от 4 до 16К слов с возможностью расширения до 64К байтов.

Системное программное обеспечение кроме монитора включает загрузчик, редактор текста, ассемблер, отладчик, программы обслуживания периферийных устройств. Программирование ведется на ассемблере. Программа ассемблирования в простейших системах данного класса обеспечивает получение абсолютного объектного кода, а в более сложных — перемещаемого кода.

В связи с ограниченным объемом ЗУ в постоянной памяти обычно находятся только программа-монитор и программа-загрузчик. Другие программы располагаются на перфоленте и при необходимости поочередно загружаются в ОЗУ с помощью телетайпа или фотосчитывающего устройства. Отлаживаемая программа вводится и редактируется построчно, а обрабатываться, ассемблером может поблочно. Ассемблер обычно является двухпроходным, а для распечатки программы может использоваться третий проход. Поэтому каждую программу на ассемблере следует вводить 2—3 раза.

Существенным шагом вперед по сравнению с телетайпом и фотосчитывающим устройством является подключение к отладочному комплексу ленточного кассетного ПЗУ. При этом ассемблер, редактор текста, программа отладки и др. размещаются в этом запоминающем устройстве, что существенно повышает удобство пользования отладочным комплексом.

Для повышения производительности труда разработчика в состав резидентной САПР в ряде случаев подключается буквенно-цифровой дисплей с клавиатурой, что обеспечивает возможность удобной отладки программ в оперативном режиме.

Дальнейшее развитие резидентных САПР связано с реализацией следующих возможностей: с обеспечением внутрисхемной эмуляции отлаживаемых программ; с программированием на языках высокого уровня; с расширением набора внешних устройств, включая накопители на гибких магнитных дисках (НГМД); с наличием развитой операционной системы.

Внутрисхемная эмуляция представляет собой замену процессора разрабатываемой микроЭВМ отладочным комплексом. Остальные технические средства разрабатываемой микроЭВМ и системы в целом сохраняются. В этом случае отладочный комплекс, являющийся прототипом процессора управляющей микроЭВМ, функционирует вместе с разрабатываемой системой в реальных условиях. Проверка программного обеспечения производится с помощью монитора отладочного комплекса. Отлаживаемые программы могут размещаться как в памяти разрабатываемой микроЭВМ, так и в памяти отладочного комплекса.

Внутрисхемные эмуляторы позволяют реализовать отладку программного обеспечения без дополнительных затрат времени, связанных при работе в обычном мониторинговом режиме с прерыва-

ниями на каждом шаге отлаживаемой программы и переходом к программе-монитору. В режиме внутрисхемной эмуляции управление отладочным комплексом и выполнение программы производятся со скоростью, равной скорости работы проектируемой САУ.

Развитые резидентные САПР обычно предназначены для разработки систем на базе не одного микропроцессора, а семейства схожих микропроцессорных средств.

Высокая эффективность развитых резидентных САПР во многом обусловлена наличием разнообразных внешних устройств, в частности – НГМД. Наличие НГМД существенно увеличивает емкость памяти, что позволяет использовать трансляторы с языков высокого уровня, дает возможность создавать библиотеки стандартных программ и подпрограмм.

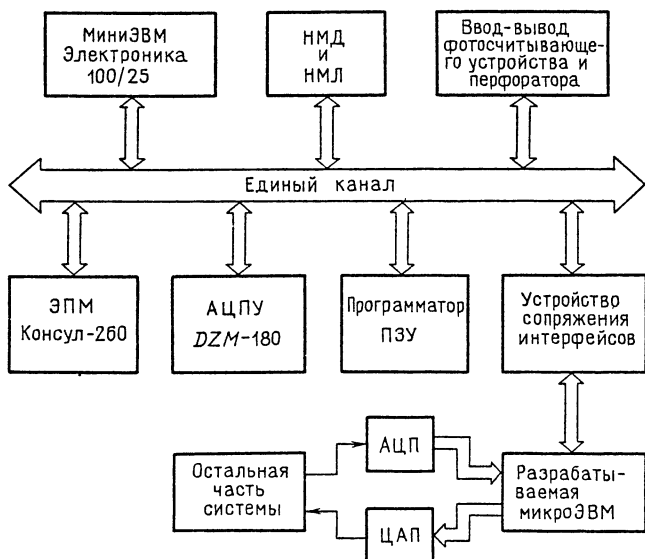
Типичной развитой резидентной САПР является автоматизированное рабочее место, выполненное на основе миниЭВМ "Электроника-100/25" [14].

Недостатком развитых резидентных САПР является их ориентация на конкретный тип микропроцессора. От этого недостатка свободны универсальные резидентные САПР. Для универсальных резидентных систем автоматизированного проектирования характерна их применимость для разработки систем на основе как 8-, так и 16-разрядных микропроцессоров разных типов, использование дорогих внешних устройств, обеспечивающих максимальную производительность САПР, в частности – запоминающих устройств на жестких магнитных дисках.

Технические средства универсальных САПР обычно включают пульт, программатор ППЗУ, память на ленте или диске, перфоленточное устройство ввода-вывода, печатающее устройство, логический тестер, ОЗУ. Блоки внутрисхемных эмуляторов и часть системного программного обеспечения являются сменными. Они предназначены для работы с конкретным микропроцессором.

В универсальных резидентных САПР разработка программного обеспечения обычно производится одновременно несколькими программистами. Поэтому большинство таких САПР включает центральное устройство с быстродействующими ЗУ на жестких магнитных дисках и ряд терминальных устройств, представляющих собой самостоятельные резидентные САПР, использующие в случае надобности ресурсы центрального устройства.

Примером универсальной резидентной САПР может служить система UMDS фирмы Philips [176]. Система предназначена для работы с семью микропроцессорами различных фирм. Она позволяет производить отладку устройств, содержащих до четырех микропроцессоров. При этом обеспечивается работа четырех внутрисхемных эмуляторов одновременно. Программное обеспе-



Р и с. 2.32. Аппаратные средства комбинированной САПР

чение системы состоит из редактора текста, перемещающих ассемблеров, редактора связей и компилятора с языка Паскаль..

Комбинированные САПР. Комбинированные САПР представляют собой двухпроцессорные, а при проектировании мультимикропроцессорных систем – многопроцессорные комплексы, в которых основные функции систем проектирования реализуют миниЭВМ, а микроЭВМ играют роль резидентных средств.

На рис. 2.32 показана структурная схема аппаратных средств комбинированной САПР [56], предназначенной для разработки программного обеспечения, комплексной отладки аппаратной части и программирования ПЗУ микропроцессора КР580ИК80.

Устройство сопряжения интерфейсов является дополнительным блоком, обеспечивающим отладку аппаратной части управляющей микроЭВМ и комплексную отладку программного и аппаратного обеспечения на опытном образце микропроцессорной системы.

Программатор ПЗУ используется для записи отлаженных программ в БИС памяти. Это устраняет трудоемкий процесс перенесения программного обеспечения с промежуточного носителя в память разрабатываемой управляющей микроЭВМ.

Программное обеспечение комплекса реализовано в рамках дисковой операционной системы. Оно содержит программу редактирования текста в диалоговом режиме, программу работы с файлами,

кросс-ассемблер, кросс-компилятор PL/M, редактор связей, эмулятор, отладчик и программу программатора ПЗУ. Сервисное программное обеспечение включает библиотеку объектных модулей, часто применяемых или стандартных микропроцессорных программ, диз-ассемблер и программу распечатки файлов.

Разработка микропроцессорных систем производится на комбинированной САПР поэтапно. Первоначально выполняется разделение функций САУ на реализуемые аппаратно и программно. Разработка аппаратной части САУ сводится к выбору требуемых плат из набора функциональных модулей, обеспечивающих решение возлагаемых на них задач. В некоторых случаях производится разработка дополнительных устройств, выполняющих функции нормализации, фильтрации, согласования по мощности с управляемыми объектами и др. Таким образом, разработка аппаратного обеспечения сводится к модульному проектированию.

Программное обеспечение komponуется также по модульному принципу из библиотеки стандартных модулей (подпрограммы обслуживания АЦП, таймера, дисплея, преобразователей информации, ввода-вывода и т.п.).

Процесс разработки программных средств также разделяется на этапы схемотехнического проектирования, которое выполняется с использованием набора стандартных аппаратных модулей, и раз-

Т а б л и ц а 2.16

Параметры	Пульты отладки	Кроссовые САПР	Резидентные САПР	Комбинированные САПР
Затраты на создание САПР	—	с	+	с
Возможности проверки аппаратной части САУ	с	—	+	+
Возможности разработки программного обеспечения	—	+	+	+
Возможности отладки мультимикропроцессорных систем	—	с	+	+
Простота перестройки под новый тип микропроцессора	—	с	—	+
Моделирование внешних устройств	—	+	с	+
Тестирование функциональных плат	—	—	—	+

Примечание: символом — обозначено низкое значение параметра, с — среднее и + высокое.

работки программного обеспечения. Для этого используются системные программы и библиотека объектных модулей. На начальной стадии работы формируется файл исходного модуля (набор логически связанных данных, определяемых исходной программой). Здесь применяется программа редактирования текста. Кросс-ассемблер или кросс-компиляторы позволяют получить объектную программу, ошибки в которой устраняются с использованием программы редактирования текста.

После завершения компиляции с помощью программы редактирования связей создается загрузочный модуль. Затем осуществляется отладка программы с помощью динамических диалоговых средств. Эта задача решается с использованием программы-отладчика, которая дает возможность использовать аппаратное обеспечение комбинированной САПР и проверить разрабатываемую программу на реальном оборудовании.

Готовый отлаженный загрузочный модуль, являющийся рабочей программой, может быть перенесен на кристалл ППЗУ с помощью программы-программатора ПЗУ.

После записи программы в ПЗУ разработчик может провести окончательное тестирование управляющей микроЭВМ с помощью программы-отладчика на реальной аппаратуре.

Анализ известных САПР управляющих микроЭВМ (табл. 2.16) позволяет сделать вывод, что комбинированные САПР являются наиболее эффективными из существующих систем проектирования. Перспективами развития комбинированных САПР являются разработка и внедрение многопользовательских операционных систем реального времени, обеспечение возможности разработки мультимикропроцессорных систем, применение средств итеративной машинной графики для автоматизации процесса создания документации на аппаратное обеспечение микропроцессорных систем.

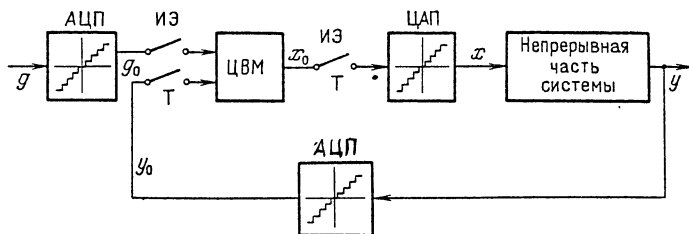
МЕТОДЫ ИССЛЕДОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ С МИКРОЭВМ

§ 3.1. ВВОДНЫЕ ЗАМЕЧАНИЯ

Структура одномерной ЦАС. Общая структура цифровой автоматической системы (ЦАС) изображена на рис. 1.1. Такая система может быть связанной многомерной, если осуществляется управление сложным многомерным объектом, и несвязанной многомерной, если осуществляется управление группой несвязанных одномерных объектов. В последнем случае ЦАС распадается на совокупность одномерных систем.

При управлении сложным объектом или группой объектов процессор обслуживает по очереди отдельные каналы управления. Эта очередь может осуществляться по жесткой программе или по мере поступления заявок от отдельных каналов с возможностью использования в последнем случае приоритетного обслуживания. При жесткой программе обслуживания каналов период дискретности оказывается обычно постоянным, хотя он может быть разным в отдельных каналах. При обслуживании по гибкой программе период дискретности оказывается случайной величиной с некоторым математическим ожиданием и дисперсией.

Основной для исследования обычно является одномерная система. Структурная схема одномерной системы с микроЭВМ изображена на рис. 3.1. Она содержит входные преобразователи непрерывной (аналоговой) величины в код АЦП. Кодироваться может входное задающее воздействие $g(t)$, представляющее собой желаемое значение управляемой величины $y(t)$, сама управляющая



Р и с. 3.1. Одномерная система управления с ЦВМ

величина, а также другая, поступающая извне информация. В результате такого кодирования на вход ЦВМ, а точнее ее процессора, поступают цифровые представления этих величин: g_0 и y_0 .

Дискретность ввода этих величин в процессор иллюстрируют импульсные элементы ИЭ, работающие с периодом дискретности T . Выходная величина процессора x_0 есть цифровое представление сигнала управления. Далее эта величина проходит через преобразователь кода в непрерывную (аналоговую) величину ЦАП и превращается в некоторую величину $x(t)$, представляющую собой в большинстве случаев электрическое напряжение. Эта величина поступает затем на непрерывную часть системы, куда относятся объект управления, исполнительные устройства, усилители, непрерывные корректирующие средства и др.

Задачи, решаемые ЦВМ. Задачи, которые приходится решать ЦВМ в системах автоматического управления, весьма разнообразны. Они могут быть связаны с обработкой поступающей информации, требующей вычислительных или логических операций, с улучшением динамических свойств системы управления за счет введения некоторых корректирующих программ, реализуемых в цифровой части системы, с операциями адаптации и оптимизации системы, с операциями контроля, подготовки к работе, поиска неисправностей и др.

Среди этих задач выделим две, наиболее часто встречающиеся в системах автоматического управления. Первая связана с большим объемом вычислительной работы, требуемой для определения задающего воздействия, которое должно воспроизводиться системой управления. Во многих случаях, особенно в настоящее время, необходимо резкое повышение точности работы систем управления самого различного назначения, и непрерывные вычислительные устройства уже не могут справиться с возросшими требованиями. В этих случаях приходится переходить на использование цифровой вычислительной техники. Задачей ЦВМ оказывается иногда только определение текущей разности между задающим воздействием и управляемой величиной, т.е., по сути дела, ЦВМ работает в этом случае как сравнивающее устройство замкнутой системы управления. Однако выработка этой разности иногда требует производства такого объема вычислений, с которым может справиться не всякая управляющая ЦВМ.

Вторая, наиболее часто возлагаемая на ЦВМ задача, заключается в обеспечении желаемых динамических характеристик системы управления посредством использования корректирующих программ в ЦВМ. В этом случае ЦВМ представляет собой по существу цифровой фильтр с заданными характеристиками.

Методы исследования ЦАС. Цифровые системы управления имеют квантование по времени, что относит их к классу им-

пульсных систем, и квантование по уровню (в АЦП и ЦАП), что делает их нелинейными. Существующие методы исследования нелинейных систем сравнительно мало пригодны для цифровых систем. Методы фазового пространства и фазовой плоскости могут применяться для сравнительно простых систем, обычно, за малым исключением, не имеющих практического значения. Метод гармонической линеаризации [173] оказывается здесь сравнительно сложным вследствие необходимости учета влияния квантования по времени. Осложнение также вызывает непрерывное смещение начальных точек отсчета на нелинейных статических характеристиках входных и выходных преобразователей, вызванное изменяющимися внешними воздействиями. Другие аналитические методы исследования [217] также еще пока малоэффективны.

Поэтому основным методом исследования цифровых систем управления является их моделирование на универсальных цифровых вычислительных машинах и на аналого-цифровых комплексах. Однако такое моделирование не может проводиться без параллельного аналитического исследования, предназначенного для обоснования структуры проектируемой системы, определения основных ее параметров и качественных показателей и предварительного выбора всех элементов. При этом аналитические методы могут предполагать вынесение сложных расчетов на вычислительную технику с целью экономии времени и возможного просмотра большого количества вариантов.

Все это предъявляет к возможным аналитическим методам требования высокой эффективности и обзорности получаемых результатов. Наиболее пригодными здесь оказываются методы расчета, основанные на рассмотрении линеаризованных импульсных систем с учетом влияния, оказываемого квантованием по уровню, в виде дополнительных шумов квантования или в виде дополнительных периодических режимов.

Такая методика принята в дальнейшем изложении. Она основана на том, что при создании новых цифровых систем управления точный расчет динамики с учетом квантования по уровню не представляет особого интереса. Здесь обычно оказывается достаточной лишь оценка дополнительных движений системы и дополнительных ошибок, вызываемых квантованием по уровню.

Многообразие возможных дополнительных режимов в цифровой системе управления и их непрерывный переход от одного вида к другому делает всю картину весьма сложной, плохо поддающейся анализу даже при моделировании подобных систем на универсальных ЦВМ или на аналого-цифровых комплексах.

Развиваемые ниже методы исследования цифровых систем управления позволяют до перехода на моделирование проекти-

руемой системы произвести оценку ожидаемых результатов и решить вопрос о целесообразности построения системы по избранной структуре.

Сведения о других возможных методах аналитического исследования цифровых систем управления имеются в работах [23, 72, 80, 139, 165]. Вопросы исследования нелинейных импульсных систем наиболее систематично и полно изложены в монографии [217].

Линеаризация ЦАС. Применительно к цифровым системам управления обычный метод линеаризации разложением в ряд Тейлора с удержанием только линейного члена оказывается непригодным. Он может использоваться только для линеаризации непрерывной части ЦАС. Распространить его на цифровую часть (ЦВМ с устройствами ввода и вывода) не представляется возможным.

Сама цифровая управляющая машина может реализовать как линейные, так и нелинейные законы управления. В последнем случае линеаризация не должна осуществляться при исследовании, так как при этом будет потеряна весь смысл использования нелинейных законов управления.

Входные и выходные преобразователи (рис. 3.1) имеют статические характеристики ступенчатого (релейного) вида. Для подобных характеристик тангенс угла наклона либо равен нулю, либо стремится к бесконечности. Поэтому для входных и выходных преобразователей ЦВМ используется иной метод линеаризации. Он основан на предположении, обратном тому, которое принимается при линеаризации непрерывных систем. Если принять, что измене-

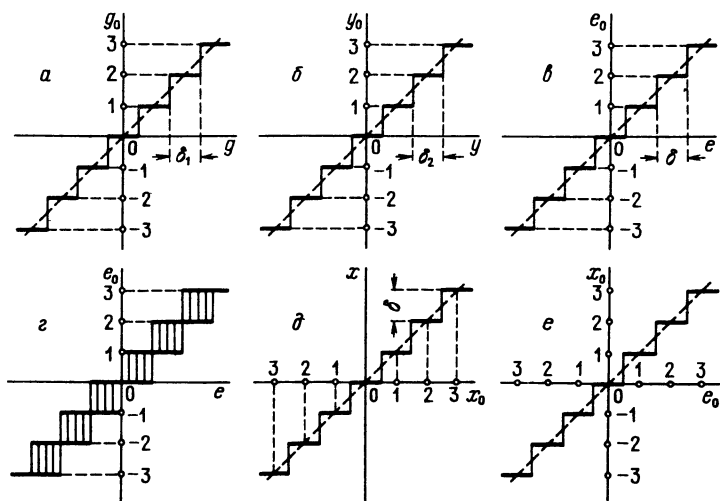
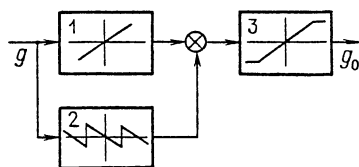


Рис. 3.2. Статические характеристики преобразователей

ния входного сигнала по своей величине значительно больше единицы младшего разряда преобразователя, то можно пренебречь влиянием ступенчатости характеристики и линеаризовать ее, проведя некоторую "среднюю" прямую. Это поясняет рис. 3.2,а, на котором изображена начальная часть статической характеристики входного преобразователя (преобразователя непрерывной величины в код).

По оси абсцисс отложено непрерывное значение задающего воздействия g , а по оси ординат — его цифровое представление (число)



Р и с. 3.3. Эквивалентное представление входного преобразователя

g_0 , получаемое на выходе входного преобразователя. Это число на рис. 3.2,а дано в десятичной системе счисления. Величина единицы младшего разряда на входе обозначена δ_1 . Эта единица младшего разряда имеет физическую размерность, совпадающую с размерностью задающего воздействия. Естественно, что единица младшего разряда на выходе преобразователя равна безразмерной единице.

В дальнейшем изложении будем предполагать, что протяженность всех горизонтальных площадок статической характеристики (рис. 3.2,а) одинакова и равна единице младшего разряда δ_1 . Это означает, что для характеристики справедлива следующая зависимость:

$$g_0 = E[g/\delta_1 + 0,5 \operatorname{sign} g], \quad (3.1)$$

где E — целая часть числа, заключенного в квадратные скобки.

Для линеаризованной характеристики, показанной на рис. 3.2,а штриховой линией, коэффициент передачи

$$k_1 = 1/\delta_1. \quad (3.2)$$

Наибольшая ошибка входного преобразователя при переходе от нелинейной характеристики к линейной не будет превышать по модулю значения $0,5 \delta_1$.

Преобразователь с характеристикой, изображенной на рис.3.2,а, может быть представлен в виде совокупности линейного и нелинейных звеньев (рис. 3.3). Звено 1 является линейным с коэффициентом передачи k_1 , определяемым формулой (3.2). Звено 2 с пилообразной статической характеристикой соответствует нелинейной

добавке, которую дает действительная нелинейная характеристика преобразователя. Наклон каждого "зубца" характеристики равен k_1 . Звено 3 соответствует ограниченно-линейному звену с единичным коэффициентом передачи на линейном участке и насыщением, которое имеет место во всех реальных преобразователях и определяется длиной его разрядной сетки.

Число отличных от нуля уровней одной ветви рассматриваемой характеристики входного преобразователя

$$\mu_1 = 2^{\alpha_1} - 1 = g_{\max} / \delta_1, \quad (3.3)$$

где α_1 — число двоичных разрядов преобразователя (без учета знакового разряда), а g_{\max} — максимальное значение задающего воздействия, соответствующее насыщению преобразователя.

Линеаризация входного преобразователя означает, по сути дела, что из трех звеньев, изображенных на рис. 3.3, рассматривается только звено 1.

Аналогичные рассуждения можно произвести и для входного преобразователя управляемой величины. Его статическая характеристика изображена на рис. 3.2, б. Символом y обозначено непрерывное значение управляемой величины, а символом y_0 — ее цифровое представление. Единице младшего разряда на входе преобразователя соответствует величина δ_2 , имеющая физическую размерность управляемой величины.

Крутизна линеаризованной характеристики

$$k_2 = 1 / \delta_2. \quad (3.4)$$

Число отличных от нуля уровней характеристики на одной ее ветви, если α_2 — число двоичных разрядов преобразователя (без знакового разряда),

$$\mu_2 = 2^{\alpha_2} - 1 = y_{\max} / \delta_2, \quad (3.5)$$

где y_{\max} — максимальное значение управляемой величины, соответствующее насыщению преобразователя.

Обычно используют такие преобразователи, что $\delta_1 = \delta_2$ и $k_1 = k_2$. Однако это условие иногда может и не выдерживаться, особенно в тех случаях, когда в системе имеется несколько задающих и управляемых величин.

Число разрядов входных преобразователей, как правило, довольно велико и может достигать 8 — 32. Так, например, если необходимо измерять угол поворота какой-либо оси с погрешностью, не превышающей 10", то единица младшего разряда должна быть выбрана из условия $\delta_1 \leq 20''$. Тогда для обеспечения измерения в пределах $\pm 180^\circ$ число разрядов в соответствии с (3.5)

должно быть

$$\alpha_1 \geq \log_2 \left(\frac{180 \cdot 60 \cdot 60}{20} + 1 \right) = 3,3 \lg 32401 = 14,2.$$

В тех случаях, когда задачей системы управления является обеспечение равенства $y = g$, принято, что $\delta_1 = \delta_2$, а входные преобразователи для задающего воздействия и управляемой величины могут быть условно объединены в один преобразователь, установленный в канале ошибки $e = g - y$. Характеристика такого преобразователя изображена на рис. 3.2, в. По оси абсцисс отложено непрерывное значение ошибки e , а по оси ординат — ее цифровое представление e_0 . Характеристика справедлива для случая, когда $g = m\delta_1 = \text{const}$, где m — целое число, либо $y = m\delta_1 = \text{const}$. Первый случай обычно вводят в рассмотрение при исследовании свободного движения системы и при исследовании периодических режимов, вызванных квантованием по уровню.

В общем случае зависимость $e_0 = f(e)$ определяет область расположения характеристик, что показано на рис. 3.2, г. Характеристика, изображенная на рис. 3.2, в, представляет, по сути, некоторую среднюю характеристику этой области, определенную для случая $g = m\delta_1$.

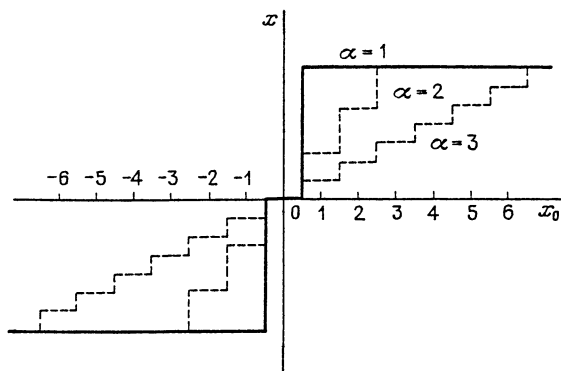
На рис. 3.2, д изображена статическая характеристика выходного преобразователя. По оси абсцисс отложена выходная величина цифровой вычислительной машины (рис. 3.1) в виде числа x_0 , а по оси ординат — величина x , представляющая собой выходную величину преобразователя кода в непрерывную величину совместно с экстраполятором. Обычно выходная величина представляет собой электрическое напряжение или ток. Единица младшего разряда выходной величины преобразователя обозначена δ , а единица младшего разряда входной величины равна безразмерной единице. Их отношение дает крутизну линеаризованной характеристике, т.е. $k = \delta$.

Если число двоичных разрядов выходного преобразователя α , то общее число отличных от нуля уровней одной ветви статической характеристики

$$\mu = 2^\alpha - 1 = x_{\max} / \delta, \quad (3.6)$$

где x_{\max} — максимальное значение выходной величины.

Число разрядов выходного преобразователя обычно бывает меньше, чем число разрядов входного, так как он установлен в канале ошибки, и в пределе может быть равно единице. На рис. 3.4 в качестве примера изображены статические характеристики выходного преобразователя $x = f(x_0)$ для случая, когда макси-



Р и с. 3.4. Примеры статических характеристик выходного преобразователя

мальное значение выходной величины преобразователя x_{\max} одно и то же, но число разрядов $\alpha = 1, 2, 3$.

Для выходного преобразователя может быть получена эквивалентная структурная схема, содержащая три звена, аналогичная изображенной на рис. 3.3.

Приведенные выше формулы (3.3), (3.5) и (3.6) справедливы для симметричных (двухтактных) характеристик. Однако не представляет труда записать их и для случая несимметричных характеристик, когда, например, $|g_{\min}| \neq g_{\max}$, $|y_{\min}| \neq y_{\max}$, $|x_{\min}| \neq x_{\max}$.

Если в цифровой вычислительной машине для установившегося режима получается прямая пропорциональность чисел на входе и выходе, т.е. $x_0 = k_0 e_0$, то машина может рассматриваться как статическое звено с коэффициентом передачи k_0 . Наиболее вероятное значение $k_0 = 1$. Однако возможны случаи, когда $k_0 \neq 1$.

Общий линеаризованный коэффициент передачи машины совместно с входным и выходным преобразователями будет

$$k_{ц} = k_0 k_1 k_2 = k_0 \delta / \delta_1. \quad (3.7)$$

Для этого случая на рис. 3.2,е изображена результирующая статическая характеристика ЭВМ совместно с преобразователями при $k_0 = 1$ в относительном (цифровом) виде, т.е. $x_0 = f(e_0)$, где $x_0 = x / \delta_1$, а $e_0 = e / \delta_1$.

Цифровая машина может сводиться не к статическому, а к интегрирующему звену. Тогда ее линеаризованный коэффициент передачи будет связывать между собой в установившемся режиме входную величину и среднюю скорость изменения выходной величины

(по линейному закону), т.е.

$$(dx/dt)_{\text{ср}} = k'_0 e/T, \quad (3.8)$$

где k'_0 — безразмерный коэффициент.

В этом случае линеаризованный коэффициент передачи ЦВМ совместно с преобразователями

$$k_{\text{ц}} = k'_0 k_1 k_2 = k'_0 \delta / \delta_1. \quad (3.9)$$

Линеаризованная цифровая система управления может рассматриваться как импульсная. При этом учитывается только явление квантования по времени, а влиянием квантования по уровню пренебрегается. Для исследования подобных систем используется аппарат исследования импульсных систем. Однако подобное исследование должно рассматриваться как первое приближение. Предполагается, что в дальнейшем явление квантования по уровню и его влияние будут исследованы дополнительно [23].

§ 3.2. МАТЕМАТИЧЕСКИЙ АППАРАТ

Основные понятия. При исследовании систем управления с ЭВМ широкое использование получили z -преобразование и w -преобразование. Возможность использования этого математического аппарата определяется тем, что все величины, характеризующие состояние объекта, рассматриваются в дискретные моменты времени $t = nT$, где $n = 0, 1, 2, \dots$ — целое число, а T — период дискретности (период квантования во времени).

Таким образом, рассматривается числовая последовательность $f[nT]$ или $f[n]$, называемая также решетчатой функцией.

Дискретные моменты времени можно выбрать с некоторым смещением во времени ϵT , причем $0 \leq \epsilon < 1$. Тогда $t = nT + \epsilon T$. Дискретная последовательность в этом случае имеет вид $f[nT + \epsilon T]$ или $f[n, \epsilon]$.

При расчете систем управления с ЦВМ обычно можно ограничиться случаем $\epsilon = 0$ и рассматривать дискретные последовательности вида $f[n]$.

Вводится понятие разностей решетчатой функции. Прямой разностью первого порядка называется дискретная последовательность (решетчатая функция)

$$\Delta f[n, \epsilon] = f[n + 1, \epsilon] - f[n, \epsilon] \quad (3.10)$$

и обратной разностью первого порядка — последовательность

$$\nabla f[n, \epsilon] = f[n, \epsilon] - f[n - 1, \epsilon]. \quad (3.11)$$

Прямая и обратная разности k -го порядка определяются выражениями

$$\begin{aligned} \Delta^k f[n, \epsilon] &= \Delta(\Delta^{k-1} f[n, \epsilon]) = \\ &= \Delta^{k-1} f[n+1, \epsilon] - \Delta^{k-1} f[n, \epsilon], \end{aligned} \quad (3.12)$$

$$\begin{aligned} \nabla^k f[n, \epsilon] &= \nabla(\nabla^{k-1} f[n, \epsilon]) = \\ &= \nabla^{k-1} f[n, \epsilon] - \nabla^{k-1} f[n-1, \epsilon]. \end{aligned} \quad (3.13)$$

Так, например, обратная разность второго порядка имеет вид

$$\begin{aligned} \nabla^2 f[n, \epsilon] &= \nabla(\nabla f[n, \epsilon] - \nabla f[n-1, \epsilon]) = \\ &= f[n, \epsilon] - 2f[n-1, \epsilon] + f[n-2, \epsilon]. \end{aligned} \quad (3.14)$$

Вводится понятие неполной суммы дискретной последовательности

$$\sigma[n, \epsilon] = \sum_{m=0}^{n-1} f[m, \epsilon] = \sum_{\nu=1}^n f[n-\nu, \epsilon] \quad (3.15)$$

и полной суммы

$$\sigma_0[n, \epsilon] = \sigma[n, \epsilon] + f[n, \epsilon] = \sigma[n+1, \epsilon] = \sum_{m=0}^n f[m, \epsilon]. \quad (3.16)$$

Отличие (3.16) от (3.15) заключается в том, что значение $f[n, \epsilon]$ в момент времени $t = nT + \epsilon T$ также участвует в формировании результата.

Если разности дискретных последовательностей являются аналогами производной непрерывной величины, то суммы представляют собой аналоги интеграла.

Нетрудно показать, что нахождение прямой разности и взятие неполной суммы, а также нахождение обратной разности и взятие полной суммы являются взаимно обратными операциями, т.е.

$$\Delta \sigma[n, \epsilon] = \nabla \sigma_0[n, \epsilon] = f[n, \epsilon]. \quad (3.17)$$

Использование z -преобразования. Вводится понятие z -преобразования дискретной последовательности

$$F(z, \epsilon) = \sum_{n=0}^{\infty} f[n, \epsilon] z^{-n} = Z\{f[n, \epsilon]\}, \quad (3.18)$$

где $z = \exp(pT)$; а $p = c + j\omega$ — оператор преобразования Лашласа.

Если ряд (3.18) сходится, то $F(z, \epsilon)$ представляет изображение оригинала $f[n, \epsilon]$.

Функцию $F(z, \epsilon)$ можно трактовать как изображение функции времени $f(t)$, определяемой в дискретные моменты времени $t = nt + \epsilon T$, т.е.

$$F(z, \epsilon) = Z \{ f(t) \} . \quad (3.19)$$

Изображения некоторых функций времени приведены в Приложении 2.

Для всех непрерывных и решетчатых функций, приведенных в Приложении, предполагается, что они тождественно равны нулю при $t < 0$. В некоторых изображениях использованы полиномы $R_k(z)$, которые могут быть представлены в виде определителя [216]

$$R_k(z) = k! \begin{vmatrix} 1 & 1-z & 0 & \dots & 0 \\ \frac{1}{2!} & 1 & 1-z & \dots & 0 \\ \frac{1}{3!} & \frac{1}{2!} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{k!} & \frac{1}{(k-1)!} & \frac{1}{(k-2)!} & \dots & 1 \end{vmatrix} .$$

Некоторые частные значения этого полинома

$$R_0(z) = 1,$$

$$R_1(z) = 1,$$

$$R_2(z) = z + 1,$$

$$R_3(z) = z^2 + 4z + 1,$$

$$R_4(z) = z^3 + 11z^2 + 11z + 1.$$

Приведем без доказательства основные свойства z -преобразования.

1. Свойство линейности. Если функция времени имеет вид

$$f(t) = \sum_{\nu=1}^N c_{\nu} f_{\nu}(t),$$

то изображение этой функции равно

$$F(z, \epsilon) = \sum_{\nu=1}^N c_{\nu} F_{\nu}(z, \epsilon), \quad (3.20)$$

где $F_{\nu}(z, \epsilon)$ есть изображение $f_{\nu}(t)$.

2. Теорема сдвига. Для функции времени $f(t - \tau)$, где τ/T — положительное число, причем $0 \leq \epsilon < \zeta$ и $f(t - \tau) \equiv 0$ при $t < \tau$, изображение равно

$$Z\{f(t - \tau)\} = z^{-(1+m)}F(z, 1 + \epsilon - \zeta). \quad (3.21)$$

Здесь m — целая, ζ — дробная часть числа τ/T , а функция $F(z, \epsilon)$ определена формулой (3.19).

Если $\zeta \leq \epsilon < 1$, то изображение равно

$$Z\{f(t - \tau)\} = z^{-m}F(z, \epsilon - \zeta). \quad (3.22)$$

В частном случае запаздывание может составлять целое число периодов дискретности, т.е. $\tau = mT$. Тогда

$$Z\{f(t - \tau)\} = z^{-m}F(z, \epsilon). \quad (3.23)$$

3. Изображение разностей. Если k — целое положительное число, то

$$\begin{aligned} Z\{\Delta^k f[n, \epsilon]\} &= \\ &= (z - 1)^k F(z, \epsilon) - z \sum_{\nu=0}^{k-1} (z - 1)^{k-1-\nu} \cdot \Delta^\nu f[0, \epsilon], \end{aligned} \quad (3.24)$$

Если дискретной последовательности $f[n, \epsilon]$ соответствуют нули в первых k точках оси времени, т.е.

$$f[0, \epsilon] = f[1, \epsilon] = \dots = f[k - 1, \epsilon] = 0,$$

то $\Delta^\nu f(0, \epsilon) = 0$ для $\nu = 0, 1, \dots, k - 1$. Тогда формула (3.24) упрощается:

$$Z\{\Delta^k f[n, \epsilon]\} = (z - 1)^k F(z, \epsilon). \quad (3.25)$$

Для обратной разности справедливо выражение

$$Z\{\nabla^k f[n, \epsilon]\} = \left(\frac{z - 1}{z}\right)^k F(z, \epsilon). \quad (3.26)$$

4. Изображение сумм. Если для оригинала изображение определяется формулой (3.18), то для неполной суммы имеем

$$Z\{\sigma[n, \epsilon]\} = Z\left\{\sum_{m=0}^{n-1} f[m, \epsilon]\right\} = \frac{F(z, \epsilon)}{z - 1}. \quad (3.27)$$

Аналогичным образом для полной суммы

$$Z\{\sigma_0[n, \epsilon]\} = Z\left\{\sum_{m=0}^n f[m, \epsilon]\right\} = \frac{z}{z - 1} F(z, \epsilon). \quad (3.28)$$

Для случая k -кратного суммирования

$$Z \{ \sigma^k [n, \epsilon] \} = \frac{F(z, \epsilon)}{(z-1)^k}, \quad (3.29)$$

$$Z \{ \sigma_0^k [n, \epsilon] \} = \left(\frac{z}{z-1} \right)^k F(z, \epsilon).$$

5: Теорема о начальном значении оригинала. Если оригиналу $f[n, \epsilon]$ соответствует изображение $F(z, \epsilon)$, то

$$f[0, \epsilon] = \lim_{z \rightarrow \infty} \frac{z-1}{z} F(z, \epsilon), \quad (3.30)$$

если этот предел существует.

6. Теорема о конечном значении оригинала. Если оригиналу $f[n, \epsilon]$ соответствует изображение $F(z, \epsilon)$, то

$$\lim_{n \rightarrow \infty} f[n, \epsilon] = \lim_{z \rightarrow 1} \frac{z-1}{z} F(z, \epsilon), \quad (3.31)$$

если эти пределы существуют.

7. Изображения решетчатых функций с измененным периодом следования. Пусть рассматривается решетчатая функция с периодом следования дискрет λT , где $\lambda \neq 1$. Тогда на основании (3.18) можно записать при $\epsilon = 0$

$$Z \{ f[n \lambda T] \} = \sum_{n=0}^{\infty} f[n \lambda T] z^{-\lambda n} = F_{\lambda}(z^{\lambda}, \lambda T). \quad (3.32)$$

Из (3.32) следует, что при изменении периода в λ раз необходимо в изображении решетчатой функции $f[n]$ заменить z на z^{λ} и T на λT . Так, например, если рассматривается решетчатая функция $\exp(-\alpha n T)$, то при введении периода λT в соответствии с Приложением 2 изображение будет

$$F_{\lambda}(z^{\lambda}, \lambda T) = Z \{ e^{-\alpha n \lambda T} \} = \frac{z^{\lambda}}{z^{\lambda} - d^{\lambda}} = \frac{z_1}{z_1 - d_1},$$

где $z_1 = z^{\lambda}$ и $d_1 = d^{\lambda}$. На рис. 3.5 построены для этого случая решетчатые функции с исходным периодом следования T (рис. 3.5, а), растянутым периодом при $\lambda > 1$ (рис. 3.5, б) и сжатым периодом при $\lambda < 1$ (рис. 3.5, в).

8. Сумма ординат решетчатой функции. Если абсцисса абсолютной сходимости решетчатой функции отрицательна ($c < 0$), то, положив в (3.18) $p = 0$, имеем

$$F(1) = \lim_{z \rightarrow 1} F(z) = \sum_{n=0}^{\infty} f[n], \quad (3.33)$$

$$F(1, \epsilon) = I_1(\epsilon) = \lim_{z \rightarrow 1} F(z, \epsilon) = \sum_{n=0}^{\infty} f[n, \epsilon].$$

Р и с. 3.5. Изменение периода решетчатой функции

9. Свертка решетчатых функций. Если

$$Z\{f_1[n]\} = F_1(z); \quad Z\{f_2[n]\} = F_2(z),$$

то можно показать, что

$$\begin{aligned} F_1(z)F_2(z) &= Z\left\{\sum_{\nu=0}^n f_1[\nu]f_2[n-\nu]\right\} = \\ &= Z\left\{\sum_{\nu=0}^n f_1[n-\nu]f_2[\nu]\right\}. \end{aligned} \quad (3.34)$$

Эта формула аналогична соответствующему выражению для свертки двух непрерывных функций.

10. Формула обращения. Если известно изображение $F(z, \epsilon)$, то оригинал (числовая последовательность) может быть найден по так называемой формуле обращения

$$f[n, \epsilon] = \frac{1}{j2\pi} \oint F(z, \epsilon) z^{n-1} dz; \quad (3.35)$$

причем интегрирование ведется по окружности с центром в начале координат и радиусом $R > |z_\nu|_{\max}$, где z_ν ($\nu = 1, 2, \dots$) — полюсы функции $F(z, \epsilon)$.

Выражение (3.35) может быть представлено в виде суммы интегральных вычетов функции:

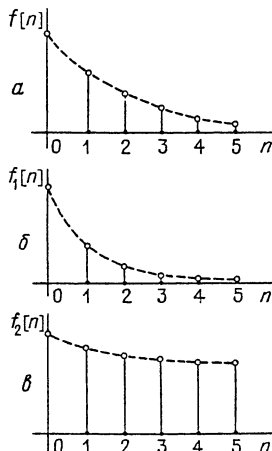
$$f[n, \epsilon] = \sum_{\nu=1}^N \text{Res } F(z, \epsilon) z^{n-1}. \quad (3.36)$$

Однако использование двух последних выражений представляет некоторые трудности. Более простые методы будут рассмотрены ниже в § 3.3.

11. Сумма квадратов дискрет решетчатой функции. Рассмотрим сумму квадратов дискрет решетчатой функции при условии, что абсцисса абсолютной сходимости $c = 0$ и $p = j\omega$. Тогда можно показать, что

$$\sum_{n=0}^{\infty} f^2[n] = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} |F(e^{j\omega T})| d\omega. \quad (3.37)$$

Здесь $F(e^{j\omega T})$ — частотное изображение решетчатой функции, получаемое из z -преобразования подстановкой $z = \exp(j\omega T)$. Выраже-



ние (3.37) представляет собой дискретный аналог формулы Рэлея или, теоремы Парсеваля, записанной для функции времени $f(t)$, отличной от нуля при $t \geq 0$ [26].

Подстановкой

$$(e^{j\omega T} - 1)/(e^{j\omega T} + 1) = j \operatorname{tg} \frac{\omega T}{2} = j\lambda \frac{T}{2},$$

где $\lambda = 2 \operatorname{tg}(\omega T/2)/T$ — абсолютная псевдочастота, или

$$z = e^{j\omega T} = \left(1 + j\lambda \frac{T}{2}\right) \left/ \left(1 - j\lambda \frac{T}{2}\right) \right.,$$

формула (3.37) приводится к виду

$$\sum_{n=0}^{\infty} f^2[n] = \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{F^*(j\lambda) d\lambda}{\left|1 + j\lambda \frac{T}{2}\right|^2}, \quad (3.38)$$

где $F^*(j\lambda)$ — частотное изображение решетчатой функции $f[n]$ при использовании псевдочастоты λ . Выражение (3.38) представляет собой другой вариант дискретного аналога формулы Рэлея.

Интегрирование (3.38) в бесконечных пределах не представляет труда и может быть сделано с использованием известных таблиц интегралов (см. Приложение 3).

Импульсные фильтры. Импульсные и линеаризованные цифровые системы в простейшем виде могут быть представлены в виде последовательного соединения импульсного элемента ИЭ, экстраполятора или формирующей части с передаточной функцией $W_3(p)$ и непрерывной части с передаточной функцией $W_0(p)$. Это показано на рис. 3.6.

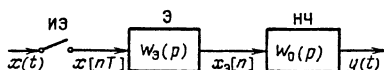


Рис. 3.6. Импульсный фильтр

Поскольку импульсный элемент замыкается в дискретные моменты времени (например, $t = 0, T, 2T, 3T$ и т.д.), то сигнал на входе экстраполятора необходимо рассматривать только в эти моменты времени.

Хотя на выходе непрерывной части сигнал и непрерывен, будем рассматривать его только в эти же дискретные моменты времени.

Таким образом, импульсным фильтром называется устройство, которое получает входные сигналы и одновременно дает выходные сигналы в определенные дискретные моменты времени $t = nT$, где $n = 0, 1, 2, \dots$

Изображения (z -преобразования) входной и выходной величин импульсного фильтра связаны между собой его передаточной

функцией

$$Y(z) = W(z)X(z). \quad (3.39)$$

Для нахождения передаточной функции удобно использовать входную величину в виде единичного одиночного импульса $x[n] = \delta_0[n]$, определяемого зависимостью

$$\begin{aligned} x[n] &= 1 \quad \text{при } n = 0, \\ x[n] &= 0 \quad \text{при } n \neq 0. \end{aligned}$$

В соответствии с (3.18) изображение такого импульса будет $Z\{\delta_0[n]\} = 1$. Реакция импульсного фильтра (непрерывной части совместно с формирующим элементом) на такое входное воздействие называется приведенной весовой функцией фильтра $w_n(t)$.

Поскольку изображение входного сигнала в этом случае равно единице, то изображение выходной величины будет совпадать с передаточной функцией фильтра. Таким образом может быть получена формула для определения передаточной функции импульсного фильтра

$$W(z) = \sum_{n=0}^{\infty} w_n[n] z^{-n}, \quad (3.40)$$

т.е. передаточная функция здесь является z -преобразованием от приведенной весовой функции импульсного фильтра. Это может быть также записано в виде

$$W(z) = Z\{w_n(t)\}. \quad (3.41)$$

Приведенную функцию веса можно представить как обратное преобразование Лапласа от передаточной функции приведенной непрерывной части:

$$w_n(t) = L^{-1}\{W_n(p)\}.$$

В свою очередь передаточная функция приведенной непрерывной части может быть представлена как произведение изображения единственного выходного импульса экстраполятора Э (рис. 3.6) при сформулированном выше условии и передаточной функции непрерывной части НЧ

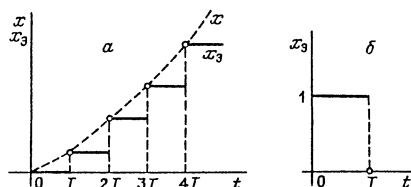
$$W_n(p) = F(p)W_0(p), \quad (3.42)$$

где $F(p)$ – изображение импульса.

Передаточные функции непрерывной части ЦАС. Рассмотрим одномерную цифровую систему (рис. 3.1). При условии, что ЦВМ не осуществляет обработки поступающего на вход разностного сигнала $e_0 = g_0 - y_0$ и в тех случаях, когда рассматриваются линеаризованные входные и выходные преобразователи, передаточная функция разомкнутого канала, связывающая цифровые представ-

ления выходного сигнала y_0 и разности e_0 в соответствии с изложенным выше должна находиться по формуле (3.42). Для этой цели необходимо найти изображение выходного импульса ЦАП при поступлении на вход единичной импульсной функции $\delta_0[n]$.

Цифро-аналоговый преобразователь в данном случае играет роль экстраполятора (рис. 3.6). Различают экстраполяторы нулевого порядка, когда выходной сигнал x_3 удерживается постоянным внутри такта работы ЦВМ, экстраполяторы первого порядка,



Р и с. 3.7. Работа экстраполятора нулевого порядка

когда выходной сигнал внутри такта экстраполируется по линейному закону, экстраполяторы второго порядка — при экстраполировании по параболическому закону и т.д.

Здесь будет рассмотрен только случай экстраполятора нулевого порядка. Сведения об использовании экстраполяторов более высоких порядков содержатся в работе [23].

Работу экстраполятора нулевого порядка поясняет рис. 3.7. Пунктирной линией на рис. 3.7, а показано изменение непрерывной входной величины x на импульсном элементе. Сплошной линией показано изменение выходной величины экстраполятора x_3 . Из рассмотрения рис. 3.7, а вытекает, что импульсный элемент совместно с экстраполятором генерирует прямоугольные импульсы с длительностью, равной периоду дискретности T .

Для того чтобы воспользоваться формулой (3.42), необходимо найти изображение Лапласа единичного прямоугольного импульса (рис. 3.7, б)

$$F(p) = \int_0^T 1 e^{-pt} dt = \frac{1 - e^{-pT}}{p} = \frac{z - 1}{zp}. \quad (3.43)$$

Тогда в соответствии с формулой (3.42) получаем дискретную передаточную функцию непрерывной части совместно с линеаризованными преобразователями

$$W_H(Z) = Z \left\{ \frac{z - 1}{zp} W_0(p) \right\} k_1 k = \frac{\delta}{\delta_1} \frac{z - 1}{z} Z \left\{ \frac{W_0(p)}{p} \right\} \quad (3.44)$$

Здесь $W_0(p)$ — передаточная функция непрерывной части, $k_1 = 1/\delta_1$ — коэффициент передачи входных преобразователей

(АЦП), $k = \delta$ — коэффициент передачи выходного преобразователя (ЦАП), δ_1 и δ — цены разрядов АЦП и ЦАП.

Если в канале управления имеется временное запаздывание $\tau = \zeta T$, то формула (3.44) приобретает вид

$$W_{\text{н}}(p) = \frac{\delta}{\delta_1} \frac{z-1}{z} Z \left\{ \frac{W_0(p)}{p} e^{-\tau p} \right\}. \quad (3.45)$$

Вычисление передаточной функции в этом случае необходимо делать с учетом формулы (3.21). Так, если запаздывание лежит в пределах $0 \leq \tau < T$ или $0 \leq \zeta < 1$, то получим

$$\begin{aligned} W_{\text{н}}(z) &= \frac{\delta}{\delta_1} \frac{z-1}{z^2} Z \{ q_0[n, 1-\zeta] \} = \\ &= \frac{\delta}{\delta_1} \frac{z-1}{z^2} \sum_{n=0}^{\infty} q_0[nT + (1-\zeta)T] z^{-n}, \end{aligned} \quad (3.46)$$

где $q_0(t) = L^{-1} \{ W_0(p)/p \}$ — переходная функция приведенной непрерывной части, равная обратному преобразованию Лапласа от передаточной функции $W_0(p)/p$.

Определим, например, дискретную передаточную функцию непрерывной части ЦАС, когда непрерывная часть имеет передаточную функцию

$$W_0(p) = k_{\text{н}}/p(1 + T_0 p),$$

где $k_{\text{н}}$ — коэффициент передачи, T_0 — постоянная времени.

Общий коэффициент усиления $k = k_{\text{н}} \delta / \delta_1$. Разложим выражение, находящееся в фигурных скобках (3.45), на простые дроби:

$$\frac{W_0(p)}{p} = \frac{k_{\text{н}}}{p^2(1 + T_0 p)} = k_{\text{н}} \left(\frac{1}{p^2} - \frac{T_0}{p} + \frac{T_0^2}{1 + T_0 p} \right).$$

Тогда из (3.45) и Приложения 2 имеем

$$\begin{aligned} W_{\text{н}}(z) &= \frac{k(z-1)}{z} Z \left\{ \frac{1}{p^2} - \frac{T_0}{p} + \frac{T_0^2}{1 + T_0 p} \right\} = \\ &= \frac{k(z-1)}{z} \left[\frac{Tz}{(z-1)^2} - \frac{T_0 z}{z-1} + \frac{T_0 z}{z-d} \right] = \\ &= \frac{k[(T - T_0 + dT_0)z + (1-d)T_0 - dT]}{(z-1)(z-d)}. \end{aligned}$$

Здесь $d = \exp(-T_0/T)$.

Передаточные функции ЦВМ. Передаточная функция ЦВМ представляет собой отношение изображений выходной и входной величин, взятых в цифровой форме. Применительно к рис. 3.1 это

может быть записано в виде

$$D(z) = \frac{X_0(z)}{E_0(z)} = \frac{b_s + b_{s-1}z + \dots + b_0z^s}{a_k + a_{k-1}z + \dots + a_0z^k}. \quad (3.47)$$

Здесь $E_0(z)$ и $X_0(z)$ — изображения (z -преобразования) решетчатых функций $e_0[n] = g_0[n] - y_0[n]$ и $x_0[n]$. Заметим, что всегда должно быть $k \geq s$. Поделим числитель и знаменатель (3.47) на z^k . Тогда для предельного случая $s = k$ имеем

$$\frac{X_0(z)}{E_0(z)} = D(z) = \frac{b_0 + b_1z^{-1} + \dots + b_kz^{-k}}{a_0 + a_1z^{-1} + \dots + a_kz^{-k}}. \quad (3.48)$$

Последнее выражение можно представить в следующем виде:

$$\begin{aligned} & (a_0 + a_1z^{-1} + \dots + a_kz^{-k}) X_0(z) = \\ & = (b_0 + b_1z^{-1} + \dots + b_kz^{-k}) E_0(z). \end{aligned} \quad (3.49)$$

При переходе к оригиналам с учетом теоремы смещения на целое число тактов (3.23) имеем

$$\begin{aligned} & a_0x_0[n] + a_1x_0[n-1] + \dots + a_kx_0[n-k] = \\ & = b_0e_0[n] + b_1e_0[n-1] + \dots + b_ke_0[n-k]. \end{aligned} \quad (3.50)$$

Заметим, что всегда можно привести (3.50) к виду, когда $a_0 = 1$.

Последнее выражение представляет собой разностное уравнение, связывающее две решетчатые последовательности $x_0[n]$ и $e_0[n]$. Из него может быть получен алгоритм работы ЦВМ при $a_0 = 1$

$$\begin{aligned} & x_0[n] = b_0e_0[n] + b_1e_0[n-1] + \dots + b_ke_0[n-k] - \\ & - (a_1x_0[n-1] + a_2x_0[n-2] + \dots + a_kx_0[n-k]). \end{aligned} \quad (3.51)$$

В соответствии с (3.51) должно формироваться текущее значение выходного сигнала ЦВМ $x_0[n]$. При этом, как следует из (3.51), должно использоваться текущее значение входного сигнала $e_0[n]$, а также предыдущие значения входного и выходного сигнала, хранящиеся в памяти ЦВМ.

Таким образом, ЦВМ соответствует некоторому фильтру с передаточной функцией $D(z)$. Свойства этого фильтра будут определяться видом передаточной функции. Если в установленном режиме, т.е. при $z = 1$, $D(1)$ представляет собой конечное число, то такой фильтр будет соответствовать статическому звену.

Если ЦВМ должна обладать интегрирующими свойствами, то знаменатель (3.47) должен содержать множитель $(z - 1)$ в соответствии с формулой (3.29).

Вид необходимой передаточной функции ЦВМ определяется при синтезе ЦАС в соответствии с требованиями по точности и динамическими свойствами проектируемой системы [23, 25]. См. также гл. 5.

Более подробно вопросы построения дискретных фильтров рассмотрены в главе 4.

Передаточные функции ЦАС. В соответствии с изложенным схема линейризованной одномерной ЦАС изображена на рис. 3.8. Передаточная функция разомкнутой системы

$$W(z) = D(z) W_{\text{н}}(z) = D(z) \frac{\delta}{\delta_1} \frac{z-1}{z} Z \left\{ \frac{W_0(p)}{p} \right\} . \quad (3.52)$$

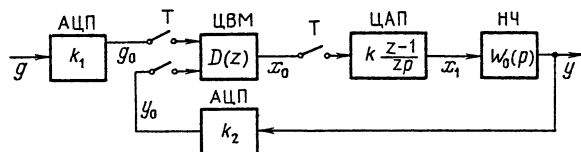
Поскольку при $\delta_1 = \delta_2$ главная обратная связь системы получается единичной, то передаточная функция замкнутой системы будет

$$H(z) = \frac{Y(z)}{G(z)} = \frac{W(z)}{1 + W(z)} . \quad (3.53)$$

Передаточная функция для ошибки

$$H_e(z) = \frac{E(z)}{G(z)} = \frac{1}{1 + W(z)} . \quad (3.54)$$

Здесь $G(z)$, $Y(z)$ и $E(z)$ – изображения задающего воздействия, управляемой величины и ошибки.



Р и с. 3.8. Линейризованная одномерная ЦАС

Схема, изображенная на рис. 3.8, является простейшей. Более сложные схемы рассмотрены в работе [23].

Частотные характеристики. Для получения частотной передаточной функции необходимо в выражении передаточной функции сделать подстановку $z = \exp(j\omega T)$. Так, например, может быть получена частотная передаточная функция разомкнутой системы

$$W(e^{j\omega T}) = W(z) \Big|_{z = \exp(j\omega T)} \quad (3.55)$$

Частотные характеристики (амплитудно-фазовая, амплитудная, фазовая и др.) в этом случае оказываются периодическими функциями частоты ω с периодом $2\pi/T$.

Более удобным для получения частотных характеристик и, в частности, логарифмических частотных характеристик является использование псевдочастоты. Обычно для этой цели применяется так называемое w -преобразование, при помощи которого окружность единичного радиуса $\exp(j\omega T)$ отображается на мни-

мую ось плоскости комплексной величины w . Для преобразования используется подстановка

$$z = \frac{(1 + w)}{(1 - w)} \quad (3.56)$$

или, соответственно,

$$w = \frac{(z - 1)}{(z + 1)}. \quad (3.57)$$

Сделав подстановку $z = \exp(j\omega T)$, получим

$$w = \frac{(e^{j\omega T} - 1)}{(e^{j\omega T} + 1)} = j \operatorname{tg} \frac{\omega T}{2} = j\lambda \frac{T}{2}; \quad (3.58)$$

где $\lambda = 2 \operatorname{tg}(\omega T/2)/T$ представляет собой абсолютную псевдочастоту, которая уже была использована выше при выводе формулы (3.38).

При использовании псевдочастоты построение частотных характеристик оказывается более удобным, так как при изменении круговой частоты в пределах $0 \leq \omega \leq \pi/T$ псевдочастота изменяется в пределах $0 \leq \lambda < \infty$. Это делает частотные характеристики дискретных систем подобными частотным характеристикам непрерывных систем. Заметим, что для малых частот $\omega < 2/T$ псевдочастота практически совпадает с обычной частотой, т.е. $\lambda \approx \omega$. Это оказывается удобным при рассмотрении низкочастотной части характеристик, определяющей точностные показатели системы [23].

Пусть, например, передаточная функция разомкнутой системы имеет вид

$$W(z) = kT/(z - 1),$$

где k — общий коэффициент усиления.

При подстановке $z = \exp(j\omega T)$ будет получена частотная передаточная функция

$$W(e^{j\omega T}) = \frac{kT}{e^{j\omega T} - 1} = -\frac{kT}{2} - j \frac{kT}{2} \operatorname{ctg} \frac{\omega T}{2}.$$

Построение частотных характеристик даже в этом простейшем случае оказывается затруднительным.

Используем подстановку

$$z = (1 + w)/(1 - w) = \left(1 + j\lambda \frac{T}{2}\right) / \left(1 - j\lambda \frac{T}{2}\right).$$

Тогда получим

$$W^*(j\lambda) = W\left(\frac{1 + j\lambda \frac{T}{2}}{1 - j\lambda \frac{T}{2}}\right) = \frac{k\left(1 - j\lambda \frac{T}{2}\right)}{j\lambda}$$

Построение частотных характеристик по последнему выражению не представляет никаких трудностей.

§ 3.3. ИССЛЕДОВАНИЕ ДЕТЕРМИНИРОВАННЫХ ПРОЦЕССОВ

Построение переходных процессов. Если передаточные функции ЦАС известны и известно входное воздействие, то могут быть найдены изображение управляемой величины (рис. 3.8)

$$Y(z) = H(z)G(z) = \frac{W(z)}{1 + W(z)} G(z) \quad (3.59)$$

и изображение ошибки

$$E(z) = H_e(z)G(z) = \frac{1}{1 + W(z)} G(z). \quad (3.60)$$

Здесь $G(z)$ — изображение известного входного воздействия $g[n]$.

Рассмотрим методику нахождения оригинала процесса применительно к управляемой величине.

Если изображение представляет собой простейшую табличную форму (см., например, Приложение 2), то переход к оригиналу не представляет трудностей. Сложная дробно-рациональная форма может быть представлена в виде суммы дробей первой степени. Рассмотрим некоторые разновидности формулы разложения.

1. Пусть изображение $Y(z)$ представляет собой отношение двух многочленов:

$$Y(z) = \frac{A(z)}{B(z)} = \frac{zA_0(z)}{B(z)},$$

причем будем предполагать, что степень числителя не выше, чем степень знаменателя, а корни знаменателя простые. Тогда изображение можно представить в виде суммы

$$Y(z) = \frac{zA_0(z)}{B(z)} = \sum_{\nu=1}^l \frac{A_0(z_\nu)}{\dot{B}(z_\nu)} \frac{z}{z - z_\nu}, \quad (3.61)$$

где $\dot{B}(z)$ — производная $B(z)$ по z , z_ν ($\nu = 1, 2, \dots$) — корни знаменателя. Элементарному слагаемому $z/(z - z_\nu)$ соответствует

оригинал $\exp(-\alpha_\nu nT) = z_\nu^n$, где $\alpha_\nu = \ln(1/z_\nu)/T$. В Приложении 2 единственный корень дроби первой степени обозначен $z_1 = d$.

Поэтому оригинал (3.61) можно записать следующим образом:

$$y[n] = \sum_{\nu=1}^l \frac{A_0(z_\nu)}{\dot{B}(z_\nu)} z_\nu^n. \quad (3.62)$$

2. Пусть изображение $Y(z)$ не имеет нулевого корня числителя, но степень числителя $A(z)$ меньше степени знаменателя. Тогда, как следует из (3.30), начальное значение решетчатой функции $y[0] = 0$. Теперь, если корни знаменателя $Y(z)$ простые, имеем

$$Y(z) = \frac{A(z)}{B(z)} = \frac{1}{z} \sum_{\nu=1}^l \frac{A(z_\nu)}{\dot{B}(z_\nu)} \frac{z}{z - z_\nu}. \quad (3.63)$$

Множитель $1/z$ перед суммой в (3.63) означает запаздывание на один такт. Следовательно, чтобы получить исходную решетчатую функцию, следует в правой части (3.62) сделать сдвиг на один такт вправо, для чего нужно заменить n на $n - 1$. В результате имеем

$$y[n] = \sum_{\nu=1}^l \frac{A(z_\nu)}{\dot{B}(z_\nu)} z_\nu^{n-1}, \quad (3.64)$$

причем последнее выражение будет справедливым только для $n \geq 1$.

3. Пусть изображение Y не имеет нулевого корня числителя $A(z)$, причем степень $A(z)$ равна степени знаменателя [23]. Тогда следует понизить степень числителя, поделив его на знаменатель, и представить $Y(z)$ в виде суммы составляющей нулевого порядка и дробно-рационального остатка $Y_0(z)$. В соответствии с (3.18) первая составляющая равна начальному значению решетчатой функции $y[n]$. Поэтому

$$Y(z) = \frac{A(z)}{B(z)} = y[0] + Y_0(z) = y[0] + \frac{A_0(z)}{B(z)}. \quad (3.65)$$

Переход от второй составляющей изображения к оригиналу может быть сделан по формуле (3.64), которая справедлива для $n \geq 1$.

4. Если изображение $Y(z)$ можно представить в виде некоторой дробно-рациональной функции $Y(z)$, умноженной на изображение единичной ступенчатой функции $1[n]$, которое равно $z/(z - 1)$, т.е.

$$Y(z) = \frac{A(z)}{B(z)} = \frac{z}{z - 1} Y_0(z) = \frac{z}{z - 1} \frac{A_0(z)}{B(z)},$$

то можно показать, что формула разложения приобретает вид

$$y[n] = \frac{A_0(1)}{B_0(1)} - \sum_{\nu=1}^l \frac{A_0(z_\nu)}{(1-z_\nu) \dot{B}_0(z_\nu)} z_\nu^n. \quad (3.66)$$

Последнее выражение представляет собой аналог известной формулы разложения Хевисайда, полученной для непрерывных систем.

5. Для изображения любого вида может быть использовано разложение в ряд Лорана. Из основного выражения для нахождения z -преобразования (3.18) следует:

$$Y(z) = \sum_{n=0}^{\infty} y[n] z^{-n} = y[0] + y[1] z^{-1} + \dots + y[k] z^{-k} + \dots$$

Разложив изображение $Y(z)$ в ряд Лорана (ряд по убывающим степеням z):

$$Y(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + \dots,$$

и сравнивая два ряда между собой, можно установить, что $c_0 = y[0]$, $c_1 = y[1]$, $c_2 = y[2]$, ... и т.д.

Разложение в ряд можно делать любым способом, так как такое разложение единственно. Наиболее удобным приемом для дробно-рациональных функций является деление числителя на знаменатель.

Применяя разложение в ряд Лорана, можно вычислить значения оригинала $y[n]$ или $y[n, \epsilon]$ в дискретных точках без нахождения полюсов изображения $Y(z)$.

Устойчивость ЦАС. Устойчивость замкнутой цифровой системы определяется видом корней характеристического уравнения. На p -плоскости корни устойчивой системы должны лежать в левой ее половине. Переход к комплексной переменной $z = \exp(pT)$ отображает левую полуплоскость во внутреннюю часть круга единичного радиуса с центром в начале координат z -плоскости. Поэтому в устойчивой системе корни характеристического уравнения (знаменателя передаточной функции замкнутой системы, приравненного нулю)

$$1 + W(z) = 0 \quad (3.67)$$

должны лежать внутри круга единичного радиуса, т.е. быть по модулю меньше единицы $|z_\nu| < 1$ для $\nu = 1, 2, \dots, k$, где k — порядок знаменателя передаточной функции замкнутой системы.

Использование известных критериев устойчивости для уравнения (3.67) невозможно. Поэтому целесообразно от комплексной величины z перейти к комплексной величине $w = (z - 1)/(z + 1)$ в соответствии с формулой (3.56). Эта операция отображает внутреннюю часть круга единичного радиуса на левую половину w -плос-

кости, что позволяет использовать известные алгебраические критерии устойчивости для характеристического уравнения

$$1 + W\left(\frac{1+w}{1-w}\right) = 0. \quad (3.68)$$

При переходе к частотной передаточной функции $W(\exp j\omega T)$ или $W^*(j\lambda)$ подстановкой $w = j\lambda T/2$ могут использоваться частотные критерии устойчивости, например критерий Найквиста.

Оценка качественных показателей. Оценку таких качественных показателей, как быстродействие и запас устойчивости, можно делать, используя обычные методы определения этих величин по переходной характеристике [26]. В цифровых системах единичной ступенчатой функции соответствует изображение $z/(z-1)$, которое и следует рассматривать как входное воздействие.

При использовании частотных методов приемы оценки качественных показателей в основном сохраняются. Для этой цели могут быть использованы как частотные передаточные функции разомкнутой системы $W(\exp j\omega T)$ и $W^*(j\lambda)$, так и передаточные функции замкнутой системы $H(\exp j\omega T)$ и $H^*(j\lambda)$. При их использовании могут быть определены такие критерии качества, как запасы устойчивости по амплитуде и по фазе, показатель колебательности, полоса пропускания и др.

Установившаяся точность системы может оцениваться по коэффициентам ошибок. Аналогично непрерывным системам, начиная с некоторого момента времени ошибку импульсной системы регулирования можно представить в виде ряда

$$e[n] = c_0 g[n] + c_1 \dot{g}[n] + \frac{c_2}{2} \ddot{g}[n] + \dots, \quad (3.69)$$

где коэффициенты ошибок c_0 , c_1 , c_2 и т.д. представляют собой коэффициенты разложения передаточной функции для ошибки $H_e(z)$ в ряд Маклорена по степеням p , т.е.

$$c_n = \left[\frac{d^n H_e(e^{pT})}{dp^n} \right]_{p=0}. \quad (3.70)$$

Величины, обратные множителям при производных выражения (3.69), по аналогии с непрерывными системами могут называться соответствующими добротностями. Например, добротности по скорости и по ускорению будут

$$\begin{aligned} k_1 &= 1/c_1; \\ k_2 &= 2/c_2. \end{aligned} \quad (3.71)$$

Вычислим, например, два первых коэффициента ошибок для системы с передаточной функцией разомкнутой цепи

$$W(z) = \frac{kT(1-d)z}{(z-1)(z-d)},$$

где $d = \exp(-T/T_1)$.

Находим передаточную функцию для ошибки:

$$H_e(z) = \frac{1}{1+W(z)} = \frac{(z-1)(z-d)}{(z-1)(z-d) + kT(1-d)z}.$$

Подстановка в это выражение $p=0$ или $z=1$ дает коэффициент $c_0=0$. Для получения коэффициента c_1 находим первую производную:

$$\frac{dH_e(e^{pT})}{dp} = \frac{kT^2(1-d)(z^3 - zd)}{[(z-1)(z-d) + kT(1-d)z]^2}.$$

Подстановка $z=1$ дает коэффициент $c_1=1/k$, а также добротность по скорости $k_1=1/c_1=k$.

§ 3.4. ИССЛЕДОВАНИЕ СЛУЧАЙНЫХ ПРОЦЕССОВ

Статистические характеристики случайных процессов. Для случайного решетчатого процесса $x[n]$ может быть введено понятие корреляционной функции

$$K[n, n_1] = M\{x[n]x[n_1]\}, \quad (3.72)$$

причем процесс предполагается центрированным, т.е. $M\{x[n]\} = 0$. Корреляционная функция дает связь между значениями процесса в моменты времени $t=nT$ и $t_1=n_1T$. Если решетчатый процесс образован дискретизацией некоторого непрерывного процесса, то его корреляционная функция может быть получена дискретизацией корреляционной функции непрерывного процесса

$$K[n, n_1] = K(t, t_1) \Big|_{t=nT, t_1=n_1T}. \quad (3.73)$$

Свойства корреляционной функции решетчатого процесса совпадают со свойствами корреляционной функции непрерывного процесса. Функция обладает симметрией, т.е. $K(n_1, n) = K[n, n_1]$. При равенстве $n_1=n$ корреляционная функция дает дисперсию процесса в момент $t=nT$, т.е. $K[n, n] = D[n]$.

Для стационарных процессов корреляционная функция зависит только от сдвига $m=n_1-n$, т.е.

$$K[m] = M\{x[n]x[n+m]\}. \quad (3.74)$$

Эта функция является четной, т.е. $K[-m] = K[m]$. При $m=0$ получаем дисперсию процесса $K[0] = D$.

Для двух случайных решетчатых процессов $x[n]$ и $y[n]$ может быть введено понятие взаимной корреляционной функции

$$K_{xy}[n, n_1] = \mathbf{M}\{x[n]y[n_1]\}, \quad (3.75)$$

которая показывает связь этих процессов друг с другом. При равенстве нулю этой функции процессы будут некоррелированными.

Для стационарных процессов взаимная корреляционная функция зависит только от сдвига $m = n_1 - n$:

$$K_{xy}[m] = \mathbf{M}\{x[n]y[n+m]\}. \quad (3.76)$$

Вводится понятие спектральной плотности стационарного решетчатого процесса как двустороннего z -преобразования корреляционной функции

$$S(z) = \sum_{m=-\infty}^{\infty} K[m]z^{-m} = F(z) + F(z^{-1}) - K[0], \quad (3.77)$$

где $F(z)$ представляет собой обычное (одностороннее) z -преобразование корреляционной функции.

Аналогично непрерывному случаю можно ввести понятие спектральной плотности как функции круговой частоты

$$\begin{aligned} S(e^{j\omega T}) &= \sum_{m=-\infty}^{\infty} K[m]e^{-j\omega mT} = \\ &= F(e^{j\omega T}) + F(e^{-j\omega T}) - K[0]. \end{aligned} \quad (3.78)$$

При учете четности $K[m]$ и сопряженности комплексных величин $F(e^{j\omega T})$ и $F(e^{-j\omega T})$ имеем

$$\begin{aligned} S(e^{j\omega T}) &= K[0] + 2 \sum_{m=1}^{\infty} K[m] \cos \omega mT = \\ &= 2\operatorname{Re}F(e^{j\omega T}) - K[0]. \end{aligned} \quad (3.79)$$

Формулой обращения для спектральной плотности является преобразование Фурье, вычисляемое на интервале $\pm\pi/T$:

$$\begin{aligned} K[m] &= \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} S(e^{j\omega T}) e^{j\omega mT} d\omega = \\ &= \frac{T}{\pi} \int_{-\pi/T}^{\pi/T} S(e^{j\omega T}) \cos \omega mT d\omega. \end{aligned} \quad (3.80)$$

Если в (3.79) положить $m = 0$, то будет получена дисперсия процесса

$$K[0] = \mathbf{D} = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} S(e^{j\omega T}) d\omega. \quad (3.81)$$

Спектральная плотность может быть представлена как функция псевдочастоты. Так как имеют место равенства

$$e^{j\omega T} = \frac{1 + j\lambda \frac{T}{2}}{1 - j\lambda \frac{T}{2}} ;$$

$$d\omega = \frac{d\lambda}{(1 + j\lambda T/2)(1 - j\lambda T/2)} = \frac{d\lambda}{|1 + j\lambda T/2|^2} ,$$

то можно положить

$$S^*(\lambda) = S\left(\frac{1 + j\lambda T/2}{1 - j\lambda T/2}\right) = 2 \operatorname{Re} F\left(\frac{1 + j\lambda T/2}{1 - j\lambda T/2}\right) - K[0]. \quad (3.82)$$

Формула обращения для $S^*(\lambda)$ приобретает вид

$$K[m] = \frac{T}{2\pi} \int_{-\infty}^{\infty} \left[\frac{1 + j\lambda T/2}{1 - j\lambda T/2} \right]^m \frac{S^*(\lambda) d\lambda}{|1 + j\lambda T/2|^2} =$$

$$= \frac{T}{2\pi} \int_{-\infty}^{\infty} e^{j2m \operatorname{arctg} \lambda T/2} \frac{S^*(\lambda) d\lambda}{|1 + j\lambda T/2|^2} =$$

$$= \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{S^*(\lambda) \cos(2m \operatorname{arctg} \lambda T/2) d\lambda}{|1 + j\lambda T/2|^2} . \quad (3.83)$$

Если в последнем выражении положить $m = 0$, то будет получена формула для определения дисперсии

$$K[m] = D = \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{S^*(\lambda) d\lambda}{|1 + j\lambda T/2|^2} . \quad (3.84)$$

Спектральная плотность $S^*(\lambda)$ удобна тем, что для нахождения интеграла (3.84) возможно использование таблиц интегралов спектральных плотностей непрерывных процессов (см. Приложение 3).

Типовые случайные процессы. Если для непрерывной функции времени $v(t)$, представляющей собой центрированную помеху, эффективное время корреляции

$$\Delta\tau = \frac{1}{K[0]} \int_0^{\infty} K[\tau] d\tau < T, \quad (3.85)$$

то такой процесс может быть представлен как дискретный белый шум с корреляционной функцией

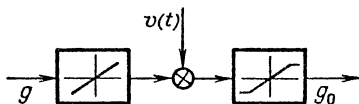
$$K_v[m] = K_v[0] \delta_0[m], \quad (3.86)$$

где $K_v [0] = D_v$ – дисперсия, а $\delta_0 [m]$ – единичная импульсная функция, равная единице при $m = 0$ и равная нулю при $m \neq 0$. Этому белому шуму соответствует спектральная плотность

$$S_v(z) = S_v(e^{j\omega T}) = S_v^*(\lambda) = D_v. \quad (3.87)$$

Дискретному белому шуму соответствует постоянное значение спектральной плотности в пределах изменения частоты $-\infty < \omega < \infty$.

К процессам такого типа обычно сводится помеха, вызываемая эффектом квантования по уровню во входных и выходных преобразователях ЦВМ. Так, входной преобразователь, характеристика



Р и с. 3.9. Эквивалентная схема квантователя по уровню

которого изображена на рис. 3.2, а, осуществляет округление входной величины $g(t)$ в соответствии с формулой (3.1). Максимальное значение ошибки округления составляет здесь $\delta_1/2$. Если считать, что все значения ошибки округления от $-\delta_1/2$ до $\delta_1/2$ равновероятны, то дисперсия ошибки округления составит

$$D_v = K_v [0] = \frac{\delta_1/2}{-\delta_1/2} \int_{-\delta_1/2}^{\delta_1/2} \frac{v^2 dv}{\delta_1} = \frac{\delta_1^2}{12}.$$

Если ввести предположение о независимости ошибок округления в каждом такте опроса, то корреляционная функция и спектральная плотность шума квантования могут быть представлены в виде

$$K_v [m] = K_v [0] \delta_0 [m] = \frac{\delta_1^2}{12} \delta_0 [m], \quad (3.88)$$

$$S_v(z) = S_v(e^{j\omega T}) = S_v^*(\lambda) = \frac{\delta_1^2}{12}.$$

Это позволяет в некоторых случаях заменить структурную схему преобразователя непрерывной величины в код (рис. 3.3) более простой схемой, изображенной на рис. 3.9. Здесь эффект квантования по уровню учитывается в виде шума $v(t)$, корреляционная функция и спектральная плотность которого определяются формулами (3.88)

Аналогичный прием замены нелинейной части преобразователя шумом квантования может применяться для других случаев округления в ЦВМ. Во многих практических случаях корреляционная функция непрерывного центрированного случайного процесса

аппроксимируется формулой $K(\tau) = D_0 \exp(-\mu|\tau|)$, где D_0 — дисперсия. Применительно к решетчатому процессу корреляционная функция должна быть записана в виде $K[m] = D_0 \exp(-\mu T|m|)$.

Двустороннее z -преобразование от этой функции дает спектральную плотность

$$S(z) = \sum_{m=-\infty}^{\infty} D_0 e^{-\mu T|m|} z^{-m} = \frac{D_0(1-d^2)z}{(z-d)(1-dz)}, \quad (3.89)$$

где $d = \exp(-\mu T)$. Подстановка $z = \exp(j\omega T)$ дает спектральную плотность в функции частоты ω :

$$S(e^{j\omega T}) = \frac{D_0(1-d^2)}{1+d^2-2d \cos \omega T}. \quad (3.90)$$

Найдем теперь спектральную плотность рассматриваемого процесса в функции псевдочастоты. Используя формулу (3.82), получаем

$$S^*(\lambda) = \frac{2T_3 D_0(1+\lambda^2 T^2/4)}{T(1+\lambda^2 T_3^2)}, \quad (3.91)$$

где эквивалентная постоянная времени

$$T_3 = \frac{1+d}{1-d} \cdot \frac{T}{2} = \frac{T}{2} \operatorname{cth} \frac{\mu T}{2}.$$

Для гармонического процесса $x(t) = A \sin(\beta t + \varphi)$ с известной амплитудой A и круговой частотой β , но со случайной начальной фазой, лежащей в интервале $-\pi \leq \varphi < \pi$, корреляционная функция оказывается равной $K(\tau) = 0,5A^2 \cos \beta\tau$. Дискретизация ее дает для решетчатого процесса $x[n]$ корреляционную функцию и спектральную плотность

$$K[m] = \frac{A^2}{2} \cos \beta m T = D_0 \cos \beta m T, \quad (3.92)$$

$$S^*(\lambda) = \frac{D\pi}{T} \left(1 + \frac{\lambda^2 T^2}{4}\right) [\delta(\lambda - \lambda_0) + \delta(\lambda + \lambda_0)].$$

Спектральная плотность содержит два бесконечных пика типа δ -функции на частотах $\pm \lambda_0$, где $\lambda_0 = 2 \operatorname{tg}(\beta T/2)/T$.

Расчет ошибок ЦАС. При прохождении случайного процесса через линейное звено с передаточной функцией $W^*(j\lambda)$ для установившегося режима в случае, когда звено является устойчивым, для спектральных плотностей на входе $S_1^*(\lambda)$ и на выходе $S_2^*(\lambda)$ имеет место зависимость

$$S_2^*(\lambda) = |W^*(j\lambda)|^2 S_1^*(\lambda). \quad (3.93)$$

Эта зависимость полностью совпадает с аналогичной зависимостью для непрерывных систем [26].

Если на входе ЦАС действуют полезный случайный сигнал со спектральной плотностью $S_g^*(\lambda)$ и помеха со спектральной плотностью $S_v^*(\lambda)$, то при отсутствии корреляции между полезным сигналом и помехой для спектральной плотности ошибки в установившемся режиме в соответствии с (3.93) имеем

$$\begin{aligned} S_e^*(\lambda) &= |H_e^*(j\lambda)|^2 S_g^*(\lambda) + |H^*(j\lambda)|^2 S_v^*(\lambda) = \\ &= \left| \frac{1}{1 + W^*(j\lambda)} \right|^2 S_g^*(\lambda) + \left| \frac{W^*(j\lambda)}{1 + W^*(j\lambda)} \right|^2 S_v^*(\lambda). \end{aligned} \quad (3.94)$$

Интегрирование этой спектральной плотности в бесконечных пределах в соответствии с (3.84) дает дисперсию ошибки

$$D_e = \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{S_e^*(\lambda) d\lambda}{|1 + j\lambda T/2|^2}. \quad (3.95)$$

Интегрирование (3.95) можно делать в соответствии с Приложением 3.

В более сложных случаях при наличии корреляции между полезным сигналом и помехой формула вычисления спектральной плотности ошибки оказывается более сложной [23].

РЕАЛИЗАЦИЯ НА МИКРОЭВМ ЭЛЕМЕНТОВ И УСТРОЙСТВ СИСТЕМ УПРАВЛЕНИЯ

§ 4.1. ДИФФЕРЕНЦИРОВАНИЕ ЦИФРОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Общие сведения. Рассматривается задача, когда непрерывная функция времени $g(t)$ дискретизируется и представляется в виде решетчатой функции $g[n]$ с периодом T . Так как эта решетчатая функция дается в виде цифровой последовательности, то квантование по времени сопровождается квантованием по уровню, которое приводит к появлению шумов квантования

Задача состоит в нахождении наибольшего приближения к производной непрерывной функции времени $dg(t)/dt$ в цифровой форме $\dot{g}[n]$.

В непрерывных системах оператором дифференцирования является $p = d/dt$. В области изображений оператором дифференцирования будет комплексная величина преобразования Лапласа $p = c + j\omega$.

Для обработки цифровой последовательности $g[n]$ необходимо найти приемлемые алгоритмы дифференцирования, которые могли бы реализоваться в ЦВМ.

Из известного выражения для комплексной величины $z = \exp(pT)$ можно найти оператор дифференцирования $p = (\ln z)/T$. Последнее выражение может быть разложено в ряд тремя способами:

$$p = \frac{1}{T} \left[(z-1) - \frac{1}{2}(z-1)^2 + \frac{1}{3}(z-1)^3 - \dots \right], \quad (4.1)$$

$$p = \frac{2}{T} \left[\frac{z-1}{z+1} + \frac{1}{3} \left(\frac{z-1}{z+1} \right)^3 + \frac{1}{5} \left(\frac{z-1}{z+1} \right)^5 + \dots \right], \quad (4.2)$$

$$p = \frac{1}{T} \left[(1-z^{-1}) + \frac{1}{2} (1-z^{-1})^2 + \frac{1}{3} (1-z^{-1})^3 + \dots \right]. \quad (4.3)$$

Выражение (4.1) может быть использовано в тех случаях, когда будущие значения дифференцируемой функции известны. Это определяется тем, что оператор $z = \exp(pT)$ соответствует опере-

дению на один такт, z^2 — упреждению на два такта и т.д. Поэтому реализовать формулу (4.1) можно при наличии хранения будущих значений функции $g[n]$ в устройствах памяти. При обработке текущей информации, поступающей на вход ЦВМ, формула (4.1) непригодна.

Выражение (4.2) не может быть использовано для целей дифференцирования, так как будет получена неустойчивая программа обработки информации на ЦВМ. Это вытекает из того, что в знаменателе всех членов (4.2) содержится корень $z_1 = -1$, соответствующий границе колебательной устойчивости.

Выражение (4.3) может быть использовано для дифференцирования, так как для его реализации необходимо знать текущее и прошлые значения обрабатываемой функции $g[n]$. Прошлые значения могут храниться в ячейках оперативной памяти ЦВМ. Выражение (4.3) полностью совпадает со случаем использования для целей дифференцирования интерполяционных полиномов Ньютона и Лагранжа [23].

Так как при реализации выражения (4.3) возможно использование только ограниченного конечного числа членов ряда, то вопрос сходимости этого усеченного ряда отпадает. Однако ряд (4.3) должен быть все же сходящимся. Это условие определяется тем, что только при наличии сходящегося ряда можно ограничиться использованием конечного числа его первых членов и получить при этом приемлемую точность дифференцирования. В этом случае ошибка дифференцирования может быть оценена по первому отброшенному члену ряда (4.3). Оценка будет тем точнее, чем быстрее сходится ряд (4.3).

Для определения условий сходимости ряда (4.3) сделаем подстановку $z = \exp(j\omega T)$, где ω — круговая частота дифференцируемого процесса, принимаемого гармоническим. Тогда

$$j\omega = \frac{1}{T} \left[(1 - e^{-j\omega T}) + \frac{1}{2} (1 - e^{-j\omega T})^2 + \frac{1}{3} (1 - e^{-j\omega T})^3 + \dots \right]. \quad (4.4)$$

Достаточное условие сходимости последнего выражения

$$|1 - e^{-j\omega T}| = \left| 2 \sin \frac{\omega T}{2} \right| < 1. \quad (4.5)$$

Отсюда следует условие для наивысшей круговой частоты дифференцируемого процесса

$$\omega_{\max} < \frac{\pi}{3} \cdot \frac{1}{T}, \quad (4.6)$$

где T — период дискретности ЦВМ. Так как $\omega_{\max} = 2\pi f_{\max}$, то

$$f_{\max} < 1/6T. \quad (4.7)$$

Условия (4.6) и (4.7) выполняются практически всегда, так как необходимость дифференцирования процессов с частотой $1/6T$ маловероятна.

Условия сходимости ряда (4.1) совпадают с (4.6) и (4.7), а для ряда (4.2) можно получить $\omega < 1/4T$.

Алгоритмы дифференцирования. Пусть в качестве дифференцируемой функции рассматривается задающее воздействие $g[n]$ на входе ЦВМ. Такая задача возникает при реализации комбинированного управления, необходимого для построения инвариантной системы [23, 26]. Тогда в соответствии с (4.3) должны быть определены обратные разности решетчатой функции $g[n]$. Это вытекает из того, что оператор $(1 - z^{-1})$ соответствует взятию первой разности, оператор $(1 - z^{-1})^2$ — второй разности и т.д.

В результате для первой производной при использовании ограниченного числа членов ряда (4.3) имеем

$$\dot{g}[n] = \left(\nabla g[n] + \frac{1}{2} \nabla^2 g[n] + \frac{1}{3} \nabla^3 g[n] + \dots + \frac{1}{m} \nabla^m g[n] \right) / T. \quad (4.8)$$

Для получения второй производной необходимо использовать оператор p^2 . Тогда из (4.8) можно получить, возведя в квадрат правую часть,

$$\begin{aligned} \ddot{g}[n] = & \left(\nabla^2 g[n] + \nabla^3 g[n] + \frac{11}{12} \nabla^4 g[n] + \dots \right. \\ & \left. \dots + \frac{2}{m} \sum_{i=1}^{m-1} \frac{1}{i} \nabla^m g[n] \right) / T^2. \end{aligned} \quad (4.9)$$

Аналогично для третьей производной нужно возвести в куб правую часть (4.8)

$$\ddot{\ddot{g}}[n] = \left(\nabla^3 g[n] + \frac{3}{2} \nabla^4 g[n] + \frac{35}{4} \nabla^5 g[n] + \dots \right) / T^3. \quad (4.10)$$

При реализации алгоритмов дифференцирования удобнее оперировать не с обратными разностями, а со значениями входной функции в дискретные моменты времени nT , $(n-1)T$, ..., $(n-m)T$, хранящимися в ячейках памяти. Их можно получить по формулам перехода от обратных разностей к значениям самой функции

$$\begin{aligned} \nabla g[n] &= g[n] - g[n-1], \\ \nabla^2 g[n] &= g[n] - 2g[n-1] + g[n-2], \\ \nabla^3 g[n] &= g[n] - 3g[n-1] + 3g[n-2] - g[n-3] \text{ и т.д.} \end{aligned} \quad (4.11)$$

Последние формулы могут быть просто записаны при использовании коэффициентов бинома Ньютона в качестве коэффициентов, входящих в правую часть (4.11).

При учете формул (4.11) выражение для первой производной приобретает вид

$$\dot{g}_M[n] = T^{-1} \sum_{i=0}^m (-1)^i \sum_{k=1}^m \frac{1}{k} C_k^i g[n-i] = T^{-1} \sum_{i=0}^m a_i g[n-i], \quad (4.12)$$

$$a_i = (-1)^i \sum_{k=0}^m \frac{1}{k} C_k^i,$$

где C_k^i — биномиальные коэффициенты, причем $C_k^i = 0$, если $i > k$. Формула (4.12) удобна для реализации на ЦВМ.

Аналогичным образом для второй производной может быть получено выражение

$$\begin{aligned} \ddot{g}_M[n] &= T^{-2} \sum_{j=0}^m (-1)^j \sum_{k=2}^m \frac{2}{k} C_k^j \sum_{i=1}^{k-1} \frac{1}{i} g[n-j] = \\ &= T^{-2} \sum_{j=0}^m b_j g[n-j], \end{aligned} \quad (4.13)$$

$$b_j = (-1)^j \sum_{k=2}^m \frac{2}{k} C_k^j \sum_{i=1}^{k-1} \frac{1}{i},$$

где C_k^j — биномиальные коэффициенты, причем $C_k^j = 0$, если $j > k$. Формула (4.13) удобна для реализации на ЦВМ.

Методические ошибки дифференцирования. Рассмотрим методическую ошибку получения первой производной. Пусть имеется случайный стационарный сигнал $g(t)$, для которого известна корреляционная функция $K(\tau) = \mathbf{M}\{x(t)x(t+\tau)\}$. Будем также считать известными корреляционную функцию его первой производной $\mathbf{M}\{\dot{g}(t)\dot{g}(t+\tau)\} = -\ddot{K}(\tau)$ и взаимную корреляционную функцию сигнала и его первой производной $\mathbf{M}\{g(t)\dot{g}(t+\tau)\} = \mathbf{M}\{g(t-\tau)\dot{g}(t)\} = \dot{K}(\tau)$. Ошибка определения производной в дискретные моменты времени $t = nT$ может быть вычислена как разность между действительным значением $\dot{g}[n]$ и машинным значением $\dot{g}_M[n]$, определяемым по формуле (4.12),

$$\delta\dot{g}[n] = \dot{g}[n] - T^{-1} \sum_{i=0}^m a_i g[n-i]. \quad (4.14)$$

Возведем левую и правую части (4.14) в квадрат и определим математическое ожидание, равное среднему квадрату ошибки,

$$\begin{aligned} \sigma_M^2 &= \mathbf{M}\{(\dot{g}[n] - g_M[n])^2\} = -\ddot{K}[0] - \\ &- \frac{2}{T} \sum_{i=0}^m a_i \dot{K}[iT] + \frac{2}{T^2} \sum_{i=0}^{m-j} a_i a_{i+j} K[jT] + \frac{1}{T^2} \sum_{i=0}^m a_i^2 K[0]. \end{aligned} \quad (4.15)$$

Относительная среднеквадратичная ошибка может быть получена делением σ_M на среднеквадратичное значение скорости изменения входного сигнала $\sigma_{\dot{g}}$:

$$\Delta_M = \sigma_M / \sigma_{\dot{g}}. \quad (4.16)$$

Пусть, например, требуется оценить точность вычисления первой производной сигнала типа нерегулярной качки по первой разности ($m = 1$). Корреляционные функции:

$$K(\tau) = D_g e^{-\mu|\tau|} \left(\cos \beta\tau + \frac{\mu}{\beta} \sin \beta|\tau| \right),$$

$$\dot{K}(\tau) = -\frac{D_g}{\beta} (\mu^2 + \beta^2) e^{-\mu|\tau|} \sin \beta|\tau|,$$

$$-\ddot{K}(\tau) = D_g (\mu^2 + \beta^2) e^{-\mu|\tau|} \left(\cos \beta\tau - \frac{\mu}{\beta} \sin \beta|\tau| \right),$$

где β — преобладающая частота, μ — коэффициент нерегулярности, D_g — дисперсия входного сигнала, $D_g(\mu^2 + \beta^2)$ — дисперсия скорости изменения входного сигнала. В соответствии с формулой (4.12) при $m = 1$ получаем $a_0 = 1$ и $a_1 = -1$. Далее находим

$$\dot{g}_M[n] = (a_0 g[n] + a_1 g[n-1]) T^{-1} = (g[n] - g[n-1]) T^{-1}.$$

Дисперсия ошибки из формулы (4.15)

$$\begin{aligned} \sigma_M^2 &= -\ddot{K}[0] - \frac{2}{T} (a_0 \dot{K}[0] + a_1 \dot{K}[T]) + \frac{2a_0 a_1}{T^2} K[T] + \\ &+ \frac{a_0^2 + a_1^2}{T^2} K[0] = -\ddot{K}[0] - \frac{2}{T} (\dot{K}[0] - \dot{K}[T]) + \\ &+ \frac{2}{T^2} (K[0] - K[T]) = D_g (\mu^2 + \beta^2) - \frac{2D_g}{T} \cdot \frac{\mu^2 + \beta^2}{\beta} e^{-\mu T} \sin \beta T + \\ &+ \frac{2D_g}{T^2} \left[1 - e^{-\mu T} \left(\cos \beta T + \frac{\mu}{\beta} \sin \beta T \right) \right]. \end{aligned} \quad (4.17)$$

Полученная формула (4.17) является точной. При выполнении условий $\mu T \ll 1$ и $\beta T \ll 1$ ее можно упростить, разлагая трансцендентные функции в степенные ряды и ограничиваясь членами низших степеней.

Весьма важен случай дифференцирования входного сигнала гармонического вида. К этому случаю могут быть сведены многие практические задачи. Кроме того, здесь получаются весьма простые и легко обозримые формулы. Пусть рассматривается сигнал вида $g = A \sin(\beta t + \psi)$, где амплитуда A и частота β заданы, а ψ пред-

ставляет собой случайную фазу с равномерным законом распределения в интервале от 0 до 2π . Для этого сигнала имеем

$$\begin{aligned}\sigma_g^2 &= D_g = 0,5A^2, & \sigma_{\dot{g}}^2 &= 0,5\beta^2 A^2, \\ K(\tau) &= 0,5A^2 \cos \beta\tau, & \dot{K}(\tau) &= -0,5\beta A^2 \sin \beta\tau, \\ \ddot{K}(\tau) &= -0,5\beta^2 A^2 \cos \beta\tau.\end{aligned}$$

Если $m = 1$, то средний квадрат ошибки дифференцирования можно получить из (4.17) при $\mu = 0$. В результате имеем

$$\sigma_M^2 = \frac{A^2}{2} \left[\beta^2 - \frac{2\beta}{T} \sin \beta T + \frac{2}{T^2} (1 - \cos \beta T) \right] \approx \frac{A^2 \beta^4 T^2}{8}.$$

Далее можно определить

$$\sigma_M = \frac{\beta^2 AT}{2\sqrt{2}}, \quad \sigma_{M\max} = \frac{\beta^2 AT}{2}, \quad \Delta_M = \frac{\sigma_M}{\dot{g}} = \frac{\sigma_{M\max}}{\dot{g}_{\max}} \approx \frac{\beta T}{2}.$$

Если $m = 2$, то соответствующие формулы имеют вид

$$\dot{g}_M[n] = \frac{1}{3} \left(\frac{3}{2} g[n] - 2g[n-1] + \frac{1}{2} g[n-2] \right) T^{-1},$$

$$\begin{aligned}\sigma_M^2 &= -\ddot{K}[0] - (3\dot{K}[0] - 4K[T] + \dot{K}[2T])T^{-1} + \left(\frac{13}{2} K[0] - \right. \\ &\left. - 8K[T] + \frac{3}{2} K[2T] \right) T^{-2} \approx \frac{\beta^6 A^2 T^4}{18},\end{aligned}$$

$$\sigma_M \approx \frac{\beta^3 AT^2}{3\sqrt{2}}, \quad \sigma_{M\max} \approx \frac{\beta^3 AT^2}{3}, \quad \Delta_M = \frac{\beta^2 T^2}{3}.$$

Продолжая рассмотрение для $m > 2$, можно показать, что для произвольного числа m учитываемых обратных разностей в формуле (4.8) методическая погрешность определяется приближенными выражениями

$$\sigma_M^2 \approx \frac{A^2 \beta^{2(m+1)} T^{2m}}{2(m+1)^2}, \quad \Delta_M = \frac{\sigma_M}{\dot{g}} = \frac{\sigma_{M\max}}{\dot{g}_{\max}} \approx \frac{\beta^m T^m}{m+1}. \quad (4.18)$$

Заметим, что эти формулы могут быть получены сразу из рассмотрения первого отброшенного члена в формуле (4.8).

Полученные формулы позволяют выбирать период дискретности T по заданному значению методической ошибки при известном значении числа учитываемых обратных разностей или определять необходимое значение m при заданном значении периода дискретности T . Так, например, если заданы величины Δ_M и m ,

Таблица 4.1

m	1	2	3	4
T, c	0,002	0,055	0,159	0,266

то период дискретности должен удовлетворять условию

$$T \leq \frac{1}{\beta} \sqrt[m]{(m+1) \Delta_M}. \quad (4.19)$$

В таблице 4.1 в качестве иллюстрации приведены требуемые значения периода дискретности при дифференцировании регулярной качки с частотой $\beta = 1$ рад/с и требуемым значением $\Delta_M = 0,001 = 0,1\%$.

Приближенную оценку (4.18) можно распространить на случай сигнала произвольной формы. Пусть непрерывный входной сигнал имеет $m + 1$ производных, а в алгоритме дифференцирования используются обратные разности $\nabla g[n], \dots, \nabla^m g[n]$. Оценим ошибку от отбрасывания не равной нулю обратной разности $\nabla^{m+1} g[n]$. Ошибка дифференцирования

$$\delta \dot{g}[n] \approx \nabla^{m+1} g[n] / (m+1)T.$$

Приближенно можно записать

$$\nabla^{m+1} g[n] \approx g^{(m+1)}[n] T^{m+1}.$$

Тогда

$$\delta \dot{g}[n] \approx \frac{g^{(m+1)}[n]}{m+1} T^m.$$

Возведя в квадрат левую и правую части последнего выражения и переходя к математическому ожиданию, получаем средний квадрат ошибки дифференцирования

$$\sigma_M^2 \approx \frac{\sigma_{m+1}^2 T^{2m}}{(m+1)^2}, \quad \Delta_M = \frac{\sigma_M}{\sigma_{\dot{g}}} \approx \frac{\sigma_{m+1} T^m}{(m+1)\sigma_{\dot{g}}}. \quad (4.20)$$

В последних выражениях σ_{m+1} — среднеквадратичное значение $(m+1)$ -й производной входного сигнала, $\sigma_{\dot{g}}$ — среднеквадратичное значение его первой производной. Подстановка в (4.20) значений

$$\sigma_{m+1}^2 = 0,5\beta^{2(m+1)} A^2,$$

$$\sigma_{\dot{g}}^2 = 0,5\beta^2 A^2,$$

справедливых для гармонического сигнала, дают формулы (4.18).

Перейдем теперь к вопросу получения на ЦВМ второй производной входного сигнала. Алгоритмы для этого случая определяются формулами (4.9) и (4.13).

Рассмотрим методическую ошибку дифференцирования. Как и в случае получения первой производной, если входной сигнал имеет конечное число производных l , то максимальный порядок обратной разности в (4.9) должен быть равен $m = l$. Тогда методическая ошибка, связанная с дискретизацией во времени, будет равна нулю. При $m < l$ появляется методическая ошибка. Для сигналов, описывающих, например, морскую качку, $l \rightarrow \infty$. Поэтому методическая ошибка дифференцирования здесь будет существовать всегда. Представим входной сигнал $g(t)$ в виде случайного стационарного процесса, для которого известны корреляционные функции

$$\mathbf{M}\{g(t)g(t+\tau)\} = K(\tau),$$

$$\mathbf{M}\{g(t)\ddot{g}(t+\tau)\} = -K_1(\tau),$$

$$\mathbf{M}\{\dot{g}(t)\ddot{g}(t+\tau)\} = K_2(\tau).$$

Ошибка определения второй производной может быть найдена как разность между ее действительным значением $\ddot{g}[n]$ и машинным $\ddot{g}_M[n]$, вычисляемым по формуле (4.13):

$$\delta\ddot{g}[n] = \ddot{g}[n] - \sum_{i=0}^m b_i g[n-i] T^{-2}. \quad (4.21)$$

Возведем левую и правую части (4.21) в квадрат и определим математическое ожидание, равное дисперсии ошибки,

$$\begin{aligned} \sigma_M^2 = \mathbf{M}\{(\ddot{g}[n] - \ddot{g}_M[n])^2\} = & K_2[0] + 2T^{-2} \sum_{i=0}^m b_i K_1[iT] + \\ & + 2T^{-4} \sum_{i=0}^{m-j} b_i b_{i+j} K[jT] + T^{-4} \sum_{i=0}^m b_i^2 K[0]. \end{aligned} \quad (4.22)$$

Относительная ошибка может быть найдена делением среднеквадратичного значения σ_M на среднеквадратичное значение второй производной входной величины

$$\Delta_M = \frac{\sigma_M}{\sigma_{\ddot{g}}} = \frac{\sigma_M}{\sqrt{K_2[0]}}. \quad (4.23)$$

Рассмотрим, например, дифференцирование гармонического сигнала $g = A \sin(\beta t + \psi)$ со случайной начальной фазой с использованием только второй разности ($m = 2$). Для этого сигнала $K(\tau) = 0,5A^2 \cos \beta\tau$, $K_1(\tau) = 0,5A^2 \beta^2 \cos \beta\tau$ и $K_2(\tau) = 0,5A^2 \beta^4 \cos \beta\tau$. В соответствии с формулами (4.9) и (4.13) алгоритм дифференцирования для этого случая будет

$$\ddot{g}_M[n] = \nabla^2 g[n] T^{-2} = (g[n] - 2g[n-1] + g[n-2]) T^{-2}.$$

Таким образом, здесь $b_0 = 1$, $b_1 = -2$ и $b_2 = 1$. Подставляя известные значения в (4.22), имеем

$$\sigma_M^2 = \frac{A^2 \beta^4}{2} + \frac{A^2 \beta^2}{T^2} (1 - 2 \cos \beta T + \cos^2 \beta T) + \frac{A^2}{T^4} (-4 \cos \beta T + \cos 2\beta T) + \frac{3A^2}{T^4}. \quad (4.24)$$

Полученная формула (4.24) является точной. При условии, что $\beta T \ll 1$, ее можно значительно упростить, разлагая косинус в ряд и ограничиваясь членами низших степеней. В результате имеем

$$\sigma_M^2 \approx \frac{A^2 \beta^6 T^2}{2}, \quad \sigma_M \approx \frac{A \beta^3 T}{\sqrt{2}}, \quad (4.25)$$

$$\sigma_{M \max} \approx A \beta^3 T, \quad \Delta_M = \frac{\sigma_M}{\sigma_g} = \frac{\sigma_{M \max}}{\dot{g}_{\max}} \approx \beta T.$$

Аналогичным образом для любого значения m можно получить приближенную формулу для относительной ошибки дифференцирования гармонического сигнала

$$\Delta_m \approx \frac{2}{m+1} \sum_{i=1}^m i^{-1} \beta^{m-1} T^{m-1} = \beta^{m-1} T^{m-1} q_1(m). \quad (4.26)$$

В формулу (4.26) введена функция $q_1(m)$, значения которой даны в табл. 4.2.

Т а б л и ц а 4.2

m	2	3	4	5	
$q_1(m)$	1,0	0,917	0,834	0,761	
m	6	7	8	9	10
$q_1(m)$	0,700	0,678	0,630	0,586	0,550

Полученная формула позволяет выбрать период дискретности по заданному значению методической ошибки при известном числе учитываемых обратных разностей. Если задана величина Δ_M , то необходимо выполнить условие

$$T \leq \frac{1}{\beta} \left[\frac{\Delta_M}{q_1(m)} \right]^{\frac{1}{m-1}}. \quad (4.27)$$

Для иллюстрации в табл. 4.3 приведены требуемые значения периода дискретности при двойном дифференцировании гармонического сигнала с частотой $\beta = 1\text{с}^{-1}$ и $\Delta_M = 0,001 = 0,1\%$.

Таблица 4.3

m	2	3	4	5
$T, \text{с}$	0,001	0,039	0,106	0,190

Приближенную оценку точности (4.26) можно распространить на случайный сигнал произвольного вида, имеющий $m + 1$ конечных производных. Произведя действия, аналогичные изложенным выше, можно получить среднеквадратичную ошибку двойного дифференцирования

$$\sigma_M \approx \sigma_{m+1} T^{m-1} \frac{2}{m+1} \sum_{i=1}^m i^{-1} = \sigma_{m+1} T^{m-1} q_1(m) \quad (4.28)$$

и относительную среднеквадратичную ошибку дифференцирования

$$\Delta_M = \frac{\sigma_M}{\sigma_2} \approx \frac{\sigma_{m+1} T^{m-1}}{\sigma_2} \cdot \frac{2}{m+1} \sum_{i=1}^m i^{-1} = \frac{\sigma_{m+1} T^{m-1} q_1(m)}{\sigma_2}, \quad (4.29)$$

где σ_{m+1} — среднеквадратичное значение $(m + 1)$ -й производной входного сигнала, σ_2 — среднеквадратичное значение второй производной. Из (4.29) можно получить формулу (4.26), если сделать справедливую для гармонического сигнала подстановку $\sigma_{m+1} = \beta^{m-1} \sigma_2$.

Влияние шумов квантования. Квантование по уровню вызывает появление дополнительной ошибки, носящей случайный характер. Статическая характеристика входного преобразователя ЦВМ изображена на рис. 3.2, а. В процессе квантования входная величина округляется до ближайшего целого числа единиц младшего разряда преобразователя. Максимальная ошибка округления при этом не может превосходить половины единицы младшего разряда — $0,5\delta_1$.

Принято исходить из равновероятного распределения ошибки квантования. Дисперсия ошибки квантования при этом составляет $D_1 = \delta_1^2/12$. Ошибка квантования может быть представлена в виде дискретного белого шума с корреляционной функцией вида $K[iT] = D_1 \delta[iT]$, где $\delta[iT]$ — единичная импульсная решетчатая функция. Тогда для дискретных моментов времени $t = iT$, где i — целое число, случайные ошибки квантования можно считать независимыми, что позволяет определить дисперсию результирующей ошибки квантования при вычислении производной суммиро-

Т а б л и ц а 4.4

m	1	2	3	4
$F_0(m)$	2	6,5	14,7	31
$\frac{\delta_{\kappa} T}{\delta_1}$	0,407	0,738	1,11	1,61
m	5	6	7	8
$F_0(m)$	68	160	396	1030
$\frac{\delta_{\kappa} T}{\delta_1}$	2,38	3,66	5,75	9,26

ванием дисперсий ошибок квантования в дискретные моменты времени.

В случае вычисления первой производной в соответствии с формулой (4.12) суммарная дисперсия ошибки округления будет

$$D_{\kappa} = \frac{\delta_1^2}{12} \sum_{i=0}^m a_i^2 = \frac{\delta_1^2}{12} F_0(m). \quad (4.30)$$

В табл. 4.4 приведены значения функции $F_0(m)$ и среднеквадратичной шумовой ошибки σ_{κ} , отнесенной к величине δ_1/T , при различном числе учитываемых обратных разностей или, что все равно, при числе используемых предыдущих тактов.

Так, например, если $T = 0,002$ с, $m = 1$ (см. табл. 4.1) и $\delta_1 = 1$ угл.мин, то среднеквадратичное значение шумовой ошибки от квантования по уровню на основании табл. 4.4 составит

$$\sigma_{\kappa} = \frac{0,407\delta_1}{T} = \frac{0,407 \cdot 1}{0,002} = 203 \text{ угл.мин/с} = 3,38 \text{ град/с.}$$

При $m = 1$ закон распределения соответствует закону Симпсона. При $m > 1$ на основании центральной предельной теоремы закон распределения будет тем точнее приближаться к нормальному, чем больше величина m . При $m = 1$ максимальное значение ошибки $\sigma_{\kappa \max} = \sigma_{\kappa} \sqrt{1,5} = 1,22 \sigma_{\kappa}$. При $m \gg 1$ приближенно можно положить $\sigma_{\kappa \max} \approx 3\sigma_{\kappa}$.

При вычислении производной по формуле (4.12) необходимо осуществить операцию умножения на T^{-1} . Эта операция заключается в масштабировании сигнала выходного преобразователя. Цена его единицы младшего разряда может быть принята равной $\delta_{11} = \delta_1 T^{-1}$. Тогда операции округления в выходном преобразователе не будут происходить и вся ошибка от квантования по уровню

будет определяться формулой (4.30). Общее число отличных от нуля уровней выходного преобразователя должно быть не менее величины $\dot{g}_{\max} \delta_{11}^{-1} = \dot{g}_{\max} \delta_1^{-1} T$. Этому соответствует требуемое число разрядов выходного преобразователя $\alpha_1 > > 3,3 \lg(1 + \dot{g}_{\max} \delta_1^{-1} T)$.

С целью уменьшения числа разрядов выходного преобразователя возможно укрупнение цены его младшего разряда за счет отбрасывания нескольких младших разрядов в вычисленном коде первой производной. Если отбрасывается α младших разрядов, то в выходном преобразователе будет $\delta_{11} = 2^\alpha \delta_1 / T$. Однако это вносит ошибку округления на выходе ЦВМ. Ошибка округления не будет влиять на точность, если выполняется неравенство

$$\frac{\delta_{11}^2}{12} < \frac{\delta_1^2}{12T^2} F_0(m), \quad (4.31)$$

которое сводится к неравенству

$$\alpha < 1,65 \lg F_0(m). \quad (4.32)$$

При невыполнении последнего неравенства вопрос об ошибке квантования на выходе ЦВМ должен быть исследован дополнительно.

При вычислении второй производной аналогичным образом на основании формулы (4.13) может быть получено выражение для дисперсии шумовой ошибки

$$D_{\kappa} = \sigma_{\kappa}^2 = \frac{\delta_1^2}{12 T^4} \sum_{j=0}^m b_j^2 = \frac{\delta_1^2}{12 T^4} q_2(m), \quad (4.33)$$

где δ_1 — цена единицы младшего разряда входного преобразователя. В табл. 4.5 приведены значения функции $q_2(m)$, а также значения среднеквадратичной шумовой ошибки σ_{κ} , отнесенной к ве-

Т а б л и ц а 4.5

m	2	3	4	5	
$q_2(m)$	6,0	46	197	692	
$\sigma_{\kappa} T^2 / \delta_1$	0,71	1,96	4,05	7,58	
m	6	7	8	9	10
$q_2(m)$	$2,88 \cdot 10^3$	$7,35 \cdot 10^3$	$2,47 \cdot 10^4$	$8 \cdot 10^4$	$2,46 \cdot 10^4$
$\sigma_{\kappa} T^2 / \delta_1$	15,5	24,7	45,2	—	—

личине δ_1/T^2 при различных значениях учитываемых обратных разностей m .

Так, если $T = 0,001$ с, $m = 2$ и $\delta_1 = 1$ угл. мин. (см. табл. 4.3), то среднеквадратичное значение шумовой ошибки на основании табл. 4.5 составит

$$\sigma_{\kappa} = \frac{0,71 \delta_1}{T^2} = \frac{0,71 \cdot 1}{0,001^2} = 0,71 \cdot 10^6 \text{ угл. мин./с}^2.$$

Этот пример иллюстрирует сложность проблемы вычисления на ЦВМ второй производной входного сигнала.

На выходе ЦВМ цена единицы младшего разряда составит

$$\delta_{12} = 2^\alpha \delta_1 / T^2, \quad (4.34)$$

где α — число младших разрядов, которые отбрасываются в полученном коде второй производной. Если $\alpha = 0$, то округление на выходе не производится и цена единицы младшего разряда выходного преобразователя $\delta_{12} = \delta_1/T^2$. Если $\alpha > 0$, то происходит округление. Дисперсия дополнительной ошибки, которая при этом вносится, будет

$$\sigma_{\text{д}}^2 = \frac{\sigma_{12}^2}{12} = 2^\alpha \frac{\delta_1^2}{12 T^4}. \quad (4.35)$$

Эта ошибка может не учитываться, если выполняется неравенство $\sigma_{\text{д}} < \sigma_{\kappa}$. Последнее сводится к неравенствам

$$2^\alpha < \sqrt{q_2(m)}, \quad (4.36)$$

$$\alpha < 3,3 \lg \sqrt{q_2(m)} = 1,65 \lg q_2(m).$$

Формулы (4.36) позволяют выбрать допустимое закругление выходного преобразователя, что снижает число его разрядов.

Оптимизация процесса дифференцирования. Оптимизация алгоритмов дифференцирования в аналитической форме достаточно просто может быть сделана для гармонического сигнала. Основанием для оптимизации является то, что изменение периода дискретности и числа учитываемых обратных разностей по-разному отражается на методической и шумовой ошибках. Исходя из формул (4.18) и (4.30), потребуем минимизации результирующей среднеквадратичной ошибки, квадрат которой при нахождении первой производной сигнала $g(t) = A \sin(\beta t + \psi)$ будет

$$\sigma^2 = \mathbf{D} = \sigma_{\text{М}}^2 + \sigma_{\kappa}^2 = \frac{A^2 \beta^{2(m+1)} T^{2m}}{2(m+1)^2} + \frac{\delta_1^2}{12 T^2} F_0(m). \quad (4.37)$$

Т а б л и ц а 4.6

m	1	2	3	4	5	6	7	8
$F_1(m)$	0,64	0,90	1,21	1,87	2,46	3,72	5,59	9,14
$F_2(m)$	0,408	0,836	1,32	1,98	2,96	4,62	7,14	12
$F_3(m)$	1,31	2,02	2,88	4,07	5,78	8,70	13,1	19,4
$F_4(m)$	1,73	2,86	4,10	5,75	8,18	12,5	18,9	30,5

Дифференцирование этого выражения по периоду дискретности и приравнивание производной нулю дает оптимальное значение периода дискретности

$$T'_{\text{опт}} = \frac{1}{\beta} \left[\frac{(m+1)^2 \delta_1^2}{6 m A^2} F_0(m) \right]^{\frac{1}{2(m+1)}}. \quad (4.38)$$

Подстановка найденного значения в (4.37), деление на средний квадрат входной скорости и извлечение квадратного корня дает минимальное значение результирующей относительной среднеквадратичной ошибки, которое может быть получено при оптимальном выборе периода дискретности

$$\Delta_3 = \left(\frac{\delta_1}{A} \right)^{\frac{m}{m+1}} \left[\frac{m+1}{m \sqrt{m+1}} \frac{F_0(m)}{6} \right]^{\frac{m}{2(m+1)}} = \left(\frac{\delta_1}{A} \right)^{\frac{m}{m+1}} F_1(m). \quad (4.39)$$

Задаваясь различными значениями m , можно при заданном значении величины A/δ_1 вычислить Δ_3 по формуле (4.39) для каждого значения m и в результате определить минимальное значение $\Delta_{3 \min}$ при вариации m . Решение этой задачи не представляет особого труда. Для облегчения расчетов в табл. 4.6 приведены значения функции $F_1(m)$.

В практических расчетах больший интерес представляет решение обратной задачи — нахождение требуемого числа разрядов входного преобразователя или, что все равно, отношения A/δ_1 , при которых обеспечивается получение заданного значения относительной среднеквадратичной погрешности Δ . Если считать, что период опроса выбран оптимальным образом, то, положив $\Delta = \Delta_3$, из (4.39) находим

$$\delta_1 \leq A \Delta^{\frac{m+1}{m}} \left[\frac{m+1}{m \sqrt{m+1}} \frac{F_0(m)}{6} \right]^{-1/2} = \frac{A \Delta^{\frac{m+1}{m}}}{F_2(m)}. \quad (4.40)$$

Значения $F_2(m)$ приведены в табл. 4.6.

Возможна другая постановка вопроса оптимизации, если в качестве критерия оптимальности принять минимум максимальной ошибки вычисления производной. Тогда вместо (4.37) следует рассмотреть формулу для максимальной ошибки

$$\sigma_{\max} = \sigma_{M \max} + \sigma_{K \max} = \frac{A \beta^{m+1} T^m}{m+1} + \frac{3 \delta_1}{T \sqrt{12}} \sqrt{F_0(m)}. \quad (4.41)$$

В выражении (4.41) принято, что максимальные ошибки $\sigma_{M \max}$ и $\sigma_{K \max}$ складываются, что не противоречит физике явления, так как частоты изменения методической и шумовой ошибок отличаются обычно на несколько порядков. Дифференцируя (4.41), можно получить оптимальное значение периода дискретности для этого случая

$$T''_{\text{опт}} = \frac{1}{\beta} \left[\frac{(m+1) 3 \delta_1}{m A \sqrt{12}} \sqrt{F_0(m)} \right]^{\frac{1}{m+1}}. \quad (4.42)$$

Далее, подставляя полученное значение $T''_{\text{опт}}$ в (4.41) и деля на максимальное значение первой производной входного сигнала $\dot{g}_{\max} = \beta A$, получим минимальное значение максимальной относительной погрешности

$$\Delta_{\varepsilon \max} = \left(\frac{\delta_1}{A} \right)^{\frac{m}{m+1}} \left[\left(\frac{m}{m+1} \right)^2 0,75 F_0(m) \right]^{\frac{m}{2(m+1)}} = \left(\frac{\delta_1}{A} \right)^{\frac{m}{m+1}} F_3(m). \quad (4.43)$$

Для решения обратной задачи – выбора числа разрядов входного преобразователя при заданном значении Δ_{\max} – формулу (4.43) можно привести к виду

$$\delta_1 \leq A \Delta_{\max}^{\frac{m+1}{m}} \left[\left(\frac{m+1}{m} \right) 0,75 F_0(m) \right]^{-1/2} = \frac{A \Delta_{\max}^{\frac{m+1}{m}}}{F_4(m)}. \quad (4.44)$$

Для удобства решения задачи минимизации $\Delta_{\varepsilon \max}$ или δ_1 при вариациях величины m функции $F_3(m)$ и $F_4(m)$ даны в табл. 4.6.

При дифференцировании сигнала произвольного вида необходимо исходить из общей формулы (4.20). Тогда результирующая дисперсия ошибки будет

$$\sigma^2 = D = \frac{\sigma_{m+1}^2 T^{2m}}{(m+1)^2} + \frac{\delta_1^2}{12 T^2} F_0(m). \quad (4.45)$$

Дифференцирование этого выражения по периоду дискретности дает оптимальное его значение

$$T'_{\text{опт}} = \left[\frac{(m+1)^2 \delta_1^2}{12 m \sigma_{m+1}^2} F_0(m) \right]^{\frac{1}{2(m+1)}} \quad (4.46)$$

и минимальное значение дисперсии ошибки дифференцирования

$$\sigma_{\text{min}}^2 = \left[\frac{\delta_1^2 \sigma_{m+1}^{2/m} F_0(m)}{12(m+1)^{2/m}} \right]^{\frac{m}{m+1}} \left[\left(\frac{1}{m} \right)^{\frac{m}{m+1}} + m^{\frac{1}{m+1}} \right]. \quad (4.47)$$

Оптимальное значение числа учитываемых обратных разностей m может быть найдено последовательным перебором его возможных значений, начиная от единицы.

Определим, например, потенциальную точность вычисления первой производной на ЦВМ, характеризуемую максимальной ошибкой, сигнала вида $g = A \sin(\beta t + \psi)$, требуемый алгоритм и период дискретности, если $A = 15^\circ$, $\beta = 1 \text{ с}^{-1}$, а $\delta_1 = 1 \text{ угл. миң}$.

В соответствии с формулой (4.43) минимальное значение амплитуды ошибки

$$\Delta_{\text{э max}} = \left(\frac{1}{15 \cdot 60} \right)^{\frac{m}{m+1}} F_3(m) = \left(\frac{1}{900} \right)^{\frac{m}{m+1}} F_3(m).$$

Задаваясь значениями $m = 1, 2, \dots$ и используя табл. 4.6, вычисляем $\Delta_{\text{э max}}$. Результаты представлены в табл. 4.7.

Из таблицы следует, что наивысшая точность достигнута при $m = 3$. Тогда $\Delta_{\text{э max}} = 1,74\%$. При этом в соответствии с формулой (4.42) требуемый период дискретности

$$T''_{\text{опт}} = \left[\frac{(3+1) \cdot 3 \cdot 1}{3 \cdot 15 \cdot 60} \sqrt{\frac{14,7}{12}} \right]^{1/4} = 0,15 \text{ с.}$$

Т а б л и ц а 4.7

m	1	2	3	4
$\Delta_{\text{э max}}, \%$	4,35	2,22	1,74	1,77
m	5	6	7	8
$\Delta_{\text{э max}}, \%$	2,01	2,56	3,44	4,55

Таблица 4.8

m	1	2	3	4	5	6
δ_1 , угл. сек	0,03	0,60	1,32	2,98	1,72	0,13

Цена единицы младшего разряда выходного преобразователя составит при этом $\delta_{11} = \delta_1/T = 1 \cdot 6,7 = 6,7$ угл. мин/с = 400 угл. сек/с. Так как $\dot{g}_{\max} = \beta A = 15^\circ/\text{с} = 54\,000$ угл. сек/с, то число разрядов выходного преобразователя должно быть при этом не меньше величины $\alpha_1 \geq 3,31 \lg(1 + 54\,000/400) = 3,31 \lg 136 = 7$.

С целью уменьшения числа разрядов в соответствии с (4.32) можно определить $\alpha < 1,65 \lg 14,7 = 1,93$. Приняв $\alpha = 1$, получим цену младшего разряда выходного преобразователя $\delta_{11} = 2 \cdot 6,7 = 13,4$ угл. мин/с, а общее число разрядов при этом должно быть не менее величины $\alpha_1 = 7 - 1 = 6$.

Определим требования к входному преобразователю для обеспечения максимальной ошибки дифференцирования, не превышающей 0,1%, для условий рассмотренного примера. В соответствии с формулой (4.44) имеем

$$\delta_1 \leq \frac{A \Delta^{\frac{m+1}{m}}}{F_4(m)} = \frac{900 \cdot 0,001^{\frac{m+1}{m}}}{F_4(m)}$$

Задавая различными значениями m и используя табл. 4.6, вычисляем требуемые значения δ_1 для различных m . Результаты расчета сведены в табл. 4.8.

Из таблицы следует, что в оптимальном случае при $m = 4$ цена единицы младшего разряда входного преобразователя должна составлять 2,98 угл. сек. При этом период дискретности в соответствии с (4.42) должен быть равен

$$T''_{\text{опт}} = \frac{1}{1} \left[\frac{(4+1)3 \cdot 1}{4 \cdot 15 \cdot 60 \sqrt{12}} \sqrt{31} \right]^{1/5} = 0,368 \text{ с.}$$

При нахождении второй производной можно поставить задачу минимизации результирующей дисперсии ошибки. В случае гармонического входного сигнала суммарный средний квадрат ошибки на основании (4.26) и (4.33) будет

$$\sigma^2 = \frac{A^2 \beta^{2m+2} T^{2m-2}}{2} q_1^2(m) + \frac{\delta_1^2}{12 T^4} q_2(m). \quad (4.48)$$

При том же сигнале максимальная ошибка будет

$$\sigma_{\max} = A\beta^{m+1}T^{m-1}q_1(m) + \frac{3\delta_1}{T^2\sqrt{12}}\sqrt{q_2(m)}. \quad (4.49)$$

Дифференцирование (4.48) и (4.49) по периоду дискретности дает условие получения минимальной среднеквадратичной ошибки

$$T'_{\text{опт}} = \frac{1}{\beta} \left[\frac{\delta_1^2}{A^2} \frac{1}{3(m-1)} \frac{q_2(m)}{q_1^2(m)} \right]^{\frac{1}{2m+2}} = \frac{1}{\beta} \left(\frac{\delta_1}{A} \right)^{\frac{1}{m+1}} G_1(m) \quad (4.50)$$

и условие получения минимума максимальной ошибки

$$T''_{\text{опт}} = \frac{1}{\beta} \left[\frac{\delta_1}{A} \frac{\sqrt{3}}{m-1} \frac{\sqrt{q_2(m)}}{q_1(m)} \right]^{\frac{1}{m+1}} = \left(\frac{\delta_1}{A} \right)^{\frac{m-1}{m+1}} G_2(m). \quad (4.51)$$

Подстановка $T'_{\text{опт}}$ в формулу (4.48) после деления на средний квадрат второй производной $0,5 A^2 \beta^4$ и извлечения квадратного корня дает минимальное значение относительной среднеквадратичной ошибки

$$\begin{aligned} \Delta_{\min} &= \\ &= \left(\frac{\delta_1}{A} \right)^{\frac{m-1}{m+1}} q_1^{\frac{2}{m+1}} \left[\frac{q_2}{3(m-1)} \right]^{\frac{m-1}{2(m+1)}} = \left(\frac{\delta_1}{A} \right)^{\frac{m-1}{m+1}} G_3(m). \end{aligned} \quad (4.52)$$

Подстановка $T''_{\text{опт}}$ в формулу (4.49) дает после деления на максимальное значение искомой второй производной $A\beta^2$ минимальное значение относительной амплитуды ошибки

$$\begin{aligned} (\Delta_{\max})_{\min} &= \\ &= \left(\frac{\delta_1}{A} \right)^{\frac{m-1}{m+1}} (3q_2)^{\frac{m-1}{2(m+1)}} \frac{m+1}{\frac{m-1}{m+1}} = \left(\frac{\delta_1}{A} \right)^{\frac{m-1}{m+1}} G_5(m). \end{aligned} \quad (4.53)$$

Для решения обратных задач – определения требований к входному преобразователю при заданных значениях Δ или Δ_{\max} – формулы (4.52) и (4.53) могут быть решены относительно δ_1 . В результате имеем условия получения для оптимального случая требуемой точности дифференцирования по относительной средне-

квадратичной ошибке

$$\delta_1 \leq \frac{A \Delta \frac{m+1}{m-1}}{[G_3(m)] \frac{m+1}{m-1}} = \frac{A \Delta \frac{m+1}{m-1}}{G_4(m)} \quad (4.54)$$

и по относительной максимальной ошибке

$$\delta_1 \leq \frac{A \Delta_{\max} \frac{m+1}{m-1}}{[G_5(m)] \frac{m+1}{m-1}} = \frac{A \Delta \frac{m+1}{m-1}}{G_6(m)}. \quad (4.55)$$

Введенные выше функции $G_1 - G_6$ от числа m приведены в табл. 4.9.

Для входного сигнала произвольного вида следует исходить из общих формул (4.28) и (4.33). Тогда результирующая дисперсия ошибки будет

$$\sigma^2 = \sigma_{m+1}^2 T^{2m-2} q_1^2(m) + \frac{\delta_1^2}{12 T^4} q_2(m). \quad (4.56)$$

Дифференцирование (4.56) по периоду дискретности дает условие оптимума

$$T'_{\text{опт}} = \left[\frac{\delta_1^2 q_2(m)}{6(m-1) \sigma_{m+1}^2 q_1^2(m)} \right]^{\frac{1}{2m+2}} \quad (4.57)$$

и минимальное значение дисперсии ошибки двойного дифференцирования

$$\begin{aligned} \sigma_{\text{min}}^2 &= \\ &= [q_2 \delta_1^2]^{\frac{m-1}{m+1}} [q_1 \sigma_{m+1}]^{\frac{4}{m+1}} \left[\left(\frac{1}{6(m-1)} \right)^{\frac{m-1}{m+1}} + \frac{1}{12} (6(m-1))^{\frac{2}{m+1}} \right]. \end{aligned} \quad (4.58)$$

Оптимальное значение числа m должно находиться перебором (4.58) по возможным его значениям $m = 2, 3 \dots$

Определим, например, потенциальную точность, оцениваемую по максимальной ошибке, вычисления второй производной угла качки при $A = 15^\circ$, $\beta = 1$ рад/с и цене младшего разряда входного преобразователя $\delta_1 = 1$ угл. мин.

В соответствии с формулой (4.52) и табл. 4.9 вычисляем минимальную относительную амплитуду ошибки для различных значений m . Результаты вычислений представлены в табл. 4.10.

Т а б л и ц а 4.9

m	$G_1(m)$	$G_2(m)$	$G_3(m)$	$G_4(m)$	$G_5(m)$	$G_6(m)$
2	1,12	1,62	1,32	2,31	1,74	5,3
3	1,32	1,59	2,24	5,02	4,90	24,1
4	1,41	1,58	3,63	8,55	8,04	31,6
5	1,47	1,57	6,05	14,8	13,9	51,3
6	1,51	1,57	10,2	25,0	25,7	91,2
7	1,54	1,58	16,8	41,7	40,1	132
8	1,56	1,59	30,1	77,6	68,1	219
9	1,58	1,60	50,5	133	121	398
10	1,59	1,61	83,2	219	191	607

Т а б л и ц а 4.10

m	2	3	4	5	
$(\Delta_{\max})_{\min}, \%$	18,1	16,6	14,4	15,1	
m	6	7	8	9	10
$(\Delta_{\max})_{\min}, \%$	20,3	24,2	34,1	53,4	72,6

Минимальное значение относительной амплитуды ошибки составляет 14,4% при $m = 4$. Требуемый оптимальный период дискретности ЦВМ, вычисленный по формуле (4.51), составляет $T''_{\text{опт}} = 0,407$ с. Напомним, что при расчете потенциальной точности определения первой производной этого же сигнала было получено значение относительной амплитуды ошибки 1,74% при более простом алгоритме ($m = 3$). Цена единицы младшего разряда выходного преобразователя составляет здесь при $\alpha = 0$, т.е. при отсутствии округления, $\delta_{12} = \delta_1 T^{-2} = 1 \cdot 0,407^{-2} = 6,05$ угл.мин/с², а потребное число разрядов

$$\alpha_2 \geq 3,3 \lg \left(1 + \frac{\beta^2 A}{\delta_{12}} \right) = 3,3 \lg \left(1 + \frac{1 \cdot 900}{6,05} \right) = 7,2.$$

Определим требования к входному преобразователю ЦВМ при необходимости обеспечить вычисление второй производной сигнала с ошибкой $\Delta_{\max} \leq 0,01 = 1\%$.

В соответствии с формулой (4.55) и табл. 4.9 рассчитываем требуемое значение δ_1 при различных числах m . Результаты расчета сведены в табл. 4.11.

Таблица 4.11

m	2	3	4	5
δ_1 , угл. сек	0,01.	0,22	0,79	1,05

m	6	7	8	9	10
δ_1 , угл. сек	0,94	0,88	0,78	0,43	0,32

Из таблицы следует, что оптимальное значение $m = 5$. В этом случае требуемая цена единицы младшего разряда входного преобразователя достигает максимального значения $\delta_1 = 1,05$ угл.сек. В оптимальном случае период дискретности в соответствии с формулой (4.51) составит $T''_{\text{опт}} = 0,24$ с.

Из рассмотренного видно, что получение на ЦВМ второй производной приводит к более сложным алгоритмам и утяжеляет требования к входным преобразователям по сравнению со случаем получения первой производной.

§ 4.2. ЦИФРОВЫЕ ИНТЕГРАТОРЫ

Общие сведения. Рассматривается задача интегрирования непрерывной функции $g(t)$, представленной в виде цифровой последовательности, полученной в результате квантования ее по времени и по уровню. Целью является нахождение алгоритмов цифрового интегрирования, дающих наибольшее или приемлемое приближение выходной величины к цифровой последовательности, образованной дискретизацией интеграла от функции $g(t)$ по времени.

Поскольку операция интегрирования обратна операции дифференцирования, то в качестве оператора интегрирования может рассматриваться величина $1/p$, где $p = d/dt$. Эта величина может быть получена из формул (4.1) — (4.3) их переворачиванием. Однако практически может рассматриваться только первое слагаемое в этих формулах. В результате получим

$$\frac{1}{p} \approx \frac{T}{z-1} = \frac{Tz^{-1}}{1-z^{-1}}, \quad (4.59)$$

$$\frac{1}{p} \approx \frac{T}{2} \frac{z+1}{z-1} = \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}}, \quad (4.60)$$

$$\frac{1}{p} \approx \frac{Tz}{z-1} = \frac{T}{1-z^{-1}}. \quad (4.61)$$

Использование последующих слагаемых в формулах (4.1) и (4.2) приводит к неустойчивым программам. Использование последующих слагаемых в формуле (4.3) не приводит к неустойчивой программе интегрирования, но практически ничего не дает в части увеличения точности интегрирования.

Если интеграл от функции времени в пределах от 0 до t обозначить $f(t)$, то из формул (4.59) – (4.61) могут быть получены разностные уравнения, определяющие алгоритмы интегрирования для машинного значения этого интеграла

$$f_M[n] = Tg[n-1] + f_M[n-1], \quad (4.62)$$

$$f_M[n] = \frac{T}{2}(g[n] + g[n-1]) + f_M[n-1], \quad (4.63)$$

$$f_M[n] = Tg[n] + f_M[n-1], \quad (4.64)$$

где $g[n]$ и $f_M[n]$ – цифровые представления входной и выходной величин интегратора.

При интегрировании постоянного во времени значения все три формулы (4.59) – (4.61) дают нулевую ошибку. Для определения ошибки интегрирования других входных сигналов введем в рассмотрение ошибку интегрирования на шаге

$$\delta_T f[n] = \nabla(f[n] - f_M[n]) = \nabla f[n] - \nabla f_M[n]. \quad (4.65)$$

Изображения рассматриваемых величин $F(z) = Z\{f[n]\}$ и $F_M(z) = Z\{f_M[n]\}$. Тогда изображение ошибки интегрирования на шаге будет

$$Z\{\delta_T f[n]\} = \frac{z-1}{z} [F(z) - F_M(z)] = \frac{z-1}{z} [F(z) - W_n(z)G(z)], \quad (4.66)$$

где $G(z) = Z\{g[n]\}$ – изображение входного сигнала, а $W_n(z)$ – передаточная функция цифрового интегратора.

Применив к (4.66) обратное z -преобразование, имеем

$$\delta_T f[n] = Z^{-1} \left\{ \frac{z-1}{z} [F(z) - W_n(z)G(z)] \right\}. \quad (4.67)$$

Общая ошибка интегрирования может быть найдена суммированием ошибки на шаге по всем шагам

$$\delta f[n] = \sum_{m=1}^n \delta_T[m]. \quad (4.68)$$

Так, например, если входной сигнал представляет собой линейно возрастающую функцию $g(t) = c_1 t$, а $f(t) = c_1 t^2/2$, то из (4.67) можно получить для формулы (4.59)

$$\begin{aligned} \delta_T f[n] &= Z^{-1} \left\{ \frac{z-1}{z} \left[\frac{c_1 T^2 z(z+1)}{2(z-1)^3} - \frac{T}{z-1} \cdot \frac{c_1 T z}{(z-1)^2} \right] \right\} = \\ &= Z^{-1} \left\{ \frac{c_1 T^2 (z+1)}{2(z-1)^2} - \frac{c_1 T^2}{(z-1)^2} \right\} = Z^{-1} \left\{ \frac{c_1 T^2}{2(z-1)} \right\} = \frac{c_1 T^2}{2} \mathbf{1}[n]. \end{aligned} \quad (4.69)$$

Таким образом, ошибка интегрирования на шаге равна здесь $c_1 T^2/2$, причем она будет наблюдаться, начиная с $t = (n-1)T$. Формула (4.61) дает в этих же условиях ошибку, равную $-c_1 T^2/2$, а формула (4.60) – нулевую ошибку. Формула (4.59) соответствует интегрированию методом прямоугольников с недостатком, формула (4.61) – интегрированию методом прямоугольников с избытком, а формула (4.60) – интегрированию методом трапеций. Результат, даваемый формулой трапеций, можно рассматривать как полусумму результатов двух методов прямоугольников.

При интегрировании линейно возрастающего сигнала суммарная ошибка интегрирования для формулы (4.59) может быть найдена умножением постоянной ошибки на шаге на число шагов

$$\delta f[n] = N \delta_T f[n] = N \frac{c_1 T^2}{2} = t \frac{c_1 T^2}{2}. \quad (4.70)$$

Здесь $N = t/T$ – число шагов, а t – интервал интегрирования. Для формулы (4.61) будет иметь место тот же результат, но с обратным знаком.

Если интегрируется входной сигнал с квадратичным возрастанием во времени $g(t) = c_2 t^2/2$, то формула (4.59) дает ошибку на шаге

$$\begin{aligned} \delta_T f[n] &= Z^{-1} \left\{ \frac{z-1}{z} \left[\frac{c_2 T^3 z(z^2+4z+1)}{6(z-1)^4} - \frac{T}{z-1} \frac{c_2 T^2 z(z+1)}{2(z-1)^3} \right] \right\} = \\ &= \frac{c_2 T^2}{6} t \cdot \mathbf{1}[n] + \frac{c_2 T^2}{3} t \cdot \mathbf{1}[n-1]. \end{aligned} \quad (4.71)$$

Начиная со второго шага, ошибка на шаге оказывается равной $c_2 T^2 t/2$, т.е. возрастает с течением времени, где $t = nT$.

При использовании метода трапеций (4.60)

$$\begin{aligned} \delta_T f[n] &= Z^{-1} \left\{ \frac{z-1}{z} \left[\frac{c_2 T^3 z(z^2+4z+1)}{6(z-1)^4} - \frac{T}{2} \frac{z+1}{z-1} \frac{c_2 T^3 z(z+1)}{2(z-1)^3} \right] \right\} = \\ &= Z^{-1} \left\{ -\frac{c_2 T^3}{12(z-1)} \right\} = -\frac{c_2 T^3}{12} \mathbf{1}[n-1]. \end{aligned} \quad (4.72)$$

Здесь ошибка на шаге оказывается постоянной и равной $-c_2 T^3/12$. Общая ошибка интегрирования составит при этом $\delta f [n] = -tc_2 T^2/12$, где t — интервал интегрирования.

Эти расчеты показывают преимущество интегрирования по методу трапеций и вместе с тем иллюстрируют ограниченные возможности по точности интегрирования при использовании простейших формул (4.59) — (4.61).

Алгоритмы точного интегрирования. Из формулы (4.66) следует, что нулевая ошибка интегрирования будет при равенстве нулю выражения в квадратной скобке. Этому соответствует передаточная функция интегратора

$$W_{\text{и}}(z) = F(z)/G(z), \quad (4.73)$$

где $G(z)$ — изображение входной (интегрируемой) функции времени, а $F(z)$ — изображение ее интеграла по времени. Если входной сигнал представляет собой степенную функцию вида

$$g(t) = c_k t^k / k!, \quad (4.74)$$

то ее изображение будет

$$G(z) = \frac{c_k T^k R_k(z)}{k!(z-1)^{k+1}}. \quad (4.75)$$

Здесь $R_k(z)$ — полином, который может быть представлен в виде определителя [216]

$$R_k(z) = k! \begin{vmatrix} 1 & 1-z & 0 & \dots & 0 \\ \frac{1}{2!} & 1 & 1-z & \dots & 0 \\ \frac{1}{3!} & \frac{1}{2!} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{k!} & \frac{1}{(k-1)!} & \frac{1}{(k-2)!} & \dots & 0 \end{vmatrix}.$$

Отсюда $R_0 = R_1 = 1$. Кроме того,

$$\begin{aligned} R_2(z) &= z + 1, \\ R_3(z) &= z^2 + 4z + 1, \\ R_4(z) &= z^3 + 11z^2 + 11z + 1. \end{aligned} \quad (4.76)$$

Если $k = 0$ и $g(t) = c_0$, то передаточная функция интегратора $W_{\text{и}}(z) = Tz^{-1}/(1 - z^{-1})$ совпадает со случаем интегрирования по методу прямоугольников (4.59).

Если $k = 1$ и $g(t) = c_1 t$, то передаточная функция $W_{\text{и}}(z) = T(1 + z^{-1})/2(z - 1)$ совпадает со случаем интегрирования по методу трапеций (4.60).

Если $k = 2$ и $g(t) = c_2 t^2/2$, то передаточная функция интегратора

$$W_{\text{и}}(z) = \frac{T(z^2 + 4z + 1)}{3(z+1)(z-1)} = \frac{T(1 + 4z^{-1} + z^{-2})}{3(1 - z^{-2})} \quad (4.77)$$

соответствует интегрированию по методу Симпсона и дает нулевую ошибку при интегрировании входного сигнала, представляющего собой временной полином степени не выше двух.

Этой передаточной функции соответствует разностное уравнение (алгоритм работы ЦВМ)

$$f_{\text{м}}[n] = T(g[n] + 4g[n-1] + g[n-2])/3 + f_{\text{м}}[n-2]. \quad (4.78)$$

Заметим, что, как следует из последнего выражения, интегрирование ведется на двойном шаге $T_{\text{и}} = 2T$ и результат интегрирования может выдаваться на выходе интегратора с периодом $2T$.

Если на вход интегратора поступает сигнал вида $g(t) = c_3 t^3/3!$, то изображение ошибки интегрирования в соответствии с (4.66) будет

$$F(z) - W_{\text{и}}(z)G(z) = \frac{c_3 T^4 z(z^3 + 11z^2 + 11z + 1)}{4!(z-1)^5} - \frac{T(z^2 + 4z + 1)}{3(z^2 - 1)} \frac{c_3 T^3 z(z^2 + 4z + 1)}{3!(z-1)^4} = -\frac{c_3 T^4 z}{72}. \quad (4.79)$$

Из последнего выражения может быть получено установившееся значение ошибки интегрирования

$$\delta_{\text{уст}} = \lim_{z \rightarrow 1} \frac{z-1}{z} \left(-\frac{c_3 T^4 z}{72(z+1)(z-1)} \right) = -\frac{c_3 T^4}{144}. \quad (4.80)$$

Нахождение установившегося значения ошибки интегрирования на шаге дает

$$\delta_{T_{\text{уст}}} = \lim_{z \rightarrow 1} \frac{z-1}{z} \frac{z-1}{z} \left(-\frac{c_3 T^4 z}{72(z+1)(z-1)} \right) = 0. \quad (4.81)$$

Если на вход этого интегратора поступает сигнал вида $g(t) = c_4 t^4/4!$, то аналогичные расчеты дают $\delta_{T_{\text{уст}}} = -c_4 T^5/90$.

Последнее выражение может быть обобщено на случай произвольного порядка временного полинома

$$\delta_{T_{уст}} = - \frac{T^5 g^{(4)}(\xi)}{90}. \quad (4.82)$$

Здесь $(n - 2)T < \xi < nT$. В точках съема выходных данных $\xi = nT$.

При $k > 2$ попытка построить интегратор с нулевой ошибкой для полиномов степени выше второй наталкивается на трудности. Получающиеся передаточные функции соответствуют неустойчивым программам работы ЦВМ. Однако для $k = 6$ программа оказывается устойчивой. В этом случае передаточная функция интегратора имеет вид

$$W_{ин}(z) = \frac{3}{10} \frac{T(1 + 5z^{-1} + z^{-2} + 6z^{-3} + z^{-4} + 5z^{-5} + z^{-6})}{1 - z^{-6}}. \quad (4.83)$$

Здесь интегрирование осуществляется блочной обработкой поступающей информации за шесть периодов. Выдача данных на выходе интегратора осуществляется с периодом $T_{ин} = 6T$.

Ошибка интегрирования на шаге $6T$ оказывается равной

$$\delta_T = - \frac{T^7 g^{(6)}(\xi)}{140} - \frac{9T^9 g^{(8)}(\xi)}{1400}. \quad (4.84)$$

Первая составляющая ошибки определяется тем, что при выводе формулы (4.83) величина $41/140$ была приближенно заменена на $3/10$. Вторая составляющая является методической ошибкой.

Полученные выше выражения для ошибок интегрирования входного сигнала позволяют выбрать алгоритм и период дискретности из условия допустимого значения этих ошибок.

Так как в реальных интеграторах входная и выходная величины представлены в виде кодов, то множитель T , входящий во все алгоритмы интегрирования, следует рассматривать как масштабный. Это означает, что цена единицы младшего разряда выходного преобразователя $\delta_2 = T\delta_1$, где δ_1 — цена единицы младшего разряда входного преобразователя.

Для того чтобы избежать ошибок округления коэффициентов, желательно все коэффициенты сделать целыми числами и дробную часть присоединить к масштабу на выходе. Так, для интегрирования по методу трапеций целесообразно принять $\delta_2 = T\delta_1/2$.

Двойное интегрирование. Аналогично формуле (4.73) можно записать выражение для передаточной функции интегратора, обеспечивающего нулевую ошибку интегрирования,

$$W_{ини}(z) = F_{ин}(z)/G(z). \quad (4.85)$$

Здесь $G(z)$ – изображение входной величины $g(t)$, $F_{\text{и}}(z)$ – изображение двойного интеграла от входной величины. Если использовать степенные функции вида $g(t) = c_k t^k / k!$, то при $k = 0$ имеем

$$W_{\text{и}}(z) = \frac{T^2(z+1)}{2(z-1)^2} = \frac{Tz^{-1}}{1-z^{-1}} \frac{T(1+z^{-1})}{2(1-z^{-1})}. \quad (4.86)$$

Эта передаточная функция соответствует последовательному включению двух интеграторов, один из которых работает по методу прямоугольников, а другой – по методу трапеций.

Аналогичным образом могут быть построены двойные интеграторы и для более сложных случаев при $k > 0$.

Ошибка на шаге интегрирования может быть определена аналогично формуле (4.66)

$$Z\{\delta_T[n]\} = \frac{z-1}{z} [F_{\text{ии}}(z) - W_{\text{ии}}(z)G(z)]. \quad (4.87)$$

Влияние шумов квантования. Как и в случае цифрового дифференцирования, на вход цифрового интегратора поступает сигнал в "зашумленном" виде. Положим аналогично изложенному выше, что дисперсия шума квантования равна $\delta_1^2/12$, где δ_1 – цена единицы младшего разряда входного преобразователя. Эффект влияния этого шума при интегрировании можно рассматривать двояко. Во-первых, можно рассматривать дополнительную ошибку интегрирования на каждом шаге. Во-вторых, можно рассматривать накапливающуюся ошибку, вызванную эффектом интегрирования.

Дисперсия ошибки интегрирования на шаге интегрирования в предположении независимости ошибки округления на шагах может быть определена выражением

$$D_T = \frac{T^2 \delta_1^2}{12} \sum_{i=0}^m a_i^2. \quad (4.88)$$

Здесь a_i – коэффициенты при дискретах входного воздействия в моменты времени $nT, (n-1)T, \dots, (n-m)T$.

Так, например, для алгоритма (4.78) имеем

$$D_T = \frac{T^2 \delta_1^2}{12} \left(\frac{1}{9} + \frac{16}{9} + \frac{1}{9} \right) = \frac{T^2 \delta_1^2}{6}.$$

Накапливающаяся шумовая ошибка интегратора может быть найдена суммированием дисперсий шумовой ошибки на каждом шаге интегрирования

$$D_{\text{ш}} = n D_T = t D_T / T_{\text{и}}, \quad (4.89)$$

где n — число сделанных шагов интегрирования, t — время работы интегратора, $T_n = \nu T$ — период интегрирования. Коэффициент ν определяется соотношением между шагом интегрирования и периодом дискретности ЦВМ. В алгоритме (4.78) этот коэффициент $\nu = 2$.

Выражение (4.89) определяет эквивалентный уход интегратора. В отличие от непрерывных интеграторов, у которых выходная величина увеличивается при уходе пропорционально времени, здесь пропорционально времени нарастает дисперсия шумовой ошибки. Этот вид ухода присущ всем цифровым интеграторам и является их отличительной особенностью.

Использование интеграторов в законах управления. В непрерывных системах управления находят применение интеграторы с передаточной функцией k_n/p или $k_n/j\omega$, где k_n — коэффициент передачи интегратора. Так как входная и выходная величины интегратора обычно имеют одинаковую физическую природу, то его передаточная функция должна быть безразмерной. Отсюда следует, что коэффициент передачи интегратора должен измеряться в секундах в минус первой степени.

Так как требования к точности подобных интеграторов невелики, то при использовании цифровых методов приемлемыми оказываются все три передаточные функции (4.59) — (4.61). Однако из них должен быть исключен масштабный множитель T . Тогда дискретные передаточные функции, а также частотные передаточные функции всех трех разновидностей цифровых интеграторов могут быть представлены следующим образом:

$$\begin{aligned} W_n(z) &= z^{-1}/(1-z^{-1}), & W_n^*(j\lambda) &= (1-j\lambda T/2)/j\lambda T, \\ W_n(z) &= (1+z^{-1})/2(1-z^{-1}), & W_n^*(j\lambda) &= 1/j\lambda T, \\ W_n(z) &= 1/(1-z^{-1}), & W_n^*(j\lambda) &= (1+j\lambda T/2)/j\lambda T. \end{aligned} \quad (4.90)$$

Наибольшее приближение к идеальному интегратору имеет второй вариант. Наибольшей простотой реализации обладает третий вариант.

Из формул (4.90) вытекает, что коэффициент передачи интегратора равен $1/T$. Воздействие на этот коэффициент различными масштабирующими устройствами обычно не может применяться, так как это приводит к загромождению (исключению нескольких младших разрядов) входного или выходного преобразователей и вызывает появление зоны нечувствительности. Подобного рода нелинейность может значительно ухудшить динамические качества системы.

Правильным является здесь воздействие на период дискретности. Обычно требуется его увеличение. Тогда частотная передаточ-

ная функция, например, второго варианта формул (4.90) будет иметь вид

$$W_{и}^*(j\lambda) = \frac{1}{j\lambda T_{и}} = \frac{1}{j\lambda \nu T}, \quad (4.91)$$

где $\nu = T_{и}/T$ — коэффициент увеличения периода дискретности. Наиболее удобно делать этот коэффициент равным некоторой степени α числа 2, но, в принципе, он может быть равным любому целому числу.

При введении коэффициента $\nu > 1$ дискретная передаточная функция рассматриваемого интегратора принимает вид

$$W_{и}(z) = \frac{1 + z^{-\nu}}{2(1 - z^{-\nu})}. \quad (4.92)$$

Выбор коэффициента ν делается в соответствии с расчетом динамики проектируемой системы управления.

Появление второго периода дискретности в системе управления обычно не утяжеляет расчетов динамики, особенно если используются частотные методы. При использовании последних учет разных периодов дискретности осуществляется автоматически.

§ 4.3. ЦИФРОВЫЕ ФИЛЬТРЫ

Два класса цифровых фильтров. В общем случае алгоритм работы цифрового фильтра описывается разностным уравнением

$$\sum_{j=0}^s b_j x[n-j] = \sum_{i=0}^k a_i y[n-i], \quad (4.93)$$

где $x[n-j]$ — выборки входной последовательности; $y[n-i]$ — выборки выходной последовательности; a_i, b_j — коэффициенты, задающие свойства фильтра.

Для однозначного определения цифрового фильтра должны быть заданы начальные условия. Эти начальные условия должны быть такими, чтобы фильтр был линейным и рекурсивным. Рекурсивность требует, чтобы любая выборка на выходе могла быть вычислена через ранее вычисленные выборки или начальные условия.

Для рекурсивного цифрового фильтра можно записать соотношение между входной и выходной величинами в момент времени $t = nT$ в виде

$$y[n] = \sum_{j=0}^s b_j x[n-j]/a_0 - \sum_{i=1}^k a_i y[n-i]/a_0 \quad (4.94)$$

с начальными условиями $y[n-1], y[n-2], \dots, y[n-k] = 0$.

В z -области фильтр, описываемый уравнением (4.94), может быть представлен своей передаточной функцией $W(z)$. Для этого выполним z -преобразование обеих частей уравнения (4.93):

$$\sum_{n=-\infty}^{\infty} \left(\sum_{j=0}^s b_j x[n-j] \right) z^{-n} = \sum_{n=-\infty}^{\infty} \left(\sum_{i=0}^k a_i y[n-i] \right) z^{-n};$$

$$\sum_{j=0}^s b_j \sum_{n=-\infty}^{\infty} x[n-j] z^{-n} = \sum_{i=0}^k a_i \sum_{n=-\infty}^{\infty} y[n-i] z^{-n};$$

$$X(z) \sum_{j=0}^s b_j z^{-j} = Y(z) \sum_{i=0}^k a_i z^{-i}.$$

Отсюда передаточная функция

$$W(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{j=0}^s b_j z^{-j}}{\sum_{i=0}^k a_i z^{-i}}. \quad (4.95)$$

Передаточная функция $W(z)$ фильтра — это рациональная функция, коэффициенты которой совпадают с коэффициентами разностного уравнения (4.93).

Допустим, что коэффициенты удовлетворяют следующим условиям: $a_0 = 1$, $a_i = 0$, $i \neq 0$. Тогда разностное уравнение (4.94) примет вид

$$y[n] = \sum_{j=0}^s b_j x[n-j], \quad (4.96)$$

и, следовательно, передаточная функция будет

$$W(z) = \sum_{j=0}^s b_j z^{-j}. \quad (4.97)$$

В этом случае выборки на выходе фильтра зависят только от входных выборок. Такие фильтры имеют конечные реакции длины $s+1$ на единичную импульсную решетчатую функцию $\delta_0[n]$.

По этой причине такие фильтры называют фильтрами с конечной импульсной характеристикой (КИХ). Фильтры с бесконечной импульсной характеристикой (БИХ-фильтры) описываются уравнением (4.94), когда по крайней мере один из коэффициентов a_i (кроме a_0) отличен от нуля.

Проектирование цифровых фильтров связано с расчетом коэффициентов a_i , b_j и реализацией полученных передаточных функ-

ций фильтров аппаратными или программными средствами. В связи с тем, что задача расчета КИХ- и БИХ-фильтров является нетривиальной, ниже дается краткая характеристика методов их расчета.

Элементы теории фильтров с конечной импульсной характеристикой. КИХ-фильтры обладают рядом важных достоинств, основными из которых являются следующие [182].

1. КИХ-фильтры можно эффективно строить как по рекурсивной, так и нерекурсивной схемам. КИХ-фильтры, реализуемые нерекурсивно, всегда устойчивы.

2. При нерекурсивной реализации шумы округления, возникающие за счет выполнения арифметических операций с конечной точностью, легко минимизируются.

3. КИХ-фильтры легко построить так, чтобы они имели линейную фазовую характеристику.

К недостаткам КИХ-фильтров относятся необходимость выполнения большого объема вычислений при аппроксимации фильтров, частотные характеристики которых имеют острые срезы, а также возможная некротность задержки в фильтрах с линейной фазовой характеристикой целому числу интервалов дискретизации.

Синтез цифровых фильтров представляет собой самостоятельную задачу, решение которой может быть получено различными методами. Из основных методов синтеза фильтров обычно используют метод взвешивания с помощью "окна", метод частотной выборки, а также метод расчета оптимальных (по Чебышеву) фильтров.

Метод взвешивания считается весьма удобным для проектирования КИХ-фильтров. Его сущность заключается в следующем. Известно, что частотная характеристика любого цифрового фильтра $W(e^{j\omega T})$ является периодической функцией частоты. Ее можно представить рядом Фурье

$$W(e^{j\omega T}) = \sum_{n=-\infty}^{\infty} w[n] e^{-j\omega n T}, \quad (4.98)$$

где весовая функция

$$w[n] = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} W(e^{j\omega T}) e^{j\omega n T} d\omega. \quad (4.99)$$

Использование соотношения (4.98) связано с двумя трудностями: с необходимостью проведения суммирования в (4.98) в бесконечных пределах и с физической нереализуемостью фильтра, так как импульсная характеристика, т.е. отклик фильтра, начинается в момент времени, равный $-\infty$.

Один из возможных методов построения физически реализуемых КИХ-фильтров заключается в усечении бесконечного ряда Фурье числом $n = \pm M$. Однако простое усечение ряда приводит к хорошо известному явлению Гиббса, которое проявляется в виде выбросов и пульсаций определенного уровня до и после точки разрыва в аппроксимируемой частотной характеристике.

Поэтому на практике применяют метод проектирования КИХ-фильтров, основанный на использовании весовой последовательности $w_k[n]$ конечной длины, которая называется "окном". В результате обеспечивается управление сходимостью ряда Фурье в формуле (4.98).

Предложено много "окон", которые обеспечивают аппроксимацию заданных характеристик. Известны, например, прямоугольное "окно", обобщенное "окно" Хэмминга, "окно" Кайзера, треугольное "окно", "окно" Дольфи — Чебышева, "окно" Ланцоша и др. [112, 182]. Метод взвешивания с помощью "окна" обладает некоторыми особенностями, которые иногда препятствуют применению "окон". Прежде всего, когда характеристика $W(e^{j\omega T})$ имеет сложный вид, формула (4.99) оказывается иногда громоздкой или неудобной для интегрирования. Кроме того, характеристика $W(e^{j\omega T})$ не всегда может быть представлена аналитически. В этом случае вообще трудно говорить об использовании "окон".

Еще одна особенность метода взвешивания заключается в том, что, например, при расчете фильтра нижних частот трудно, как правило, точно определить граничную частоту полосы пропускания, поскольку окно "размывает" разрыв идеальной характеристики. Точное определение граничной полосы пропускания фильтра важно, например, при обеспечении коррекции САУ цифровыми методами [23].

Таким образом, основной недостаток метода взвешивания с помощью "окна" состоит в том, что получаемые КИХ-фильтры не удовлетворяют заданным критериям оптимальности и в большинстве случаев они могут быть существенно улучшены. Более эффективными в указанном смысле являются метод частотной выборки и метод расчета оптимальных (по Чебышеву) фильтров.

Метод частотной выборки основан на прямом получении импульсной характеристики фильтра, передаточная функция которого совпадает с N указанными значениями в N точках на оси частот.

Если выбрать N значений частотной характеристики $W(e^{j\omega T})$, которые обозначены $W_N(k)$, $k = 0, N-1$, то конечная импульсная характеристика цифрового фильтра получается с помощью форму-

лы обратного дискретного преобразования Фурье

$$w[n] = \frac{1}{N} \sum_{k=0}^{N-1} W_N(k) \exp\left(j \frac{2\pi kn}{N}\right). \quad (4.100)$$

Из формулы (4.100) следует, что процедура получения импульсной весовой характеристики фильтра проста и сводится к выбору N равноотстоящих значений (выборки) частотной характеристики и вычислению обратного дискретного преобразования Фурье.

Прямая процедура (4.100) не представляет, к сожалению, практического интереса, поскольку невозможно предсказать поведение частотной характеристики между выборками частотной характеристики. Более того, форма частотной характеристики между выборками может иметь значительные пульсации, вызванные наложением характеристик во временной области.

Для уменьшения пульсаций частотной характеристики часть частотных отсчетов делают независимыми переменными. Значения этих независимых переменных обычно рассчитывают таким образом, чтобы минимизировать некоторую функцию ошибки аппроксимации. Независимыми переменными могут быть, например, частотные отсчеты, расположенные в переходной полосе между двумя полосами, внутри которых частотная характеристика определена.

Наиболее эффективный метод оптимизации основан на решении задачи аппроксимации методами линейного программирования, в частности симплекс-методом.

В качестве иллюстрации рассмотрим следующий пример. Выражение для интерполирующей частотной характеристики получено в общем виде в работах [112, 182]:

$$W(e^{j\omega T}) = \sum_{k=0}^M |W_N(k)| \psi_k(e^{j\omega T}). \quad (4.101)$$

Это выражение можно переписать, выделяя в явном виде переменные частотные выборки в совокупности $A_k = |W_N(k)|$:

$$W(e^{j\omega T}) = B(e^{j\omega T}) + \sum_{m=L_1}^{L_2} A_m \psi_m(e^{j\omega T}), \quad (4.102)$$

где $B(e^{j\omega T})$ представляет собой вклад постоянных частотных выборок в частотную характеристику; $[L_1, L_2]$ — диапазон изменения переменных частотных выборок.

Напишем условия, необходимые для обеспечения минимального значения максимальной ошибки аппроксимации. Общее выражение

для ошибки аппроксимации имеет вид

$$E(e^{j\omega T}) = |W_v(e^{j\omega T})(W_u(e^{j\omega T}) - W(e^{j\omega T}))|, \quad (4.103)$$

где частота ω изменяется в областях постоянных неизменяемых частотных выборок; $W_u(e^{j\omega T})$ — желаемая частотная характеристика; $W_v(e^{j\omega T})$ — произвольная функция, позволяющая устанавливать различные уровни ошибок на различных участках интервала аппроксимации.

Обозначим через δ максимальную ошибку в областях постоянных частотных выборок. Тогда, проводя выборки из частотной характеристики на плотной сетке частот, из (4.103) получаем следующие ограничения:

$$W_v(e^{j\omega_k T}) [W_u(e^{j\omega_k T}) - W(e^{j\omega_k T})] \leq \delta;$$

$$-W_v(e^{j\omega_k T}) [W_u(e^{j\omega_k T}) - W(e^{j\omega_k T})] \leq \delta;$$

$$k = 0, 1, 2, \dots, N.$$

Используя выражение (4.102), имеем

$$W_v(e^{j\omega_k T}) W_u(e^{j\omega_k T}) - W_v(e^{j\omega_k T}) B(e^{j\omega_k T}) -$$

$$-W_v(e^{j\omega_k T}) \sum_{m=L_1}^{L_2} A_m \psi_m(e^{j\omega_k T}) \leq \delta;$$

$$-W_v(e^{j\omega_k T}) W_u(e^{j\omega_k T}) + W_v(e^{j\omega_k T}) B(e^{j\omega_k T}) +$$

$$+ W_v(e^{j\omega_k T}) \sum_{m=L_1}^{L_2} A_m \psi_m(e^{j\omega_k T}) \leq \delta.$$

Отсюда следует, что коэффициенты $\{A_m\}$ должны удовлетворять ограничениям

$$-W_v(e^{j\omega_k T}) \sum_{m=L_1}^{L_2} A_m \psi_m(e^{j\omega_k T}) - \delta \leq -W_v(e^{j\omega_k T}) \times$$

$$\times W_u(e^{j\omega_k T}) + W_v(e^{j\omega_k T}) B(e^{j\omega_k T});$$

$$W_v(e^{j\omega_k T}) \sum_{m=L_1}^{L_2} A_m \psi_m(e^{j\omega_k T}) - \delta \leq W_v(e^{j\omega_k T}) \times$$

$$\times W_u(e^{j\omega_k T}) - W_v(e^{j\omega_k T}) B(e^{j\omega_k T}),$$

обеспечивающим минимум максимальных пульсаций в областях постоянных частотных выборок.

Метод частотных выборок для построения фильтров дает хорошие результаты, когда полученные КИХ-фильтры реализуются рекурсивно. Этот метод хорош также, когда синтезируются узкополосные фильтры, так как в этом случае большинство частотных выборок равно нулю.

Третий подход к синтезу КИХ-фильтров основан на чебышевской аппроксимации.

Взвешенный функционал ошибки аппроксимации $E(e^{j\omega T})$ функции $W_u(e^{j\omega T})$ аппроксимирующей функцией $W(e^{j\omega T})$ имеет вид (4.103)

Чебышевский критерий оптимальности состоит в минимизации максимальной ошибки на множестве A точек ω . При чебышевской аппроксимации аппроксимирующую функцию $W(e^{j\omega T})$ представляют в унифицированном виде

$$W(e^{j\omega T}) = P(e^{j\omega T}) Q(e^{j\omega T}), \quad (4.104)$$

где

$$Q(e^{j\omega T}) = \sum_{n=0}^{N-1} a(n) \cos n \omega T \quad (4.105)$$

– линейная комбинация косинусов, $P(e^{j\omega T})$ – подходящая весовая функция.

Тогда ошибку аппроксимации можно записать в следующем виде:

$$\begin{aligned} E(e^{j\omega T}) &= |W_v(e^{j\omega T}) [W_u(e^{j\omega T}) - P(e^{j\omega T}) Q(e^{j\omega T})]| = \\ &= |W_v(e^{j\omega T}) P(e^{j\omega T}) [W_u(e^{j\omega T}) / P(e^{j\omega T}) - Q(e^{j\omega T})]|, \end{aligned} \quad (4.106)$$

который справедлив везде, за исключением, возможно, точек 0 и π . Обозначим

$$W_P(e^{j\omega T}) = W_v(e^{j\omega T}) P(e^{j\omega T}), \quad (4.107)$$

$$W_Q(e^{j\omega T}) = W_u(e^{j\omega T}) / P(e^{j\omega T}). \quad (4.108)$$

Тогда выражение для ошибки аппроксимации можно записать в виде

$$E(e^{j\omega T}) = |W_P(e^{j\omega T}) [W_Q(e^{j\omega T}) - Q(e^{j\omega T})]|, \quad (4.109)$$

который используется для синтеза фильтров любого типа.

Чебышевские аппроксимации основаны на теореме о так называемом чебышевском альтернансе. Эта теорема может быть сформулирована следующим образом [112]: если $Q(e^{j\omega T})$ – линейная комбинация N косинусов, заданных выражением (4.106), то необходимым и достаточным условием того, чтобы $Q(e^{j\omega T})$ была единственной и наилучшей чебышевской аппроксимацией непрерывной функции $W_Q(e^{j\omega T})$ на множестве A , является наличие

у взвешенной функции ошибки $E(e^{j\omega T})$ по крайней мере $N+1$ экстремальных частот на A .

Для построения оптимальных (по Чебышеву) фильтров предложено несколько методов. Наиболее эффективным методом, для которого созданы машинные программы, является так называемый второй алгоритм Ремеза.

Определенными достоинствами, заключающимися по сравнению с алгоритмом Ремеза, в частности, в большей гибкости, обладает чебышевский метод, сформулированный в виде задачи линейного программирования [112].

Таким образом, фильтры с конечной импульсной характеристикой могут быть синтезированы различными методами. Для многих задач цифровой фильтрации в системах автоматического управления предпочтительными являются метод частотной выборки и метод расчета оптимальных (по Чебышеву) фильтров. Однако для конкретных приложений, связанных, например, с задачей доплеровской фильтрации, возможно применение метода взвешивания с помощью окна [178].

Независимо от выбора метода синтез КИХ-фильтров связан обычно с использованием довольно громоздких вычислительных процедур, которые реализуются в соответствующих программах, ориентированных на использование больших и миниЭВМ. В частности, в работе [112] приведена программа на фортране для построения цифровых фильтров с использованием метода взвешивания с помощью "окна". В программе используются "окно" Ланцоша и "окно" веберовского типа. В этой же работе приведены программа синтеза фильтров методом частотной выборки с использованием алгоритма линейного программирования и программа синтеза оптимальных фильтров на основе алгоритма Ремеза.

Общая характеристика методов расчета БИХ-фильтров. Достижения в построении цифровых фильтров в значительной степени связаны с существованием аналоговых фильтров. Хорошо развитый математический аппарат синтеза фильтров непрерывного времени широко используется для синтеза цифровых фильтров. При этом первоначально осуществляется синтез аналоговых фильтров, а затем применяются простые методы отображения, позволяющие преобразовать фильтры из непрерывной области в дискретную.

Другую группу методов расчета составляют прямые методы расчета в Z -плоскости. Третий подход к расчету БИХ-фильтров заключается в использовании процедур оптимизации для нахождения такого расположения полюсов и нулей в z -плоскости, при котором обеспечивается аппроксимация заданной характеристики фильтра.

Наиболее распространенными методами расчета цифровых фильтров по фильтрам непрерывного времени являются метод

отображения дифференциалов, метод инвариантного преобразования импульсной характеристики, метод билинейного преобразования и метод согласованного z -преобразования [182].

Метод отображения дифференциалов является одним из наиболее простых. Сущность его заключается в замене дифференциалов в дифференциальном уравнении фильтра на конечные разности.

При использовании обратных и прямых разностей производятся соответственно следующие замены:

$$\frac{dy}{dt} \leftrightarrow \frac{y[n] - y[n-1]}{T} = \frac{\nabla y[n]}{T},$$

$$\frac{dy}{dt} \leftrightarrow \frac{y[n+1] - y[n]}{T} = \frac{\Delta y[n]}{T}.$$

Недостатком метода отображения дифференциалов является то, что характеристики аналогового фильтра не сохраняются.

Метод инвариантного преобразования импульсной характеристики позволяет обеспечить соответствие частотных характеристик исходного аналогового фильтра и рассчитываемого цифрового фильтра.

При реализации метода инвариантного преобразования импульсной характеристики переход от передаточной функции $W_n(p)$ аналогового фильтра к передаточной функции цифрового фильтра осуществляется с помощью отображения, при котором используется замена

$$\frac{1}{p+d} \leftrightarrow \frac{T}{1-z^{-1}e^{-d\tau}} \quad (4.110)$$

Применение (4.110) требует разложения исходной передаточной функции аналогового фильтра на простые дроби вида

$$W_n(p) = \sum_{i=1}^N \frac{c_i}{p+d_i},$$

причем каждый коэффициент d_i определяет положение i -го полюса.

Наиболее распространенным методом отображения p -плоскости в z -плоскость, обеспечивающим совпадение частотных характеристик непрерывного и цифрового фильтров, является метод, основанный на билинейном z -преобразовании. Билинейное преобразование использует следующую замену:

$$p \leftrightarrow \frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})}, \quad (4.111)$$

которая совпадает с первым членом разложения (4.2).

Недостаток метода билинейного преобразования заключается в том, что между частотными шкалами аналогового и цифрового фильтров имеется нелинейная зависимость. Ее удается учесть лишь в том случае, когда частотная характеристика аналогового фильтра имеет вид ступенчатой функции. Кроме того, при билинейном преобразовании импульсная и фазовая характеристики аналогового и цифрового фильтров не совпадают.

При согласованном z -преобразовании, называемом также методом прямого z -преобразования, отображающая замена имеет вид

$$p + a \leftrightarrow (1 - z^{-1} e^{-aT})/T. \quad (4.112)$$

Метод согласованного z -преобразования довольно прост в использовании. Однако во многих случаях он неприемлем. Так, например, если частоты, соответствующие нулям аналогового фильтра, превышают половину частоты дискретизации, то положение нулей цифрового фильтра будет существенно искажено эффектом наложения. Кроме того, согласованное z -преобразование неприменимо также в случае, когда передаточная функция аналогового фильтра имеет только полюсы.

Считается, что использование инвариантного преобразования импульсной характеристики или билинейного z -преобразования предпочтительнее применения согласованного z -преобразования.

Среди прямых методов расчета цифровых фильтров отметим расчет по квадрату амплитудной характеристики и расчет БИХ-фильтров во временной области.

Расчет по квадрату амплитудной характеристики основан на использовании выражения

$$|W(e^{j\omega T})|^2 = \frac{1}{1 + A_n^2(\omega)},$$

где $A_n^2(\omega)$ — рациональный полином n -го порядка по тригонометрическим функциям. Соответствующий выбор функции $A_n^2(\omega)$ позволяет получить цифровые фильтры различных типов. Так, фильтру нижних частот Баттерворта соответствует функция

$$A_n^2(\omega) = \frac{\operatorname{tg}^{2n}(T\omega/2)}{\operatorname{tg}^{2n}(T\omega_c/2)},$$

где ω_c — частота среза фильтра.

Применение метода расчета цифровых фильтров по квадрату амплитудной характеристики сопряжено с необходимостью разложения $|W(e^{j\omega T})|^2$ на множители для нахождения нулей и полюсов. Выполнить такое разложение, как правило, непросто. Это ограничивает область применения рассматриваемого метода.

Расчет БИХ-фильтров во временной области основан на подборе такого набора коэффициентов a_i и b_j цифрового фильтра

$$W(z) = \frac{\sum_{j=0}^s b_j z^{-j}}{\sum_{i=0}^k a_i z^{-i}} = \sum_{k=0}^{\infty} w_{\Pi}[k] z^{-k},$$

для которого ошибка

$$\langle \epsilon \rangle = \sum_{k=0}^{P-1} (g[k] - w_{\Pi}[k])^2 h[k] \quad (4.113)$$

минимальна. Здесь $g[k]$ — заданная последовательность в диапазоне $0 \leq k \leq P-1$; $h[k]$ — положительная весовая функция последовательности ошибки.

Характеристика $w_{\Pi}[k]$ зависит от параметров фильтра $\{a_i\}$ и $\{b_j\}$ нелинейно. Поэтому задача минимизации ошибки ϵ решается методами последовательных приближений.

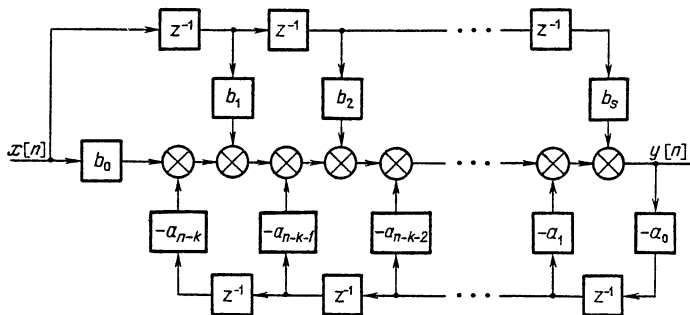
В частном случае, когда $P = n + m + 1$, искомые параметры фильтра, минимизирующие величину ϵ , можно найти, решив систему из $(n + m)$ линейных уравнений.

При аппроксимации заданной импульсной характеристики цифрового фильтра путем воспроизведения ее первых $(m + n + 1)$ отсчетов предполагается, что в целом импульсная и частотная характеристики цифрового фильтра не будут существенно отличаться от заданных характеристик. Однако простого метода для нахождения даже приближенных оценок отклонений любой из этих характеристик в цифровом фильтре от аналогового пока не существует [182].

Последний класс методов расчета БИХ-фильтров основан на методах оптимизации. Отличительная черта этих методов заключается в том, что система уравнений для определения коэффициентов фильтра не может быть решена в явной форме. Поэтому для нахождения коэффициентов используют методы последовательных приближений, с помощью которых в конечном счете сводят некоторую ошибку к минимуму.

Для расчета цифровых фильтров методами оптимизации используют метод минимизации среднеквадратичной ошибки, оптимизацию в w -плоскости с использованием всепропускающих цепей и др. [182].

В заключение отметим, что целесообразность использования того или иного класса методов расчета БИХ-фильтров — методов преобразования аналоговых фильтров в цифровые, прямых методов расчета в z -плоскости и методов оптимизации — определяется конкретными условиями и зависит от многих факторов. На практике широкое распространение получил метод билинейного пре-



Р и с. 4.1. Прямая форма 1 виртуальной структуры

образования стандартных аналоговых фильтров. При проектировании нестандартных БИХ-фильтров рекомендуется использовать также и другие методы. В частности, для расчета фильтров с нестандартными характеристиками, задаваемыми в частотной области, в большинстве случаев наиболее подходящими будут алгоритмы оптимизации.

Виртуальные структуры некоторых цифровых фильтров. При программной реализации цифровых фильтров удобным является представление об их виртуальной структуре. Виртуальная структура цифровых фильтров может быть представлена наглядно в форме схемы. Это позволяет оптимизировать программу работы микроЭВМ по определенным критериям (например, по минимуму объема памяти, минимуму числа операций умножения и т.д.).

В общем случае алгоритм работы цифрового фильтра задается уравнением (4.94). Виртуальная структура фильтра, реализующего это разностное уравнение, показана на рис. 4.1. [182]. Она носит название прямой формы 1. Характерными чертами этой структуры являются ее простота и непосредственная связь с разностным уравнением. Однако, если полюсы $W(z)$ расположены близко друг от друга или от единичной окружности, то, как показано в работе [182], при использовании фильтров данной структуры возникает трудно разрешимая проблема чувствительности характеристик фильтра к погрешностям коэффициентов.

Структура в виде прямой формы 1 требует использования $N = s - 1 + k$ ячеек памяти и выполнения $s + k$ операций умножения.

Получим структуру фильтра в прямой форме 2. Для этого перепишем формулу (4.95) в виде

$$W(z) = \left(\frac{1}{\sum_{i=0}^k a_i z^{-i}} \right) \left(\sum_{j=0}^s b_j z^{-j} \right) = W_1(z) W_2(z).$$

В результате цифровой фильтр, соответствующий формуле (4.95), представлен состоящим из двух последовательно соединенных фильтров с передаточными функциями соответственно $W_1(z)$ и $W_2(z)$. Разностные уравнения этих фильтров

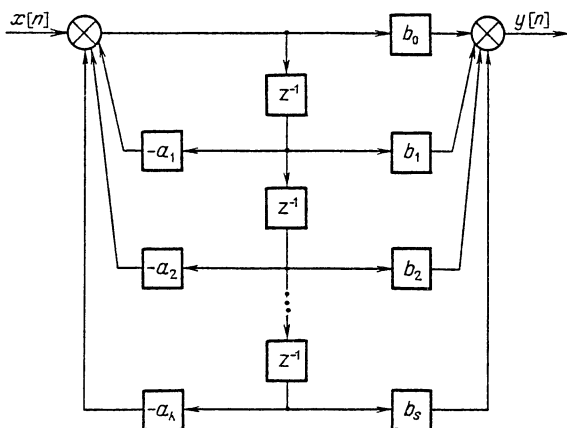
$$\zeta[n] = x[n] - \sum_{i=1}^k a_i \zeta[n-1], \quad (4.114)$$

$$y[n] = \sum_{j=0}^s b_j \zeta[n-j] \quad (4.115)$$

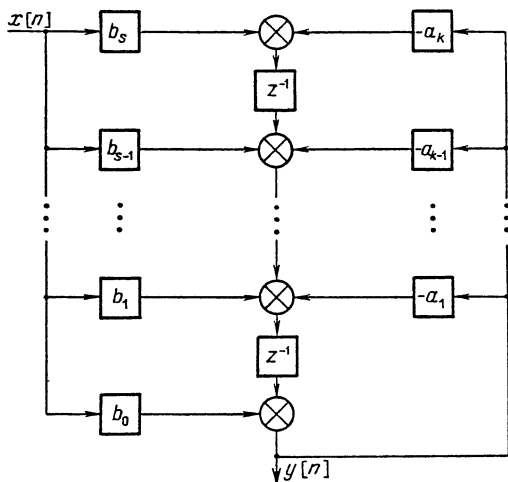
можно реализовать в виде, показанном для $k = s$ на рис. 4.2. [23]. Такую структуру называют также первой канонической формой. В ней используется минимальное количество ячеек памяти, равное $N = s - 1 (s = k)$. Прямая форма 2 имеет такие же недостатки, как и прямая форма 1.

Вторая каноническая форма цифрового фильтра показана на рис. 4.3. [23]. В частном случае знаменатель дроби (4.95) может быть постоянным и равным для простоты единице. При этом фильтр становится нерекурсивным, т.е. текущее значение отклика $y[n]$ зависит только от текущего и конечного числа предшествующих значений входной последовательности. Такой фильтр является КИХ-фильтром.

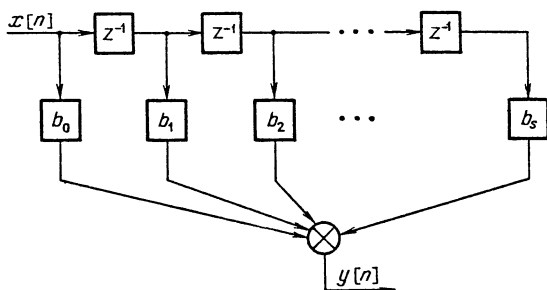
Для построения структурных схем КИХ-фильтров обычно используют прямую форму, показанную на рис. 4.4. Однако для нерекурсивных фильтров применяют также еще несколько виртуальных структур, которые не имеют аналогов с фильтрами об-



Р и с. 4.2. Прямая форма 2 (первая каноническая форма) структуры рекурсивного фильтра



Р и с. 4.3. Вторая каноническая форма представления структуры рекурсивного фильтра



Р и с. 4.4. Виртуальная структура нерекурсивного фильтра

щего вида, содержащими и нули, и полюсы. Наиболее распространенная из них основана на методе быстрой свертки [23].

Приведенные виртуальные структуры не исчерпывают всех возможных вариантов построения цифровых фильтров. В частности, нами не рассмотрены последовательная и параллельная схемы БИХ-фильтров [112]. За пределами рассмотрения оставлены также вопросы повышения скорости обработки информации в КИХ-фильтрах на основе распараллеливания вычислительных операций. Эти вопросы подробнее освещены в § 1.4. Аппаратная реализация быстродействующих КИХ-фильтров детально исследована в работе [182].

Требования к точности задания коэффициентов цифровых фильтров. Рассмотрим простейший случай воспроизведения в цифровой форме аperiodического звена первого порядка с дискрет-

ной частотной передаточной функцией

$$W^*(j\lambda) = \frac{1}{1 + j\lambda T_1} \quad (4.116)$$

При замене $j\lambda = 2(z-1)/[(z+1)T]$ ей соответствует дискретная передаточная функция

$$W(z) = \frac{1-a}{2} \cdot \frac{z+1}{z-a} \quad (4.117)$$

Параметры передаточных функций

$$a = \frac{2T_1 - T}{2T_1 + T} < 1, \quad (4.118)$$

$$T_1 = \frac{1+a}{1-a} \cdot \frac{T}{2}, \quad (4.119)$$

где T – период дискретности.

При реализации дискретного фильтра с передаточной функцией (4.117) неизбежно округление коэффициента (4.118) вследствие ограниченности разрядной сетки цифровой части. Если этот коэффициент может быть реализован с точностью Δa , то постоянная времени (4.119) будет реализована с точностью

$$\Delta T_1 \approx \frac{\partial T_1}{\partial a} \Delta a = \frac{(2T_1 - T)^2}{4T} \Delta a. \quad (4.120)$$

Относительная точность реализации заданного значения постоянной времени

$$\frac{\Delta T_1}{T_1} \approx \frac{(2T_1 - T)^2}{4TT_1} \Delta a \approx \frac{T_1}{T} \Delta a. \quad (4.121)$$

Последнее выражение может служить для формулирования требований к точности реализации требуемого коэффициента (4.118) в цифровой части и, в частности, к допустимому округлению этого коэффициента за счет ограниченности разрядной сетки. Эти требования ужесточаются при снижении периода дискретности.

В более сложном случае реализации аperiodического звена второго порядка с дискретной частотой передаточной функцией

$$W^*(j\lambda) = \frac{1}{(1 + j\lambda T_1)(1 + j\lambda T_2)} \quad (T_1 > T_2) \quad (4.122)$$

дискретная передаточная функция имеет вид:

$$W(z) = \frac{(1-a)(1-b)(z+1)^2}{4(z-a)(z-b)} = \frac{c(z^2 + 2z + 1)}{z^2 + a_1z + a_2} \quad (4.123)$$

Параметры передаточной функции

$$a = \frac{2T_1 - T}{2T_1 + T}, \quad b = \frac{2T_2 - T}{2T_2 + T},$$

$$T_1 = \frac{1+a}{1-a} \cdot \frac{T}{2}, \quad T_2 = \frac{1+b}{1-b} \cdot \frac{T}{2}, \quad (4.124)$$

$$a = \frac{a_1}{2} + \sqrt{\frac{a_1^2}{4} - a_2},$$

$$b = \frac{a_1}{2} - \sqrt{\frac{a_1^2}{4} - a_2}.$$

При реализации на машине передаточной функции (4.123) необходимо установить коэффициенты c , a_1 и a_2 .

Найдем связь между отклонениями Δa_1 и Δa . Из пятого равенства (4.124), выражающего связь между a , a_1 и a_2 , следует

$$\Delta a \approx \frac{\partial a}{\partial a_1} \Delta a_1 \approx \frac{\frac{a_1}{2} + \sqrt{\frac{a_1^2}{4} - a_2}}{2\sqrt{\frac{a_1^2}{4} - a_2}} \cdot \Delta a_1 = \frac{a}{a-b} \Delta a_1. \quad (4.125)$$

При близких значениях коэффициентов a и b числитель в формуле (4.125) может значительно превышать знаменатель. Это приведет к тому, что будет выполняться неравенство $\Delta a \gg \Delta a_1$. В результате требования по реализации коэффициента a_1 оказываются значительно более жесткими, чем требования по реализации коэффициентов a и b . При росте порядка знаменателя передаточной функции $W(z)$ требования могут стать настолько жесткими, что реализация их не может быть осуществлена из-за требований по числу разрядов ЭВМ.

Для устранения этого недостатка переходят от прямого к последовательному или к параллельному программированию. При последовательном программировании реализуемая передаточная функция цифрового фильтра разбивается на элементарные множители первого или второго порядка

$$W(z) = b_0 \prod_{i=1}^N \frac{1 + a_i z^{-1}}{1 + b_i z^{-1}} \prod_{j=1}^L \frac{1 + a_{1j} z^{-1} + a_{2j} z^{-2}}{1 + b_{1j} z^{-1} + b_{2j} z^{-2}}, \quad (4.126)$$

содержащие вещественные коэффициенты. Структурная схема реализации передаточной функции $W(z)$ будет представлять при этом последовательно соединенные цифровые фильтры с элемен-

тарными передаточными функциями. Так, например, передаточная функция (4.123) будет представлена в виде

$$W(z) = W_1(z)W_2(z),$$

$$W_1(z) = \frac{1-a}{2} \frac{z+1}{z-a} = \frac{1-a}{2} \frac{1+z^{-1}}{1-az^{-1}} \quad (4.127)$$

$$W_2(z) = \frac{1-b}{2} \frac{z+1}{z-b} = \frac{1-b}{2} \frac{1+z^{-1}}{1-bz^{-1}}$$

Требования к точности реализации коэффициентов a и b оказываются здесь такими же, как и в простейшем случае (4.121).

При параллельном программировании передаточная функция цифрового фильтра представляется в виде суммы элементарных дробей. При однократных корнях знаменателя эта сумма равна

$$W(z) = \frac{b_k}{a_k} + \sum_{i=1}^N \frac{A_i z^{-1}}{1 + b_i z^{-1}} + \sum_{j=1}^L \frac{A_{1j} z^{-1} + A_{2j} z^{-2}}{1 + b_{1j} z^{-1} + b_{2j} z^{-2}} \quad (4.128)$$

Структурная схема цифрового фильтра в этом случае может быть представлена в виде параллельного соединения элементарных цифровых фильтров, входящих в (4.128). Как и при последовательном программировании, требования к точности воспроизведения отдельных коэффициентов оказываются не столь жесткими по сравнению с прямым программированием.

§ 4.4. ОБНАРУЖЕНИЕ И КОРРЕКЦИЯ НЕИСПРАВНОСТЕЙ

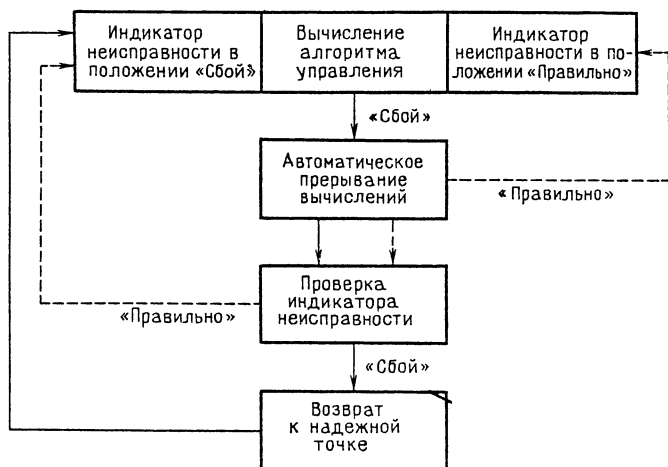
Кратковременные отказы. Наиболее частым видом неисправностей являются кратковременные отказы. Этот тип неисправности определяется тем, что в одном или нескольких разрядах чисел, находящихся в процессоре либо в оперативном запоминающем устройстве, может произойти сбой. Последствиями этого могут быть ошибки в вычислениях, неверная запись в оперативное запоминающее устройство, ошибка в пересылке программы и др.

Одна из возможных логических схем обнаружения ошибок передачи управляющих сигналов одного из каналов управления изображена на рис. 4.5. До окончания каждого цикла вычислений алгоритм управления индикатор неисправностей показывает "сбой". Если все вычисления данного цикла произведены в заданной последовательности, то индикатор будет находиться в положении "правильно". Затем вычисления прерываются для проверки состояния индикатора. Если результат проверки дает ответ "правильно", то подпрограмма прерывания заканчивается, а программа вычисления алгоритма управления данного канала возвращается к соответствующему моменту реального времени и вычисления продолжают.

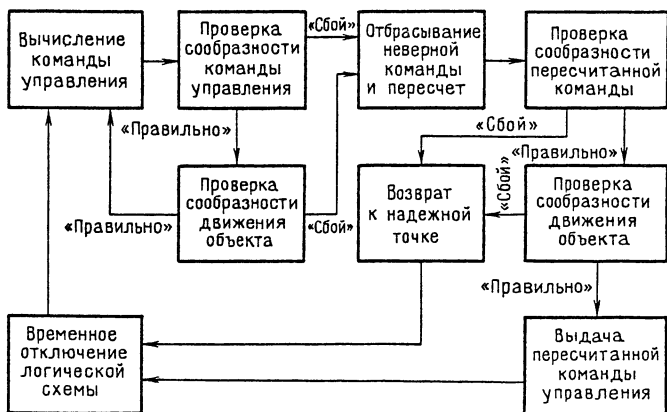
Если же вычисления в алгоритме управления канала не закончены, а программа перешла к вычислениям по другим алгоритмам, например, для управления следующим каналом, то подпрограмма прерывания вернет программу вычислений в необходимое положение. Если при этом индикатор неисправностей будет находиться в положении "сбой", то программа вычислений будет возвращена к некоторой "надежной точке" (последней точке промежуточного значения, начальной точке вычислений, опирающихся на константы, записанные в долговременном запоминающем устройстве, и др.). При введении "надежной точки" предполагается, что сбой может исказить информацию, которая хранится в оперативной памяти. Поэтому при реализации "надежной точки" необходимо повторно вводить максимум требующихся исходных данных из долговременного устройства.

Штриховой линией на рис. 4.5 показано прохождение сигналов при наличии сбоя, сплошной — при правильной работе.

Ошибки вычислений. Ошибки вычислений и записи чисел в оперативное запоминающее устройство могут определяться и корректироваться по схеме, изображенной на рис. 4.6. Сначала проверяется конечная точка вычислений — команда, подаваемая на исполнительное устройство. Сообразность этой команды может определяться сравнением разности двух последовательных команд с порогом, полученным при моделировании и соответствующим максимальным возмущением, действующим на объект управления. Если указанная разность удовлетворяет критерию сообразности, то входные данные канала управления (управляемая величина или ее про-



Р и с. 4.5. Логический контур поиска неисправностей



Р и с. 4.6. Логический контур проверки соответствия вычислений

изводные) сравниваются с некоторыми константами, также полученными при моделировании. Эта проверка предназначена для обнаружения уходов от устойчивого состояния, вызванных, например, неисправностями канала управления.

Если в результате проверки окажется, что команда на управление объектом не соответствует его движению, то текущая команда на управление игнорируется и до вычисления новой команды в следующем цикле сохраняется команда предыдущего цикла. Новые команды проверяются таким же способом. Такая проверка позволяет сохранить работоспособность системы при однократных сбоях. В том случае, когда несколько раз подряд не удовлетворяется критерий соответствия, программа может перейти к процедуре задания "надежной точки" или к выдаче сигнала аварийной ситуации.

§ 4.5. УСТРОЙСТВА КЛАССИФИКАЦИИ СЛУЧАЙНЫХ СИГНАЛОВ

Постановка задачи. Классификация сигналов, т.е. определение принадлежности наблюдаемых непрерывных или дискретных процессов к одному из многих заранее определенных или неизвестных, но выделяемых в процессе обучения классов сигналов, связана с решением многих прикладных задач теории автоматического управления. Необходимость классификации сигналов возникает при технической диагностике объектов управления, таких, как ядерные реакторы, технологические процессы, турбо- и гидрогенераторы, двигатели внутреннего сгорания и многие другие. Классификация случайных сигналов как одна из функций управления необходима при автоматическом распознавании речи, локационных сиг-

налов, временных рядов в экономике, статистике, метеорологии, геофизической разведке и др. [28, 142, 188].

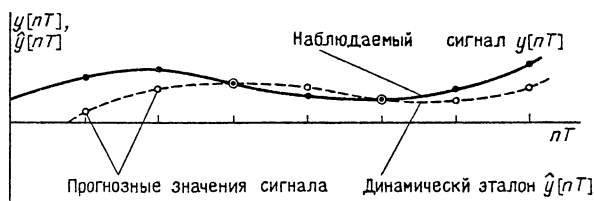
Для классификации используют характерные признаки сигналов: какие-либо их параметры, функции от этих параметров или характеристики этих функций. Часто в качестве признаков классифицируемых сигналов используют некоторые характеристики корреляционных функций или спектральных плотностей сигналов, например, характеристики формы этих функций. Возникающая в этих случаях необходимость в проведении длительного статистического эксперимента, связанного с реализацией алгоритмов быстрого преобразования Фурье, быстрой свертки и других, снижает возможность решения задачи в темпе поступления данных.

В связи с этим особый интерес представляют методы классификации сигналов, алгоритмическая реализация которых приспособлена к функциональным возможностям микроЭВМ, работающих в реальном масштабе времени.

Рассмотрим один из этих методов, основанный на динамической классификации [88, 221]. Идея динамической классификации заключается в том, что для каждого класса сигналов используются, как и обычно, эталоны. Но формируются эти эталоны не заблаговременно, а в процессе классификации. Для этого используется некоторая информация о моделях сигналов.

Методы формирования эталонов в этом случае могут быть разными, но все они основаны на каузальности (причинной обусловленности) наблюдаемых выборочных значений случайных сигналов [112]. Каузальность проявляется в наличии для каждого из классифицируемых сигналов своей строго определенной функциональной связи между предыдущими, текущим и средними последующими их выборочными значениями.

Эталон для каждого класса формируется на основе статистического прогноза на один или несколько шагов вперед, осуществляемого с использованием некоторого количества известных выборочных значений наблюдаемого сигнала [88]. В связи с этим каждый эталон $\hat{y}[nT]$ зависит от текущих значений наблюдаемых данных $y[nT]$ и используемого уравнения прогноза (рис. 4.7).



Р и с. 4.7. Принцип формирования динамического эталона

Такой эталон может быть назван динамическим, а процесс классификации на его основе — динамической классификацией.

В каждый момент времени динамический эталон определяет ожидаемое среднее значение сигнала соответствующего класса. Отклонение фактически наблюдаемого выборочного значения сигнала от ожидаемого среднего подчиняется нормальному закону и характеризуется известным среднеквадратичным отклонением (или дисперсией).

Математические модели классифицируемых сигналов. Широко распространенными представлениями дискретных стационарных случайных сигналов являются математические модели смешанного процесса авторегрессии-скользящего среднего.

Предполагается [28], что смешанный процесс авторегрессии-скользящего среднего можно получить при помощи линейного фильтра с передаточной функцией

$$W(z) = \frac{\sum_{j=0}^S b_j z^{-j}}{1 + \sum_{i=1}^K a_i z^{-i}}, \quad (4.129)$$

на вход которого подан нормально распределенный дискретный белый шум $x[\cdot]$ с нулевым средним и дисперсией σ_x^2 . В выражении (4.129) a_i, b_j — весовые коэффициенты; K — порядок модели авторегрессии; S — порядок модели скользящего среднего.

Если воспользоваться свойством линейности z -преобразования (3.20) и теоремой смещения (3.23), то из выражения (4.129) уравнение математической модели процесса авторегрессии-скользящего среднего для момента времени $t^* = (n+r)T$ представляется в виде

$$y[(n+r)T] = y[n+r] = X - Y, \quad (4.130)$$

где

$$X = \sum_{j=0}^S b_j x[n+r-j] \quad (4.131)$$

— ненаблюдаемая S -мерная компонента;

$$Y = \sum_{i=1}^K a_i y[n+r-i] \quad (4.132)$$

— наблюдаемая K -мерная компонента.

Представление (4.130) называют параметрическим. Соответствующий ему процесс обладает каузальностью или причинной связанностью [75, 112].

Если допустить, что порядок модели скользящего среднего $S = 0$, то в этом случае выражение (4.130) приводится к виду

$$y[n+r] = b_0 x[n+r] + Y. \quad (4.133)$$

Процесс (4.133) является процессом авторегрессии. Он представляет собой конечную взвешенную сумму K предыдущих значений процесса $y[n+r-1]$, $y[n+r-2]$, ..., $y[n+r-K]$ и случайного импульса $x[n+r]$.

Процессу скользящего среднего соответствует математическая модель

$$y[n+r] = X. \quad (4.134)$$

Из выражения (4.134) следует, что процесс скользящего среднего представляет собой взвешенную сумму S значений дискретного белого шума $x[n+r-j]$, $j = \overline{0, S}$.

На практике часто оказывается, что адекватное описание дискретных случайных сигналов достигается при использовании моделей авторегрессии, скользящего среднего или смешанной модели, в которых K и S не больше, а часто и меньше 2. Приведем некоторые примеры моделей сигналов, основанные на результатах работы [23]. Так, математическая модель сигнала с корреляционной функцией экспоненциального вида $K_y[m] = \sigma_y^2 \exp(-|m|T/T_1)$ есть

$$y[n+r] = (1-d)x[n+r] + dy[n+r-1], \quad (4.135)$$

где $d = e^{-T/T_1}$.

Математическая модель сигнала с корреляционной функцией $K_y[m] = \sigma_y^2 \left(\frac{T_1}{T_1 - T_2} e^{-|m|T/T_1} - \frac{T_2}{T_1 - T_2} e^{-|m|T/T_2} \right)$, содержащей две экспоненты, имеет вид

$$y[n+r] = (1-d_1)(1-d_2)x[n+r] + (d_1+d_2)y[n+r-1] - d_1d_2y[n+r-2], \quad (4.136)$$

где $d_1 = \exp\{-T/T_1\}$, $d_2 = \exp\{-T/T_2\}$.

Типовой выходной сигнал следящей системы представляется математической моделью

$$y[n+r] = (1-d)x[n+r-1] + (1+d)y[n+r-1] - dy[n+r-2]. \quad (4.137)$$

Из приведенных примеров видно, что сигналы, характеризуемые математическими моделями (4.135) и (4.136), соответствуют процессам авторегрессии первого и второго порядков, а математической моделью (4.137) описывается смешанный процесс авторегрессии-скользящего среднего. Причем в последнем примере $S = 1$ и $K = 2$.

В заключение отметим, что сигналы разных классов могут отличаться порядками S и K , а также коэффициентами a_i , b_j математических моделей.

Статистические характеристики каузальных сигналов. Для каузальных сигналов характерно наличие функционально-статистической связи наблюдаемой компоненты Y со значением сигнала $y[n+r]$, где r — упреждение [88].

Функциональная связь обусловлена наличием неслучайной зависимости $y[n+r]$ от $-Y$, которая задается вторым слагаемым в уравнении (4.130). Статистический характер в эту зависимость вносит слагаемое X , являющееся случайной величиной, распределенной по нормальному закону, что следует из соотношения (4.131).

Так как величина X распределена по нормальному закону, то на основании уравнения (4.130) значение сигнала $y[n+r]$ также распределено по нормальному закону.

Математическое ожидание сигнала для момента времени $t^* = (n+r)T$ определяется из выражения

$$\begin{aligned} \mathbf{M}\{y[n+r]\} &= \mathbf{M}\left\{\sum_{j=0}^S b_j x[n+r-j] - \sum_{i=1}^K a_i y[n+r-i]\right\} = \\ &= \mathbf{M}\left\{\sum_{j=0}^{r-1} b_j x[n+r-j]\right\} + \mathbf{M}\left\{\sum_{j=r}^S b_j x[n+r-j]\right\} - \\ &- \mathbf{M}\left\{\sum_{i=1}^{r-1} a_i y[n+r-i]\right\} - \mathbf{M}\left\{\sum_{i=r}^K a_i y[n+r-i]\right\}. \quad (4.138) \end{aligned}$$

Здесь первое слагаемое равно нулю. Значения $y[n+r-i]$ для $i=r, K$ в четвертом слагаемом известны, а во втором слагаемом значения $x[n+r-j]$, $j=r, S$, неизвестны, но уже сформированы. Математическое ожидание

$$\mathbf{M}\left\{\sum_{i=1}^{r-1} a_i y[n+r-i]\right\} = \sum_{i=1}^{r-1} a_i \hat{y}[n+r-i], \quad (4.139)$$

т.е. равно взвешенной сумме прогнозов $\hat{y}[n+r-i]$, $i=1, r-1$. Доказательство утверждения (4.139) содержится в работе [28].

Введем обозначения

$$\sum_{j=r}^S b_j x[n+r-j] = X^{(S)}, \quad (4.140)$$

$$\sum_{i=1}^{r-1} a_i \hat{y}[n+r-i] = \hat{Y}^{(r-1)}, \quad (4.141)$$

$$\sum_{i=r}^K a_i y[n+r-i] = Y^{(K)}. \quad (4.142)$$

Тогда в сокращенной форме (4.138) переписывается в виде

$$\mathbf{M}\{y[n+r]\} = m[n+r] = X^{(S)} - \hat{Y}^{(r-1)} - Y^{(K)}. \quad (4.143)$$

Соотношение (4.143) определяет уравнение оптимального прогноза на r шагов. Критерием оптимальности является минимум среднеквадратичной ошибки прогноза [28, 168]. Это же уравнение определяет математическое ожидание каузального сигнала для момента времени $t^* = (n+r)T$, где r — упреждение.

Приведем некоторые примеры, связанные с определением математических ожиданий каузальных сигналов.

Пусть каузальный сигнал описывается моделью (4.135). В этом случае $S = 0$. Допустим, что упреждение $r = 1$. Тогда слагаемые $X^{(S)}$ и $\hat{Y}^{(r-1)}$ в уравнении (4.143) отсутствуют. Следовательно, математическое ожидание

$$\mathbf{M}\{y[n+1]\} = dy[n]. \quad (4.144)$$

Если $r = 2$, то слагаемое $Y^{(K)}$ в уравнении (4.143) отсутствует, и

$$\mathbf{M}\{y[n+2]\} = d\hat{y}[n+1] = d^2y[n]. \quad (4.145)$$

Пусть математическая модель сигнала имеет вид (4.136). Здесь $S = 0$, $K = 2$. Допустим, что упреждение $r = 1$. Тогда

$$\mathbf{M}\{y[n+1]\} = (d_1 + d_2)y[n] - d_1d_2y[n-1]. \quad (4.146)$$

Получим выражение для математического ожидания значения типового входного сигнала (4.137) в момент времени $t^* = (n+1)T$.

Для этого сигнала $S = 1$, $K = 2$. Следовательно,

$$\mathbf{M}\{y[n+1]\} = (1-d)x[n] + (1+d)y[n] - dy[n-1]. \quad (4.147)$$

Значение $x[n]$ в момент времени nT сформировано. Но оно является ненаблюдаемым. Поэтому значение $x[n]$ получают через разность

$$x[n] = y[n] - \hat{y}[n]. \quad (4.148)$$

Здесь значение $y[n]$ известно. Однако прогноз $\hat{y}[n]$ должен быть вновь записан в виде (4.147), т.е.

$$\hat{y}[n] = (1-d)x[n-1] + (1+d)y[n-1] - dy[n-2],$$

где значение $x[n-1]$ вновь неизвестно. Эта процедура имеет теоретически бесконечный характер.

В связи с этим на практике в начальный момент наблюдения полагают некоторое значение $x[n-l] = 0$ (как наиболее вероятное) и записывают уравнение (4.147) относительно $y[n-l]$. Если l выбрано достаточно большим, то переходный процесс, связанный с незнанием $x[n-l]$, можно считать завершенным и использовать полученные оценки для $\mathbf{M}\{y[n+1]\}$.

$$\begin{aligned} \epsilon[n+r-i-\dots-\mu] &= \\ &= X_{(i+u+\dots+\mu)}^{(r-1)} - \sum_{\nu=1}^{r-i-\dots-\mu-1} a_{\nu} \epsilon[n+r-i-\dots-\mu-\nu], \end{aligned}$$

$$\text{где } X_{(i+u+\dots+\mu)}^{(r-1)} = \sum_{j=0}^{r-(i+u+\dots+\mu)-1} b_j x[n+r-(i+u+\dots+\mu-j)].$$

Решением системы уравнений (4.155) является

$$\begin{aligned} \epsilon[n+r] &= X^{(r-1)} - \sum_{i=1}^{r-1} a_i (X_{(i)}^{(r-1)} - \sum_{u=1}^{r-i-1} a_u (X_{(i+u)}^{(r-1)} - \\ &- \sum_{\nu=1}^{r-i-u-1} a_{\nu} (\dots (X_{(i+u+\dots+\mu)}^{(r-1)} - \\ &- \sum_{\nu=1}^{r-i-u-\dots-\mu-1} a_{\nu} \epsilon[n+r-i-\dots-\mu-\nu]) \dots)). \end{aligned} \quad (4.156)$$

После подстановки (4.156) в соотношение (4.150) получаем

$$\begin{aligned} \mathbf{D}\{y[n+r]\} &= \sigma^2[n+r] = \mathbf{M}\left\{\left(X^{(r-1)} - \sum_{i=1}^{r-1} a_i (X_{(i)}^{(r-1)} - \right. \right. \\ &- \sum_{u=1}^{r-i-1} a_u (X_{(i+u)}^{(r-1)} - \sum_{\nu=1}^{r-i-u-1} a_{\nu} (\dots (X_{(i+u+\dots+\mu)}^{(r-1)} - \\ &- \sum_{\nu=1}^{r-i-u-\dots-\mu-1} a_{\nu} \epsilon[n+r-i-\dots-\mu-\nu]) \dots))\left.\right)^2\}. \end{aligned} \quad (4.157)$$

Получим выражение для дисперсии сигнала (4.135) в момент времени $t^* = (n+1)T$.

Решение следует из выражения (4.157) непосредственно, если учесть, что

$$X^{(r-1)} = \sum_{j=0}^{r-1} b_j x[n+r-j] = b_0 x[n+1] = (1-d)x[n+1],$$

и сумма вида $\sum_{i=1}^{r-1} (\dots)$ отсутствует, так как $r=1$. Тогда

$$\mathbf{D}\{y[n+1]\} = (1-d)^2 \sigma_x^2. \quad (4.158)$$

Найдем выражение для дисперсии сигнала (4.135) в момент времени $t^* = (n+2)T$.

В этом случае

$$\begin{aligned} D\{y[n+2]\} &= M\{(X^{(r-1)} - a_1 X_{(1)}^{(r-1)})^2\} = \\ &= M\left\{\left(\sum_{j=0}^1 b_j x[n+2-j] - a_1 b_0 x[n+1]\right)^2\right\}. \end{aligned}$$

Имея в виду, что $b_0 = (1-d)$, $b_1 = 0$, $a_1 = d$, получаем

$$D\{y[n+2]\} = (1-d)^2(1+d^2)\sigma_x^2. \quad (4.159)$$

Легко показать, что дисперсия сигнала (4.136) в момент времени $t^* = (n+1)T$ есть

$$\begin{aligned} D\{y[n+1]\} &= M\{(X^{(r-1)})^2\} = M\{(b_0 x[n+1])^2\} = \\ &= b_0^2 \sigma_x^2 = (1-d_1)^2(1-d_2)^2 \sigma_x^2. \end{aligned} \quad (4.160)$$

Что касается дисперсии типового сигнала автоматической системы для момента времени $t^* = (n+1)T$, то она равна нулю, так как коэффициент b_0 математической модели (4.137) равен нулю.

Алгоритмы классификации. Решающее правило для классификации каузальных сигналов определяется методами теории статистических решений. В двувальтернативной задаче, решение которой легко распространяется на случай m сигналов, выдвигаются и проверяются гипотезы H_1 , H_2 о соответствии математического описания наблюдаемого процесса одной из двух моделей. Для удобства в качестве меры соответствия выберем разности

$$\epsilon[n+r] = y_1[n+r] - M\{y_1[n+r]\} \text{ для гипотезы } H_1,$$

или

$$\epsilon[n+r] = y_2[n+r] - M\{y_2[n+r]\} \text{ для гипотезы } H_2,$$

являющиеся соответственно ошибками прогноза по двум гипотезам.

Если наблюдается процесс $y_1[n+r]$, то на основании формул (4.139), (4.143), (4.151) и (4.157) математическое ожидание и дисперсия ошибки прогноза равны соответственно:

$$\begin{aligned} M_1\{\epsilon[n+r]\} &= m_1[n+r] = M_1\{y_1[n+r] - \hat{y}_1[n+r]\} = \\ &= M_1\{X_1^{(r-1)} - Y_1^{(r-1)} + \hat{Y}^{(r-1)}\} = 0, \end{aligned} \quad (4.161)$$

$$\begin{aligned} D_1\{\epsilon[n+r]\} &= \sigma_1^2[n+r] = M_1\{(y_1[n+r] - \hat{y}_1[n+r])^2\} = \\ &= M_1\left\{\left(X_1^{(r-1)} - \sum_{i=1}^{r-1} a_{1i} X_{1(i)}^{(r-1)} - \sum_{u=1}^{r-i-1} a_{1u} X_{1(i+u)}^{(r-1)} - \right. \right. \\ &\quad \left. \left. - \sum_{v=1}^{r-i-u-1} a_{1v} (\dots (X_{1(i+u+v)}^{(r-1)} + \mu) \dots)\right)^2\right\} \end{aligned}$$

Таким образом, плотности распределения вероятности ошибки прогноза в момент времени $t^* = (n + r)T$ для двух гипотез нормальны и имеют параметры соответственно $N_1 \{0, \sigma_1^2 [n + r]\}$ и $N_2 \{m_2 [n + r], \sigma_2^2 [n + r]\}$.

Дискриминантная функция в этом случае нелинейна и для байесова критерия оптимальности алгоритм классификации при использовании одного отсчета $\epsilon [n + r]$ имеет вид

$$\ln \Lambda(\epsilon [n + r]) = A\epsilon^2 [n + r] + B\epsilon [n + r] + C = 0, \quad (4.173)$$

где

$$A = \frac{1}{2\sigma_2^2 [n + r]} - \frac{1}{2\sigma_1^2 [n + r]}; \quad B = \frac{m_2 [n + r]}{\sigma_2^2 [n + r]};$$

$$C = \frac{m_2^2 [n + r]}{2\sigma_2^2 [n + r]} + \ln \frac{\sigma_2 [n + r]}{\sigma_1 [n + r]}. \quad (4.174)$$

Дискриминантной функции (4.173) соответствует однопороговое решающее правило. Если найти корни уравнения (4.173), то решающее правило становится двупороговым, а алгоритм классификации — линейным, определяемым при $q = \sigma_1 [n + r] / \sigma_2 [n + r] \geq 1$ уравнениями

$$\epsilon_H [n + r] \leq \epsilon [n + r] \leq \epsilon_B [n + r] \quad \text{для } H_1, \quad (4.175)$$

$$\epsilon_B [n + r] < \epsilon [n + r] < \epsilon_H [n + r] \quad \text{для } H_2,$$

где $\epsilon_H [n + r] = \frac{-B - \sqrt{B^2 - 4AC}}{2A}$ — нижний пороговый уровень

принятия решения; $\epsilon_B [n + r] = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$ — верхний

пороговый уровень.

На практике широко используется разновидность байесова критерия — критерий Неймана — Пирсона. В этом случае решающее правило (4.175) сохраняется, а величины пороговых уровней получают из следующих соображений [82].

Условная вероятность ошибки второго рода, называемой ложной тревогой,

$$\alpha [n + r] = 1 - \operatorname{erf}_* \left\{ \frac{\epsilon_B [n + r] - m_2 [n + r]}{\sigma_2 [n + r]} \right\} +$$

$$+ \operatorname{erf}_* \left\{ \frac{\epsilon_H [n + r] - m_2 [n + r]}{\sigma_2 [n + r]} \right\}, \quad (4.176)$$

где

$$\operatorname{erf}_* [\cdot] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{[\cdot]} e^{-t^2/2} dt$$

— интеграл вероятности.

Так как функция $N_2 \{m_2 [n + r], \sigma_2^2 [n + r]\}$ симметрична относительно оси, проходящей через точку $m_2 [n + r]$, то для критерия Неймана—Пирсона $|\epsilon_B [n + r] - m_2 [n + r]| = |\epsilon_H [n + r] - m_2 [n + r]|$. Следовательно, выражение (4.176) можно переписать в виде

$$\alpha [n + r] = 2 \operatorname{erf}_* \left\{ \frac{\epsilon_H [n + r] - m_2 [n + r]}{\sigma_2 [n + r]} \right\} .$$

Отсюда при заданной величине $\alpha [n + r]$ значение

$$\epsilon_H [n + r] = m_2 [n + r] + \sigma_2 [n + r] \operatorname{erf}_*^{-1} \{0,5\alpha [n + r]\} . \quad (4.177)$$

Тогда очевидно, что

$$\epsilon_B [n + r] = m_2 [n + r] - \sigma_2 [n + r] \operatorname{erf}_*^{-1} \{0,5\alpha [n + r]\} . \quad (4.178)$$

Уравнения (4.177) и (4.178) задают пороги принятия решения при динамической классификации сигналов на основе прогноза с упреждением r . В частном случае, когда используется прогноз на один шаг вперед,

$$\epsilon_H [n + 1] = m_2 [n + 1] + \sigma_2 [n + 1] \operatorname{erf}_*^{-1} \{0,5\alpha [n + 1]\} ; \quad (4.179)$$

$$\epsilon_B [n + 1] = m_2 [n + 1] - \sigma_2 [n + 1] \operatorname{erf}_*^{-1} \{0,5\alpha [n + 1]\} . \quad (4.180)$$

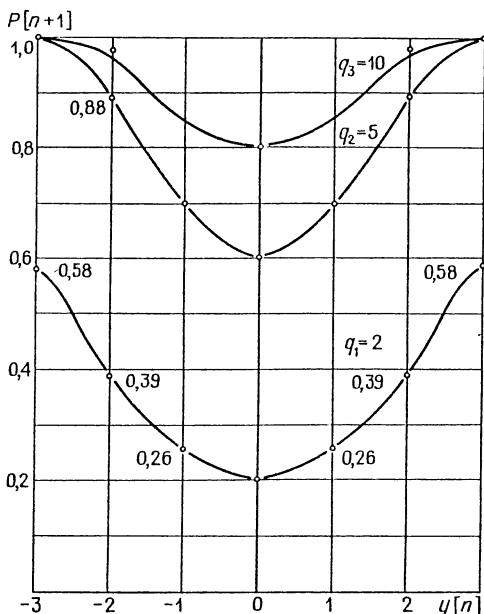
Уравнения (4.177), (4.178) и (4.179), (4.180) необходимы для определения порогов принятия решения. Так как математическое ожидание $m_2 [n + r]$ на основании уравнений (4.165), (4.166), (4.167) является функцией ненаблюдаемых значений дискретного шума $X_h^{(S)}$, прогнозов $\hat{Y}_h^{(r-1)}$ и наблюдаемых значений сигналов $Y_h^{(K)}$, то величины $\epsilon_H [n + 1]$ и $\epsilon_B [n + 1]$ не являются постоянными и должны вычисляться для каждого очередного шага классификации. В изложенном смысле алгоритму классификации, основанному на статистике $\epsilon [n + r]$, соответствует параметрически адаптивное устройство классификации [138].

Самостоятельный интерес представляет нахождение выражения для условной вероятности правильного решения, используемое обычно для построения рабочих характеристик.

В рассматриваемом случае условная вероятность правильного решения

$$\begin{aligned} P [n + r] &= 1 - \operatorname{erf}_* \left\{ \frac{m_2 [n + r] - \sigma_2 [n + r] \operatorname{erf}_*^{-1} \{0,5\alpha [n + r]\}}{\sigma_1 [n + r]} \right\} + \\ &+ \operatorname{erf}_* \left\{ \frac{m_2 [n + r] + \sigma_2 [n + r] \operatorname{erf}_*^{-1} \{0,5\alpha [n + r]\}}{\sigma_1 [n + r]} \right\} = \\ &= 1 - \operatorname{erf}_* \{m_{2H} [n + r] - q^{-1} [n + r] \operatorname{erf}_*^{-1} \{0,5\alpha [n + r]\}\} + \\ &+ \operatorname{erf}_* \{m_{2H} [n + r] + q^{-1} [n + r] \operatorname{erf}_*^{-1} \{0,5\alpha [n + r]\}\} , \quad (4.181) \end{aligned}$$

где $m_{2H} [n + r] = m_2 [n + r] / \sigma_1 [n + r]$ — нормированное математи-



Р и с. 4.8. Зависимость вероятности правильного решения от уровня прогнозируемого сигнала

ческое ожидание ошибки прогноза при наблюдении сигнала, соответствующего альтернативе; $q[n+r] = \frac{\sigma_1[n+r]}{\sigma_2[n+r]}$ – параметр классификации.

Получим для примера уравнение рабочих характеристик устройства классификации сигналов с экспоненциальными корреляционными функциями на основе динамической классификации с прогнозом на шаг вперед.

В этом случае на основании уравнения (4.135) математические модели сигналов, соответствующих гипотезе H_1 и альтернативе H_2 , соответственно

$$y_1[n+1] = (1 - d_1)x[n+1] + d_1y[n],$$

$$y_2[n+2] = (1 - d_2)x[n+1] + d_2y[n].$$

Будем полагать, что $d_2 > d_1$. Тогда на основании равенств (4.163), (4.172) и (4.168) соответственно $\sigma_1[n+1] = (1 - d_1)\sigma_x$, $\sigma_2[n+1] = (1 - d_2)\sigma_x$, $m_2[n+1] = (d_1 - d_2)y[n]$. Следовательно,

$$q[n+1] = \frac{1 - d_1}{1 - d_2}; \quad (4.182)$$

а нормированное математическое ожидание

$$m_{2н}[n+1] = (d_1 - d_2)y[n]/(1 - d_1)\sigma_x. \quad (4.183)$$

Выражение (4.181) с учетом соотношений (4.182) и (4.183) может быть преобразовано к виду

$$P[n+1, \alpha] = 1 - \operatorname{erf}_* \{ q^{-1}(y[n] - k_\alpha) - y[n] \} + \\ + \operatorname{erf}_* \{ q^{-1}(y[n] + k_\alpha) - y[n] \}, \quad (4.184)$$

где

$$k_\alpha = \operatorname{erf}_*^{-1} \{ 0,5\alpha[n+1] \}.$$

Зависимость $P[n+1]$ от значения $y[n]$ при $\alpha = 10^{-2}$ для трех значений параметра q ($q_1 = 2$, $q_2 = 5$, $q_3 = 10$) показана на рис. 4.8. Из рисунка следует, что вероятность правильного решения находится в прямой зависимости от модуля прогнозируемого сигнала. Высокая вероятность правильного решения может быть получена при $y[n] \geq 2$ и параметре $q \geq 5$. Дальнейшее увеличение вероятности правильного решения может быть получено введением накопления [138].

Реализация алгоритмов классификации на микроЭВМ. Алгоритм динамической классификации каузальных сигналов определяется соотношением (4.175), в котором используются значения нижнего порогового $\epsilon_n[\cdot]$ и верхнего $\epsilon_b[\cdot]$ порогового уровней принятия решений, вычисляемые соответственно по формулам (4.177) и (4.178).

Из анализа соотношений (4.177), (4.178) следует, что для определения значений $\epsilon_n[\cdot]$ и $\epsilon_b[\cdot]$ необходимо для каждого значения наблюдаемого процесса $y[n]$, $y[n-1]$, ... вычислять математическое ожидание $m_2[\cdot]$, используя для этого соотношение (4.168).

В частности, при классификации сигналов с корреляционной функцией экспоненциального типа величина

$$m_2[n+1] = (d_1 - d_2)y[n]. \quad (4.185)$$

Следовательно, так как второе слагаемое в уравнениях (4.177) и (4.178) является постоянной величиной, которая может быть записана в память микроЭВМ как константа, задача вычисления пороговых уровней и реализации решающего правила (4.175) носит тривиальный характер. Она может быть решена с использованием традиционных микропроцессорных вычислительных структур, рассмотренных, например, в § 1.4.

Глава 5

ДИНАМИЧЕСКИЙ СИНТЕЗ ЦИФРОВЫХ СИСТЕМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

§ 5.1. ОБЩИЕ СВЕДЕНИЯ О СИНТЕЗЕ СИСТЕМ УПРАВЛЕНИЯ

Статические и динамические требования. В процессе разработки систем автоматического управления и регулирования приходится учитывать весьма разнообразный комплекс требований, связанных с различными их характеристиками. Эти требования можно объединить в некоторые основные группы.

К первой группе следует отнести требования, связанные со статическими и динамическими свойствами. Среди них важнейшее место занимают точностные характеристики. Они определяют ошибки, которые могут иметь место в системе управления в различных режимах. На ранних стадиях развития теории автоматического управления главенствовал детерминистский подход, когда входные воздействия, полезные и возмущающие, задавались в виде известных функций времени. При этом, конечно, нельзя было гарантировать, что в реальной системе все будет происходить подобным же образом.

Затем стал использоваться статистический подход, когда воздействия считались случайными функциями времени, но с известными их характеристиками. Для линейных систем задание корреляционных функций или спектральных плотностей воздействий позволило не только решить задачу об оценке точности рассматриваемой системы, но и спроектировать ее оптимальным образом в смысле получения экстремального (чаще всего минимального) значения некоторой оценки точности.

Такой подход вызвал появление большого числа работ, посвященных решению проблемы оптимального построения систем управления.

Были разработаны критерии качества (критерии оптимальности) в виде функционалов, которые следовало минимизировать в процессе синтеза системы. Можно, например, сформулировать задачу оптимального управления как такую задачу, когда при работе в течение длительного времени установившееся значение дисперсии ошибки минимально при заданных полезном входном воздействии и возмущениях. Возможны формулировки понятия оп-

тимальности на основе минимизации времени переходного процесса, минимизации потребления энергии и т.п.

Появление оптимальных методов проектирования характерно не только для области автоматического управления и регулирования. Эти методы находят в настоящее время использование практически во всех областях науки, техники, экономики. Однако достигнутые успехи в части построения оптимальных систем автоматического управления в ряде случаев имеют лишь методическое, а не практическое значение. Это обусловлено тем, что решение задачи оптимизации требует знания априорной информации о воздействиях в системе в смысле знания их статистических характеристик (корреляционных функций, спектральных плотностей или иных эквивалентных исходных данных).

Эта информация, как правило, достоверно неизвестна, что приводит во многих случаях к невозможности спроектировать хорошо работающую оптимальную систему. В принципе возможно построение оптимальных систем, в которых происходит уточнение априорной информации о входных воздействиях. Однако это приводит в большинстве случаев к значительному усложнению системы, что сказывается на других ее показателях (стоимости, надежности и др.).

Вероятно, в настоящее время и в ближайшем обозримом будущем одной из основных форм задания точности систем управления и регулирования остается задание допустимых ошибок (максимальных, среднеквадратичных и др.). Методы синтеза, основанные на таких формах задания точности, будут рассмотрены в § 5.4. Это не исключает использования оптимальных методов в отдельных случаях, когда существуют достаточно надежные исходные данные по входным воздействиям и требуемым критериям качества. Вероятно, со временем, по мере накопления статистики, круг задач, имеющих оптимальное решение, будет непрерывно расширяться.

Решение задач, относящихся к первой группе требований, называется динамическим синтезом систем управления и регулирования.

Материал § 5.2 посвящен решению задачи динамического синтеза при априорно известных статистических характеристиках входных воздействий. Естественно, что в таких условиях задача динамического синтеза должна решаться оптимальным образом. Для этой цели используются оптимальные фильтры Винера.

Винеровская фильтрация соответствует получению наилучших точностных показателей в установившемся режиме. Для получения наилучших показателей также и в переходном режиме используются фильтры Калмана. Однако здесь они не рассматриваются. В настоящее время по этим фильтрам имеется обширная литература [80].

Материал § 5.3 посвящен решению задачи динамического синтеза при отсутствии достоверной информации о статистических характеристиках входных воздействий.

Другие требования при синтезе. Ко второй группе относятся требования, связанные с надежностью работы систем управления, ее устойчивостью к влиянию внешних воздействий (климатических, механических, химических) и способностью сохранять свои характеристики в течение заданного промежутка времени. Сюда относятся в первую очередь такие требования, как вероятность безотказной работы, интервал рабочих температур, вибростойкость и вибропрочность, ресурс, требуемое время, условия хранения и т.п.

К третьей группе относятся требования, связанные с характером эксплуатации систем управления. Сюда относятся условия обслуживания системы в процессе ее работы, квалификация обслуживающего персонала, возможность ремонта и восстановления, периодичность производства проверок и т.д.

К четвертой группе относятся требования, связанные с допустимой массой и габаритами системы и допустимым потреблением энергии. Кроме общего уровня мощности потребления, часто важным является вид энергии (постоянный или переменный ток, пневмопитание), а также стабильность источников питания.

К пятой группе относятся требования, связанные с технологичностью изготовления системы управления. Сюда могут относиться такие требования, как необходимость использования уже освоенных или унифицированных элементов и узлов, простота сборочных и регулировочных операций, экономические показатели и т.п.

К шестой группе относятся требования, связанные с общей ситуацией, имеющей место при проектировании. Сюда относятся требования патентной новизны и чистоты, необходимость использования имеющихся научных, конструкторских и производственных заделов, наличие сложившихся исследовательских и конструкторских кадров и др.

Из этого сравнительно беглого рассмотрения видно, что в процессе проектирования системы управления необходимо учитывать весьма большой круг технических требований. Эти требования могут вступать в противоречие друг с другом. В принципе возможно построение оптимальной системы управления, когда из этих частных требований формируется единый критерий качества системы, экстремальное значение которого будет соответствовать оптимальному построению системы. Однако современное состояние теории оптимизации не позволяет надеяться, что эта задача получит строгое и обоснованное решение в ближайшие годы.

Критерии запаса устойчивости. При синтезе систем управления на основе выполнения требований по точности в смысле ограниче-

ния максимальных или среднеквадратичных ошибок часто возникает необходимость такого значения общего коэффициента усиления разомкнутого контура системы, при котором она либо теряет устойчивость, либо имеет сильную склонность к колебанию, определяемую близостью к границе колебательной устойчивости. Это приводит, как правило, к необходимости использовать в системе корректирующие (демпфирующие) средства.

Достаточность удаления системы от колебательной границы устойчивости может определяться различными критериями. Используются (например в [26]) такие оценки, как колебательность (отношение мнимой части корня характеристического уравнения к вещественной), запасы устойчивости по амплитуде и по фазе, перегулирование, показатель колебательности и др. Все эти критерии в общем приводят к цели.

При расчете систем управления частотными методами и, в частности по логарифмическим частотным характеристикам, удобно использование и частотных оценок запаса устойчивости. Для этой цели наиболее удобен показатель колебательности, равный отношению максимального пика амплитудной частотной характеристики замкнутой системы к ее начальной ординате на нулевой частоте.

В работах [23, 25] рассмотрены принципы построения логарифмических амплитудных частотных характеристик (л.а.х.) цифровых систем управления, имеющих в замкнутом состоянии заданный запас устойчивости, определяемый значением показателя колебательности. Там же приведены типовые формы л.а.х. с заданным запасом устойчивости, которые могут быть использованы при синтезе цифровых систем.

Критерии быстродействия. Наиболее универсальным показателем быстродействия системы управления является время, протекающее от момента ее включения до момента, когда она начинает выполнять свои функции. В это время входят различные составляющие. К их числу относятся, например, время разгона двигателей, время достижения требуемого температурного режима, время переходного процесса и т.п.

В некоторых случаях понятие быстродействия имеет более узкое значение и может оцениваться по времени переходного процесса, полюсу пропускания, степени устойчивости и т.п., т.е. здесь принимаются во внимание только динамические характеристики самой системы управления.

В отличие от критериев запаса устойчивости, для характеристики быстродействия не существует универсальных оценок. Они должны устанавливаться в каждом конкретном случае в зависимости от требований, предъявляемых к системе управления в части времени готовности к выполнению возложенных на нее функций.

Это время может быть самым различным: от нескольких часов в крупных энергетических установках до нескольких миллисекунд в электронных системах управления.

Особенности использования микроЭВМ. При использовании в цифровых системах управления микроЭВМ приходится учитывать ряд факторов. Прежде всего следует отметить, что специализированные микропроцессорные вычислительные устройства, создаваемые для использования в цифровых системах, могут иметь в своем составе несколько микропроцессоров, что позволяет разделять задачи, решаемые в микроЭВМ, между отдельными процессорами. Цель — повышение надежности при введении дублирования или мажоритарной обработки, использование иерархического построения системы, разделения каналов с различным темпом обработки информации и др.

В большинстве случаев вследствие ограниченности разрядной сетки микроЭВМ оказывается невозможным использование дополнительных младших разрядов в арифметическом устройстве, что необходимо иметь в виду при составлении алгоритма работы цифровой части системы управления. Следствие этого — отсутствие явления округления (шумов квантования) в выходном преобразователе микроЭВМ.

При использовании микроЭВМ с одним процессором может возникать проблема ограниченности быстродействия цифровой части при достаточно сложной необходимой обработке поступающей информации.

§ 5.2. ИСПОЛЬЗОВАНИЕ ЦИФРОВЫХ ВИНЕРОВСКИХ ФИЛЬТРОВ

Основные соотношения для винеровских фильтров. При синтезе непрерывной системы управления по критерию минимума среднеквадратичной ошибки может быть поставлена задача нахождения оптимальной структуры и параметров системы управления, если известны статистические характеристики полезного сигнала и помехи, действующие на ее входе.

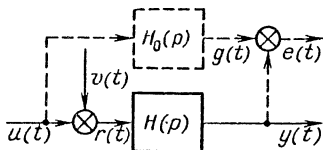
Постановку задачи поясняет рис. 5.1. На этом рисунке обозначено: $u(t)$ — полезный сигнал, $v(t)$ — помеха, $r(t) = u(t) + v(t)$ — результирующий сигнал, $y(t)$ — управляемая величина, $g(t)$ — желаемое значение управляемой величины (задающее воздействие), $e(t) = g(t) - y(t)$ — ошибка системы управления, $H_0(p)$ — оператор обработки полезного сигнала, $H(p)$ — передаточная функция замкнутой системы управления.

В качестве критерия оптимальности рассматривается минимум дисперсии ошибки

$$D_e = \sigma_e^2 = \mathbf{M} \{ e^2(t) \} = \lim_{T \rightarrow 0} \frac{1}{2T} \int_{-T}^T e^2(t) dt. \quad (5.1)$$

Если $H_0(p) = 1$, то это будет задача оптимального сглаживания, т.е. выделение сигнала $g(t) = u(t)$ из аддитивной смеси сигнала и помехи. При равенстве помехи нулю решение задачи сглаживания имеет тривиальный вид: $H(p) = 1$.

В задачах оптимального статистического упреждения $H_0(p) = \exp(p\tau_0)$, где τ_0 – время упреждения. Решение получается нетривиальным даже в случае отсутствия помехи. В задачах дифференцирования сигнала при наличии помех заданный линейный опе-



Р и с. 5.1. Оптимальный фильтр Винера

ратор имеет вид $H_0(p) = p^k$, где k – порядок отыскиваемой производной.

При нахождении оптимальной частотной передаточной функции $H(j\omega)$ конструктор должен попытаться реализовать ее посредством использования тех элементов, которыми он располагает и из которых должна быть построена система управления. Так как в большинстве практических случаев точное воспроизведение оптимальной передаточной функции оказывается невозможным, то приходится использовать квазиоптимальную или субоптимальную систему, более или менее близко совпадающую по своим параметрам с оптимальной.

Задача винеровской фильтрации может быть решена и для многомерного случая, когда рассматриваются матрицы-столбцы величин $u(t)$, $v(t)$ и $g(t)$. Схему, изображенную на рис. 5.1, следует тогда рассматривать как многомерную. В качестве критерия оптимальности здесь принимается минимум математического ожидания $M \{ e' \Gamma e \}$, где Γ – любая положительно определенная матрица, а e – матрица-столбец ошибок. В этом случае минимизируется и каждая составляющая e .

Используются спектральные плотности стационарных случайных процессов: $S_u(\omega)$ – полезного сигнала, $S_v(\omega)$ – помехи, $S_{uv}(\omega)$ и $S_{vu}(\omega)$ – взаимные спектральные плотности полезного сигнала и помехи, $S_{gu}(\omega) = H_0(j\omega)S_u(\omega)$ – взаимная спектральная плотность задающего воздействия и полезного сигнала. При этом для взаимной спектральной плотности задающего воздействия и помехи принято $S_{gv}(\omega) = 0$. В частном случае, когда рассматривается задача оптимальной фильтрации, $H_0(j\omega) = 1$. Тогда $S_{gu}(\omega) = S_g(\omega) = S_u(\omega)$. Кроме того, вводится спектральная плотность $S_{gr}(\omega) = S_{gu}(\omega) + S_{gv}(\omega) = S_{gu}(\omega)$.

Спектральная плотность смеси полезного сигнала и помехи

$$S_r(\omega) = S_u(\omega) + S_v(\omega) + S_{uv}(\omega) + S_{vu}(\omega). \quad (5.2)$$

К ней применяется факторизация (разложение на комплексно-сопряженные множители)

$$\begin{aligned} S_r(\omega) &= \psi(j\omega) \psi(-j\omega), \\ \psi(j\omega) &= [S_r(\omega)]^+, \\ \psi(-j\omega) &= [S_r(\omega)]^-. \end{aligned} \quad (5.3)$$

Полюсы и нули $\psi(j\omega)$ полезно выбирать так, чтобы они лежали в верхней полуплоскости ω (в левой полуплоскости $p = j\omega$).

Решение задачи винеровской фильтрации дает значение оптимальной частотной передаточной функции замкнутой системы

$$H(j\omega) = \frac{1}{\psi(j\omega)} \left[\frac{S_{gr}(\omega)}{\psi(-j\omega)} \right]_+ = \frac{B(j\omega)}{\psi(j\omega)}. \quad (5.4)$$

Функция $B(j\omega)$ определяется следующим образом. Если $S_{gr}(\omega)$ есть отношение двух полиномов по степеням $j\omega$, то можно записать

$$\frac{S_{gr}(\omega)}{\psi(j\omega)} = F(j\omega) + \sum_{i=1}^q \frac{a_i}{j\omega + \alpha_i} + \sum_{i=1}^l \frac{b_i}{-j\omega + \beta_i}, \quad (5.5)$$

где $F(j\omega)$ — некоторый полином. Первая сумма содержит все члены, соответствующие левой половине p -плоскости (включая ось $j\omega$), а вторая сумма — все члены, соответствующие полюсам в правой половине p -плоскости. В этом разложении реализуемая часть определяется суммой

$$B(j\omega) = F(j\omega) + \sum_{i=1}^q \frac{a_i}{j\omega + \alpha_i}. \quad (5.6)$$

Эти формулы позволяют определить оптимальную частотную передаточную функцию замкнутой системы $H(j\omega)$, от которой можно перейти к оптимальной функции $H(p)$ подстановкой $j\omega = p$.

Расчетные формулы для определения дисперсии ошибки в оптимальном случае, т.е. при использовании передаточной функции, в соответствии с (5.4) имеют вид

$$D_{e \min} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_g(\omega) d\omega - \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2 S_r(\omega) d\omega. \quad (5.7)$$

При учете формулы (5.4), из которой следует зависимость

$$|H(j\omega)|^2 = \frac{|B(j\omega)|^2}{|\psi(j\omega)|^2} = \frac{|B(j\omega)|^2}{S_r(\omega)},$$

выражение (5.7) можно привести к виду

$$D_{e \min} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_g(\omega) d\omega - \frac{1}{2\pi} \int_{-\infty}^{\infty} |B(j\omega)|^2 d\omega. \quad (5.8)$$

Однако эти формулы оказываются малоприменимыми в тех случаях, когда задающее воздействие $g(t)$ соответствует нестационарному процессу с неограниченно возрастающей дисперсией. Это может быть, например, при рассмотрении задающего воздействия, представляющего собой типовой входной сигнал следящей системы [23]. При этом наличие в следящей системе астатизма приводит к стационарности и конечности дисперсии ошибок системы. Преобразуем для этого случая формулу (5.7), ограничиваясь условием $H_0(p) = 1$, а также отсутствием корреляционных связей между полезным сигналом $g(t) = u(t)$ и помехой $v(t)$. Тогда

$$D_{e \min} = \frac{1}{2\pi} \int_{-\infty}^{\infty} [1 - |H(j\omega)|^2] S_g(\omega) d\omega - \\ - \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2 S_v(\omega) d\omega. \quad (5.9)$$

Выражение в квадратных скобках в соответствии с изложенным выше должно содержать множитель ω^{2l} , совпадающий с таким же делителем, содержащимся в спектральной плотности $S_g(\omega)$. Поэтому интегрирование выражения, находящегося под знаком первого интеграла (5.9), не дает расходящегося результата, что соответствует конечности дисперсии ошибки системы.

Формулу (5.9) можно записать в другом виде. При отсутствии корреляции между полезным сигналом и помехой дисперсия ошибки определяется выражением

$$D_{e \min} = \frac{1}{2\pi} \int_{-\infty}^{\infty} |1 - H(j\omega)|^2 S_g(\omega) d\omega + \\ + \frac{1}{2\pi} \int_{-\infty}^{\infty} |H(j\omega)|^2 S_v(\omega) d\omega. \quad (5.10)$$

Формулы (5.9) и (5.10) адекватны и могут одинаково использоваться при условии, что передаточная функция замкнутой системы выбрана в соответствии с формулой (5.4).

Цифровые винеровские фильтры. При реализации винеровских фильтров в дискретных (цифровых) системах возникает задача учета явлений квантования по времени и квантования по уровню.

Импульсный характер работы цифровых систем будет оказывать заметное влияние на работу системы и должен учитываться при рас-

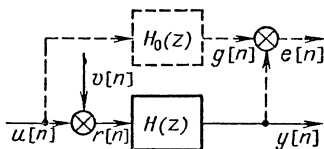
чете в тех случаях, когда период дискретности не может считаться малым. Понятие малости связано с видом полезного сигнала, который поступает на вход системы, а точнее — с его спектральной плотностью.

Частотная передаточная функция линейризованной цифровой системы практически совпадает с частотной передаточной функцией непрерывной системы, если будет выполняться условие

$$|\alpha_{\max}| \ll 2/T, \quad (5.11)$$

где α_{\max} — наибольший по модулю чисто мнимый полюс частотной

Р и с. 5.2. Дискретный вариант фильтра Винера



передаточной функции (он соответствует наибольшему по модулю вещественному полюсу в p -плоскости передаточной функции), T — период дискретности. Для комплексных корней в p -плоскости ($p_{1,2} = -\gamma \pm j\beta$) аналогичное условие имеет вид

$$\beta_{\max} \ll 2/T, \quad (5.12)$$

где β_{\max} — наибольшее значение мнимой части комплексного корня. При выполнении условий (5.11) и (5.12) применительно к полюсам спектральной плотности полезного сигнала учет квантования по времени оказывается ненужным и система может рассматриваться как непрерывная с последующим использованием ЦВМ.

При действии коррелированной помехи $v(t)$ сформулированное условие сохраняет свою силу, если при выбранном периоде дискретности время корреляции помехи меньше периода дискретности и она может рассматриваться как дискретный белый шум.

При невыполнении условий (5.11) и (5.12), точнее, при невозможности выполнить эти условия соответствующим выбором периода дискретности, необходимо перейти к рассмотрению дискретных фильтров Винера. Дискретный вариант фильтра изображен на рис. 5.2. Полезный сигнал и помеха представлены в виде решетчатых функций $u[n]$ и $v[n]$. Передаточной функции оптимального фильтра $H(z)$ соответствует приведенная весовая функция $h[n]$, связанная с передаточной функцией z -преобразованием. В качестве критерия качества рассматривается дисперсия решетчатой функции ошибки $e[n] = g[n] - y[n]$ в виде

$$D_e = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N e^2[n]. \quad (5.13)$$

Как и ранее, изображение задающего воздействия $G(z)$ связано с изображением полезного сигнала $U(z)$ зависимостью

$$G(z) = H_0(z)U(z). \quad (5.14)$$

Вводятся спектральные плотности, аналогичные спектральным плотностям, введенным при рассмотрении непрерывных систем: $S_u^*(\lambda)$, $S_v^*(\lambda)$, $S_{uv}^*(\lambda)$, $S_{vu}^*(\lambda)$, $S_{gu}^*(\lambda)$, $S_{gr}^*(\lambda)$ и $S_{gr}^*(\lambda)$. Отличие заключается в том, что они определены для решетчатых сигналов в функции псевдочастоты $\lambda = 2 \operatorname{tg}(\omega T/2)/T$ (см. § 3.4).

Запишем выражение для дисперсии ошибки дискретной системы при условии отсутствия корреляции между полезным сигналом и помехой и при $H_0(z) = 1$:

$$\begin{aligned} D_e &= \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{|1 - H^*(j\lambda)|^2}{1 + \lambda^2 T^2/4} S_g^*(\lambda) d\lambda + \\ &+ \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{|H^*(j\lambda)|^2}{1 + \lambda^2 T^2/4} S_v^*(\lambda) d\lambda. \end{aligned} \quad (5.15)$$

Сравнение этого выражения с (5.10) показывает, что отличие заключается в наличии множителя $T/(1 + \lambda^2 T^2/4)$. Поэтому для минимизации дисперсии ошибки можно воспользоваться формулой (5.4) при условии использования спектральных плотностей вида $S_g^*(\lambda)/(1 + \lambda^2 T^2/4)$ и $S_v^*(\lambda)/(1 + \lambda^2 T^2/4)$. В результате может быть получено выражение, определяющее оптимальную частотную передаточную функцию замкнутой дискретной системы

$$H^*(j\lambda) = \frac{1 + j\lambda T/2}{\psi^*(j\lambda)} \left[\frac{S_g^*(\lambda)}{\psi^*(-j\lambda)(1 + j\lambda T/2)} \right]_+. \quad (5.16)$$

Знак плюс у квадратной скобки означает, что необходимо выделить у выражения, находящегося в квадратных скобках, составляющие, соответствующие полюсам, находящимся в верхней полуплоскости переменной λ^* .

Формула (5.16) может быть распространена и на другие случаи винеровской фильтрации: при наличии корреляционной связи между полезным сигналом и помехой, а также в случае, когда $H_0(z) \neq 1$.

Таким образом, процедура нахождения оптимальной частотной передаточной функции для дискретных систем оказывается аналогичной непрерывным системам, если вместо обычной круговой частоты использовать псевдочастоту.

*) Этот способ определения оптимальной передаточной функции предложен А.В.Небыловым.

Переход к передаточной функции замкнутой системы $H(z)$ может быть сделан подстановкой $j\lambda = 2(z - 1) / [(z + 1)T]$.

После нахождения оптимальной передаточной функции может быть найдено минимальное значение дисперсии ошибки в соответствии с формулой (5.15).

Определение периода дискретности и вторичная оптимизация. Найденное значение дисперсии ошибки обычно является функцией периода дискретности, т.е. $D_e = D_e(T)$. Это дает возможность так выбрать период дискретности T , чтобы дисперсия ошибки D_e или среднеквадратичная ошибка σ_e не превышали бы допустимых значений.

В некоторых случаях, особенно при реализации цифрового дифференцирования решетчатых последовательностей, может оказаться, что дисперсия ошибки имеет минимум при некотором значении периода дискретности. Это дает возможность произвести вторичную оптимизацию винеровского фильтра и найти период дискретности, минимизирующий дисперсию ошибки на основании выражения

$$\partial D_e(T) / \partial T = 0. \quad (5.17)$$

Если этот период дискретности возможно реализовать в микроЭВМ, то будет получено оптимальное решение задачи фильтрации в цифровом виде.

В тех случаях, когда помеха представляет собой шумы квантования на входе системы (см. § 3.4), дисперсия ошибки оказывается функцией двух переменных: периода дискретности T и цены единицы младшего разряда входного преобразователя δ_1 , т.е. $D_e = D_e(T, \delta_1)$. Тогда задача выбора T и δ_1 может быть решена комплексно из условия, чтобы среднеквадратичное значение ошибки не превышало заданного значения.

Рассмотрим, например, задачу оптимального выделения полезного сигнала из его смеси с помехой. Для полезного сигнала примем спектральную плотность вида

$$S_g^*(\lambda) = \frac{C(1 + \lambda^2 T^2/4)}{\lambda^2}.$$

Здесь C — величина, пропорциональная среднему квадрату скорости изменения полезного сигнала. В качестве помехи примем дискретный белый шум с спектральной плотностью $S_v^*(\lambda) = D_v$, где D_v — дисперсия белого шума.

Рассмотрим суммарную спектральную плотность

$$\begin{aligned} S_r^*(\lambda) &= S_g^*(\lambda) + S_v^*(\lambda) = \frac{C(1 + \lambda^2 T^2/4)}{\lambda^2} + D_v = \\ &= C \frac{1 + \lambda^2 \tau^2}{\lambda^2} = \psi^*(j\lambda) \psi^*(-j\lambda), \end{aligned}$$

где $\tau^2 = D_v / C + T^2/4$.

Далее находим

$$\psi^*(j\lambda) = \sqrt{C} \frac{1+j\lambda\tau}{j\lambda} ; \quad \psi^*(-j\lambda) = \sqrt{C} \frac{1-j\lambda\tau}{-j\lambda} .$$

В соответствии с формулой (5.16) имеем

$$\begin{aligned} H^*(j\lambda) &= \frac{(1+j\lambda T/2)j\lambda}{\sqrt{C}(1+j\lambda\tau)} \left[\frac{C(1+\lambda^2 T^2/4)(-j\lambda)}{\lambda^2 \sqrt{C}(1-j\lambda\tau)(1+j\lambda T/2)} \right]_+ = \\ &= \frac{(1+j\lambda T/2)j\lambda}{(1+j\lambda\tau)} \left[\frac{1-j\lambda T/2}{j\lambda(1-j\lambda\tau)} \right]_+ . \end{aligned}$$

Выражение в квадратных скобках может быть разложено на слагаемые

$$\frac{1-j\lambda T/2}{j\lambda(1-j\lambda\tau)} = \frac{1}{j\lambda} + \frac{1+T/2\tau}{1-j\lambda\tau} .$$

Отбрасывая второе слагаемое, у которого полюс лежит в нижней полуплоскости, имеем

$$H^*(j\lambda) = \frac{1+j\lambda T/2}{1+j\lambda\tau} .$$

Переход к дискретной передаточной функции $H(z)$ может быть сделан подстановкой $j\lambda = 2(z-1)/(z+1)T$.

После нахождения $H^*(j\lambda)$ по формуле (5.16) можно определить минимальную дисперсию ошибки

$$\begin{aligned} D_{e \min} = \sigma_{e \min}^2 &= \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{(\tau - T/2)^2 C d\lambda}{|1+j\lambda\tau|^2} + \\ &+ \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{D_v d\lambda}{|1+j\lambda\tau|^2} . \end{aligned}$$

Используя Приложение 3, находим

$$\sigma_{e \min}^2 = T[C(\tau - T/2)^2 + D_v]/2\tau .$$

После подстановки значения τ дисперсия ошибки будет зависеть от трех величин: C , D_v и T . Выбирая период дискретности, можно выполнить условие, чтобы среднеквадратичная ошибка была меньше заданного значения.

Если помеха на входе вызвана явлением квантования по уровню на входе системы, то $D_v = \delta_1^2/12$, где δ_1 — цена единицы младшего разряда входного преобразователя. В этом случае дисперсия ошибки будет зависеть от величин C , T и δ_1 . Здесь можно ограничить среднеквадратичную ошибку, изменяя как период дискретности T , так и цену младшего разряда δ_1 .

§ 5.3. РОБАСТНЫЕ ЦИФРОВЫЕ СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ

Общие сведения. Построение достаточно достоверных статистических моделей входных воздействий в большинстве случаев сделать не удастся. Это объясняется как недостаточностью имеющейся обычно априорной информации, так тем, что эти модели могут непрерывно изменяться в зависимости от изменения внешних условий.

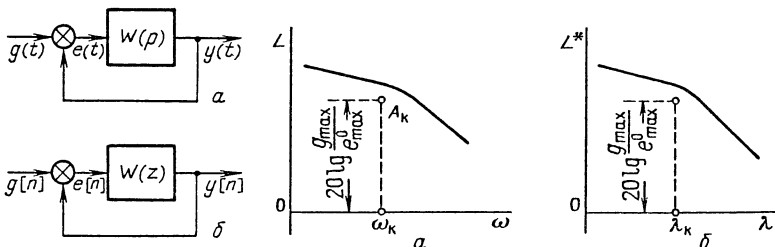
Использование адаптивных методов может быть сделано не всегда вследствие неизбежного усложнения системы управления, а также вследствие часто имеющих место низких динамических качеств контура адаптации.

В свете этого представляется целесообразным развитие таких методов построения систем управления, которые обеспечивают приемлемую их работу при изменении статистических характеристик входных воздействий в широких пределах. Такие системы получили название робастных (robust), что в переводе означает "крепкий, сильный".

Для построения робастных систем требуется минимум априорных сведений о входных воздействиях. Обычно эти сведения ограничиваются установлением максимальных или среднеквадратичных значений самих воздействий и их первых производных (скоростей, ускорений и т.д.). Ниже развиваются методы, обеспечивающие получение заданной точности системы на основе формулирования требований к низкочастотной части л.а.х. разомкнутой системы.

Обеспечение требуемой точности. Ошибка воспроизведения полезного сигнала (рис. 5.3, а) равна разности $e(t) = g(t) - y(t)$. В цифровой форме аналогичная зависимость может быть записана для решетчатых функций $e[n] = g[n] - y[n]$ (рис. 5.3, б).

Требования по ограничению максимальной ошибки наиболее просто могут быть сформулированы для гармонического сигнала $g(t) = g_{\max} \sin(\omega_{\kappa} t + \psi_{\kappa})$. Предполагается, что амплитуда g_{\max} и



Р и с. 5.3. Непрерывная и дискретная системы управления

Р и с. 5.4. Контрольные точки для л.а.х. из условия точности

частота ω_k известны, а начальная фаза ψ_k имеет случайное значение.

Амплитуда ошибки

$$e_{\max} = g_{\max} / |1 + W(j\omega_k)|. \quad (5.18)$$

Так как в реальных системах $e_{\max} \ll g_{\max}$, то можно положить

$$e_{\max} \approx g_{\max} / |W(j\omega_k)|. \quad (5.19)$$

Из формулы (5.19) вытекают требования к низкочастотной части л.а.х. разомкнутой системы. Для того чтобы заданное гармоническое воздействие воспроизводилось с максимальной ошибкой не более e_{\max}^0 , л.а.х. проектируемой системы должна проходить не ниже контрольной точки A_k с координатами

$$\omega = \omega_k; \quad L(\omega_k) = 20 \lg |W(j\omega_k)| = 20 \lg \frac{g_{\max}}{e_{\max}^0}. \quad (5.20)$$

На рис. 5.4, а построена эта контрольная точка и низкочастотная часть л.а.х. системы, удовлетворяющей требованию, чтобы максимальная ошибка не превышала бы заданного значения e_{\max}^0 .

В дискретных системах все оказывается аналогичным при выполнении условия $\omega_k \ll 1/T$. Тогда и действительная частота входного воздействия ω_k может быть заменена псевдочастотой

$$\lambda_k = 2 \operatorname{tg}(\omega_k T/2)/T \approx \omega_k. \quad (5.21)$$

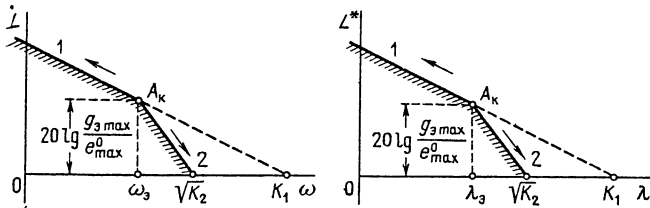
Ошибка системы в дискретные моменты времени $t = nT$ не будет превышать заданного максимального значения e_{\max}^0 , если л.а.х. системы будет проходить не ниже контрольной точки A_k с координатами

$$\lambda = \lambda_k; \quad L^*(\lambda_k) = 20 \lg |W^*(j\lambda_k)| = 20 \lg \frac{g_{\max}}{e_{\max}^0}, \quad (5.22)$$

где $W^*(j\lambda_k)$ — частотная передаточная функция разомкнутой системы при значении псевдочастоты $\lambda = \lambda_k$. Построение показано на рис. 5.4, б.

Заметим, что принятое условие $\omega_k \ll 1/T$ должно выполняться в цифровых системах всегда, так как иначе полезное воздействие не может быть воспроизведено достаточно точно.

Пусть на входе системы известны максимальное по модулю значение первой производной \dot{g}_{\max} (скорости) и максимальное по модулю значение второй производной \ddot{g}_{\max} (ускорения) входного сигнала $g(t)$. При этом не накладывается никаких других ограничений на вид этого сигнала. Это может быть стационарный случайный сигнал с любой спектральной плотностью. Априорная информация здесь сведена к минимуму и определяется величина-



Р и с. 5.5. Запретные области по условиям точности для астатических систем

ми \dot{g}_{\max} и \ddot{g}_{\max} , которые в большинстве случаев могут быть легко оценены.

Для этих величин подберем эквивалентный режим гармонического воздействия. Рассмотрим такой эквивалентный режим

$$g_3(t) = g_{3\max} \sin(\omega_3 t + \psi_3), \quad (5.23)$$

который характеризуется максимальным значением первой производной $\omega_3 g_{3\max} = \dot{g}_{\max}$ и максимальным значением второй производной $\omega_3^2 g_{3\max} = \ddot{g}_{\max}$. Для этого режима можно определить частоту и амплитуду

$$\omega_3 = \ddot{g}_{\max} / \dot{g}_{\max}, \quad g_{3\max} = \dot{g}_{\max}^2 / \ddot{g}_{\max} \quad (5.24)$$

при произвольном значении начальной фазы ψ_3 . В соответствии с этими формулами на рис. 5.5, а построена контрольная точка A_k .

Если по-прежнему $\omega_3 \ll 1/T$, то частоту ω_3 можно заменить на псевдо частоту $\lambda_3 \approx \omega_3$. По вычисленным значениям ω_3 и $g_{3\max}$ при заданной величине максимальной ошибки e_{\max}^0 можно построить для л.а.х. $L^*(\lambda)$ контрольную точку с координатами (рис. 5.5, б)

$$\lambda_3 \approx \ddot{g}_{\max} / \dot{g}_{\max}, \quad (5.25)$$

$$L^*(\lambda_3) = 20 \lg |W^*(\lambda_3)| = 20 \lg [\dot{g}_{\max}^2 / (e_{\max}^0 \ddot{g}_{\max})].$$

Пусть теперь максимальное значение первой производной входного воздействия по-прежнему равно \dot{g}_{\max} , а максимальное значение второй производной будет меньше \ddot{g}_{\max} . Тогда в соответствии с (5.25) частота контрольной точки будет пропорционально уменьшаться, а амплитуда $g_{3\max}$ в соответствии с (5.23) будет пропорционально возрастать. При этом контрольная точка на рис. 5.5, а будет перемещаться в область более низких частот (влево) по прямой, имеющей отрицательный наклон 20 дБ/дек (единичный наклон). Если максимальное значение второй производной устремить к нулю, то частота $\omega_k \rightarrow 0$. Это соответствует режиму работы системы с постоянной скоростью $\dot{g} = \dot{g}_{\max}$. В этом случае будет получено предельное значение общего коэффициента усиления

разомкнутой системы с астатизмом первого порядка (добротность по скорости)

$$K_1 = \dot{g}_{\max} / e_{\max}^0. \quad (5.26)$$

Ниже этого значения в реальной системе с астатизмом первого порядка нельзя иметь общий коэффициент усиления при заданных значениях e_{\max}^0 и \dot{g}_{\max} .

Пусть теперь максимальное значение второй производной входного воздействия будет по-прежнему равно \ddot{g}_{\max} , а максимальное значение первой производной будет меньше \dot{g}_{\max} . Аналогичными рассуждениями можно показать, что контрольная точка A_K (рис. 5.5, а) будет перемещаться в область более высоких частот, двигаясь по прямой, имеющей отрицательный наклон 40 дБ/дек (двойной наклон). Квадрат частоты точки пересечения этой прямой с осью абсцисс равен предельному коэффициенту усиления разомкнутой системы (добротности по ускорению).

$$K_2 = \ddot{g}_{\max} / e_{\max}^0. \quad (5.27)$$

Ниже этого предельного значения не может быть выбран общий коэффициент усиления разомкнутой системы с астатизмом второго порядка. Область, расположенная ниже двух пересекающихся прямых с отрицательными наклонами 20 дБ/дек и 40 дБ/дек, представляет собой запретную область для л.а.х. проектируемой системы. При работе с ограниченными значениями первой и второй производных от входного воздействия, не превышающими \dot{g}_{\max} и \ddot{g}_{\max} , максимальная ошибка системы не будет превышать заданного значения e_{\max}^0 .

Все эти рассуждения будут справедливыми и для дискретных систем (рис. 5.5, б), если вместо обычной частоты ω рассматривать псевдочастоту λ . При этом должно выполняться условие, что вся запретная область располагается левее частоты $\omega = 1/T$. Тогда можно положить для этой области $\lambda \approx \omega$.

Если считать, что отношения между максимальными и среднеквадратичными значениями одинаковы или примерно одинаковы для процессов $g(t)$, $y(t)$ и $e(t)$, то координаты контрольной точки в непрерывном случае (рис. 5.4, а) могут быть найдены для централированных случайных процессов из выражений

$$\omega_3 = \sqrt{D_2/D_1}; \quad L(\omega_3) = 20 \lg(D_1/\sqrt{D_e^0 D_2}). \quad (5.28)$$

Здесь D_1 — дисперсия первой производной входного сигнала (скорости), D_2 — дисперсия второй производной входного сигнала (ускорения), D_e^0 — допустимая дисперсия ошибки воспроизведения полезного входного сигнала. Заметим, что соотношения между максимальными и среднеквадратичными значениями оказываются строго одинаковыми для нормального закона распределения.

Как и ранее, для дисперсии второй производной входного сигнала, меньшей, чем заданное значение D_2 , контрольная точка A_k (рис. 5.4, а) будет смещаться влево по прямой с отрицательным наклоном 20 дБ/дек. Если $D_2 \rightarrow 0$, то вместо (5.26) получим

$$K_1 = \sqrt{D_1/D_e^0}. \quad (5.29)$$

При уменьшении дисперсии первой производной входного сигнала контрольная точка будет двигаться вправо по прямой с отрицательным наклоном 40 дБ/дек. Если $D_1 \rightarrow 0$, то вместо (5.27) получим

$$K_2 = \sqrt{D_2/D_e^0}. \quad (5.30)$$

Прямые линии с отрицательными наклонами 20 дБ/дек и 40 дБ/дек формируют, как и ранее, запретную область для л.а.х. разомкнутой системы (рис. 5.5).

Из вида запретной области следует, что задание ограничений только на первую и вторую производную при отсутствии ограничения на сам сигнал $g(t)$ или $g[n]$ приводит к необходимости использовать только астатические системы управления.

Для того чтобы в реальной системе управления действительная точность воспроизведения задающего воздействия соответствовала принятой для построения запретной области, важным условием оказывается наличие в замкнутой системе достаточного запаса устойчивости. Выполнение обычных критериев запаса устойчивости [26] оказывается достаточным для того, чтобы построенная запретная область для л.а.х. (рис. 5.5) гарантировала получение желаемой точности.

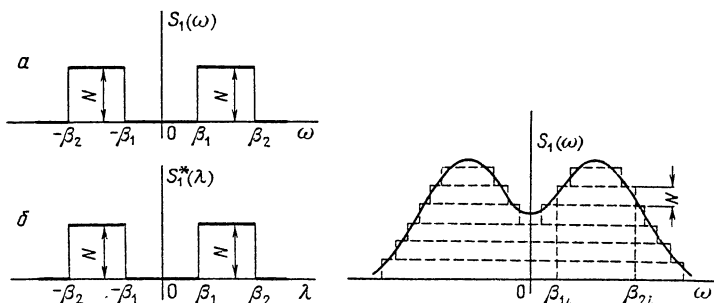
Использование типовых передаточных функций. Рассмотренная методика обеспечения требуемой точности воспроизведения полезного сигнала может быть обоснована более строго для некоторых важных с практической точки зрения случаев. Это, в частности, относится к системам, в которых используются типовые передаточные функции [23, 25, 26].

Пусть спектральная плотность первой производной задающего воздействия имеет вид, изображенный на рис. 5.6, а. Дисперсия этой производной (скорости изменения входного воздействия)

$$D_1 = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_1(\omega) d\omega = \frac{1}{\pi} \int_{\beta_1}^{\beta_2} N d\omega = \frac{N(\beta_2 - \beta_1)}{\pi}. \quad (5.31)$$

Дисперсия второй производной (ускорения изменения входного воздействия)

$$D_2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_1(\omega) \omega^2 d\omega = \frac{1}{\pi} \int_{\beta_1}^{\beta_2} N \omega^2 d\omega = \frac{N(\beta_2^3 - \beta_1^3)}{3\pi}. \quad (5.32)$$



Р и с. 5.6. Спектральные плотности непрерывного и дискретного сигнала

Р и с. 5.7. Замена произвольной спектральной плотности суммой равномерных спектров

В соответствии с этими значениями дисперсий можно построить запретную область вида, изображенного на рис. 5.5, *a*. Для нее частота контрольной точки

$$\omega_3 = \sqrt{D_2/D_1} = \sqrt{(\beta_1^2 + \beta_1\beta_2 + \beta_2^2)/3}. \quad (5.33)$$

В случае периодического режима $\beta_1 = \beta_2 = \beta$. Тогда $\omega_3 = \beta$. Предельный коэффициент усиления системы с астатизмом первого порядка определяется формулой (5.29), а системы с астатизмом второго порядка — формулой (5.30). При этом, конечно, следует знать допустимую дисперсию ошибки D_e^0 .

Выберем систему с астатизмом первого порядка простейшего вида с передаточной функцией разомкнутой системы равной $W(p) = K_1/p$, где добротность K_1 определяется формулой (5.29). Проверим правильность сделанного выше утверждения о том, что в этой системе действительная дисперсия ошибки не будет превышать заданного значения.

Спектральная плотность ошибки

$$S_e(\omega) = \frac{\omega^2}{K_1^2 + \omega^2} \cdot \frac{N}{\omega^2} = \frac{N}{K_1^2 + \omega^2}; \quad (\beta_1 \leq \omega \leq \beta_2). \quad (5.34)$$

Дисперсия ошибки в рассматриваемой системе

$$D_e = \frac{1}{\pi} \int_{\beta_1}^{\beta_2} \frac{Nd\omega}{K_1^2 + \omega^2} = \frac{N}{\pi K_1} \left(\arctg \frac{\beta_2}{K_1} - \arctg \frac{\beta_1}{K_1} \right). \quad (5.35)$$

Так как $|\arctg x| < |x|$, то из последнего равенства вытекает условие

$$D_e < \frac{N(\beta_2 - \beta_1)}{\pi K_1^2} = \frac{D_1}{K_1^2} = D_e^0. \quad (5.36)$$

Таким образом, при любых значениях β_1 и β_2 дисперсия ошибки в выбранной системе управления не превышает заданного значения. Только в предельном случае, когда $\beta_1 \rightarrow \beta$ и $\beta_2 \rightarrow \beta$, что соответствует случаю гармонического входного сигнала, получается условие $D_e \rightarrow D_e^0$.

Аналогичный пример может быть рассмотрен и для дискретного случая при условии выполнения неравенств $\beta_1 \ll 1/T$ и $\beta_2 \ll 1/T$, где T — период дискретности.

Покажем теперь, что полученный результат можно распространить на спектральную плотность $S_1(\omega)$ любого вида. На рис. 5.7 изображена спектральная плотность $S_1(\omega)$ произвольного вида, приближенно представляемая в виде совокупности равномерных спектров высотой N и протяженностью от $\pm\beta_{1i}$ до $\pm\beta_{2i}$, где $i = 1, 2, \dots, n$. Дисперсия скорости для совокупности равномерных спектров

$$D_1 = \frac{N}{\pi} \sum_{i=1}^n (\beta_{2i} - \beta_{1i}). \quad (5.37)$$

Дисперсия ускорения

$$D_2 = \frac{N}{3\pi} \sum_{i=1}^n (\beta_{2i}^3 - \beta_{1i}^3). \quad (5.38)$$

Действительная дисперсия ошибки для совокупности равномерных спектров в соответствии с (5.35) будет

$$D_e = \frac{N}{\pi K_1} \sum_{i=1}^n \left(\operatorname{arctg} \frac{\beta_{2i}}{K_1} - \operatorname{arctg} \frac{\beta_{1i}}{K_1} \right). \quad (5.39)$$

Аналогично изложенному выше

$$D_e < \frac{N}{\pi K_1^2} \sum_{i=1}^n (\beta_{2i} - \beta_{1i}) = \frac{D_1}{K_1^2} = D_e^0. \quad (5.40)$$

При бесконечном уменьшении высоты равномерных спектров и при одновременном бесконечном увеличении их числа исходную спектральную плотность $S_1(\omega)$ можно аппроксимировать с любой степенью точности. Следовательно, доказана справедливость утверждения, что действительная дисперсия ошибки $D_e < D_e^0$ при любом виде спектральной плотности $S_1(\omega)$, но при ограниченных значениях D_1 и D_2 .

Следует, однако, подчеркнуть, что рассмотренное доказательство справедливо только для спектральных плотностей (рис. 5.7), имеющих в области положительных частот лишь один пик. При более сложном виде спектральной плотности, например при наличии двух пиков, запретная область должна формироваться более

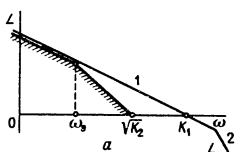
сложным образом с использованием двухчастотного эквивалентного режима.

В работе [23] показано, что и при других видах передаточной функции разомкнутой системы будет выполняться условие $D_e \leq \leq D_e^0$, если л.а.х. системы не будет заходить в запретную область и при условии, что в системе имеется достаточно большой запас устойчивости (в пределах обычных рекомендаций). Однако при использовании асимптотических л.а.х. следует учитывать их отличие от действительных л.а.х. Поэтому при совпадении точек излома асимптотической л.а.х. и запретной области следует вводить дополнительное поднятие вверх асимптотической л.а.х. на 3 дБ. Это показано на рис. 5.8. Если изломы л.а.х. достаточно удалены от излома запретной области (рис. 5.8, а и в), то асимптота может сливаться с границей запретной области, а при совпадении (рис. 5.8, б) следует ввести дополнительный подъем на 3 дБ.

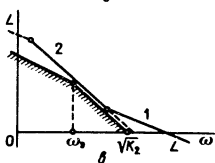
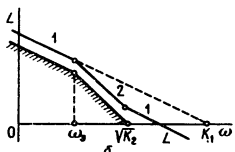
В работе [23] также показано, что при использовании в качестве исходной спектральной плотности ускорения входного воздействия результаты оказываются аналогичными.

В статических системах наклон первой асимптоты л.а.х. нулевой. Поэтому условие нахождения всей л.а.х. выше запретных областей (рис. 5.8) не может быть выполнено ни при каком значении общего коэффициента усиления. В связи с этим статические системы могут работать только при наложении дополнительного ограничения на модуль или дисперсию входного сигнала.

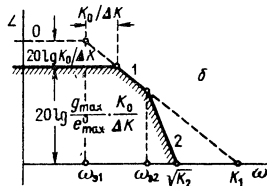
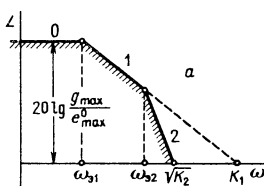
Можно показать [23], что в этом случае запретные области для л.а.х. имеют вид, изображенный на рис. 5.9. Если в системе используется единичная обратная связь, то запретная область (рис. 5.9, а)



Р и с. 5.8. Расположение низкочастотной части л.а.х. относительно запретной области по точности



Р и с. 5.9. Запретные области по условиям точности для статических систем



должна формироваться на основании выражений

$$\begin{aligned} \omega_{\varepsilon 1} &= \dot{g}_{\max} / g_{\max}, & \omega_{\varepsilon 2} &= \ddot{g}_{\max} / \dot{g}_{\max}, \\ 20 \lg K_0 &= L(\omega_{\varepsilon 1}) = 20 \lg (g_{\max} / e_{\max}^0), \end{aligned} \quad (5.41)$$

где g_{\max} , \dot{g}_{\max} , \ddot{g}_{\max} — максимальные значения входного воздействия и его первой и второй производных. В случае, если известны дисперсии этих величин, аналогичные формулы имеют вид

$$\begin{aligned} \omega_{\varepsilon 1} &= \sqrt{D_1 / D_0}, & \omega_{\varepsilon 2} &= \sqrt{D_2 / D_1}, \\ 20 \lg K_0 &= L(\omega_{\varepsilon 1}) = 20 \lg \sqrt{D_0 / D_e^0}. \end{aligned} \quad (5.42)$$

Если в системе используется неединичная обратная связь (масштабирование выходного сигнала), то высота запретной области может быть снижена пропорционально отношению $K_0 / \Delta K_0$, где ΔK_0 — возможное отклонение (нестабильность) общего коэффициента усиления от его номинального значения. Это показано на рис. 5.9, б.

Рассмотренные доказательства выполнения требования по точности при неизвестной форме спектральной плотности входного воздействия можно распространить и на цифровые системы. Непременным условием здесь должно быть расположение всей запретной области левее частоты $\omega = 1/T$. Тогда для интервала частот, в котором располагается запретная область, будет выполняться условие $\lambda \approx \omega$.

Обобщая изложенное по методике построения запретных областей для л.а.х., рассмотрим возможные случаи применительно к цифровым системам управления.

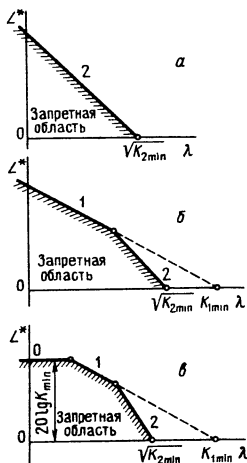
При задании ограниченного значения только дисперсии второй производной задающего воздействия D_2 и при неограниченных значениях дисперсии первой производной D_1 и дисперсии самого задающего воздействия D_0 запретная область сводится к прямой двойного наклона (-40 дБ/дек), что показано на рис. 5.10, а.

Как следует из вида запретной области, задача воспроизведения входного сигнала может быть решена только системой управления с астатизмом второго порядка. Минимальное значение общего коэффициента усиления разомкнутой системы определяется выражением

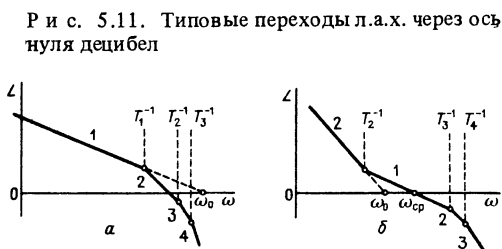
$$K_{2 \min} = \sqrt{D_2 / D_e^0}, \quad (5.43)$$

где D_e^0 — заданное значение дисперсии ошибки.

При задании ограниченных значений дисперсий второй производной D_2 и первой производной D_1 и неограниченном значении дисперсии самого задающего воздействия D_0 запретная область, образованная прямой двойного наклона, усекается прямой единичного наклона. Это показано на рис. 5.10, б. Здесь задача воспроизведе-



Р и с. 5.10. Возможные формы запретных областей по точности



Р и с. 5.11. Типовые переходы л.а.х. через ось нуля децибел

ния задающего воздействия может осуществляться как системой управления с астатизмом второго порядка, так и системой с астатизмом первого порядка. При астатизме второго порядка минимальное значение общего коэффициента усиления разомкнутой системы определяется по-прежнему формулой (5.43). При астатизме первого порядка минимальное значение общего коэффициента усиления

$$K_{1 \min} = \sqrt{D_1/D_e^0} \quad (5.44)$$

При задании ограниченных значений дисперсий D_2 , D_1 и D_0 запретная область, образованная прямыми с двойным и единичным наклонами (рис. 5.10, б), дополнительно усекается прямой нулевого наклона. Это показано на рис. 5.10, в. В этом случае задача воспроизведения задающего воздействия может осуществляться как системами управления с астатизмом второго и первого порядков, так и статической системой. Для астатических систем минимальные значения общего коэффициента усиления определяются формулами (5.43) и (5.44). Для статической системы минимальное значение общего коэффициента усиления для случая единичной главной обратной связи

$$K_{0 \min} = \sqrt{D_0/D_e^0} \quad (5.45)$$

и для случая неединичной обратной связи

$$K_{0 \min} = \frac{\Delta K_0}{K_0} \sqrt{D_0/D_e^0} \quad (5.46)$$

Оба случая формирования запретной области показаны на рис. 5.9.

Обеспечение запаса устойчивости. Если точность системы определяется видом частотной передаточной функции разомкнутой системы (или л.а.х.) в ее низкочастотной области, то запас устойчивости определяется ее видом в среднечастотной области, т.е. в районе, где модуль частотной передаточной функции разомкнутой системы близок к единице. При использовании л.а.х. запас устойчивости может быть оценен по району перехода ее оси нуля децибел.

В непрерывных системах по виду в среднечастотной области различают л.а.х. "несимметричного" вида (рис. 5.11, а) и "симметричного" вида (рис. 5.11, б).

Если перед переходом оси нуля децибел имеется достаточно длинный участок асимптотической л.а.х. (протяженность, определяемая по отношению частот конца и начала участка не менее 8 — 10), то будет иметь место случай, изображенный на рис. 5.11, а. Тогда для выполнения требования по запасу устойчивости, определяемого по величине показателя колебательности (рекомендуемое значение $M = 1,1 - 1,5$), необходимо выполнение неравенства

$$\sum_{i=1}^n T_i \leq \frac{1}{\omega_0} \frac{M^2 + M\sqrt{M^2 - 1}}{2}, \quad (5.47)$$

где ω_0 — базовая частота л.а.х. В простейшем случае, когда рассматривается система с астатизмом первого порядка, $\omega_0 = K_1$, где K_1 — добротность по скорости. Для статических систем и систем с астатизмом второго порядка вид л.а.х. в низкочастотной области большого значения не имеет.

Формула (5.47) является приближенной. При $n = 1$ и любом показателе колебательности или при $M = 1$ и любом числе постоянных времени она становится точной.

Таким образом, в районе перехода оси нуля децибел передаточная функция разомкнутой системы может быть записана в виде

$$W(p) = \frac{\omega_0}{p(1 + T_1 p)(1 + T_2 p) \dots}. \quad (5.48)$$

Неравенство (5.47) определяет требования к сумме всех постоянных времени, входящих в знаменатель (5.48).

Если перед переходом оси нуля децибел имеется достаточно длинный участок асимптотической л.а.х. с двойным наклоном (рис. 5.11, б), то для выполнения требований по показателю колебательности необходимо выполнение условий

$$T_2 = \frac{1}{\omega_0} \sqrt{\frac{M}{M-1}}, \quad \sum_{i=3}^n T_i \leq \frac{1}{\omega_0} \frac{\sqrt{M(M-1)}}{M+1} \quad (5.49)$$

или, в другой форме,

$$T_2 \geq \frac{1}{\omega_{\text{ср}}} \frac{M}{M-1}, \quad \sum_{i=3}^n T_i \leq \frac{1}{\omega_{\text{ср}}} \frac{M}{M+1}. \quad (5.50)$$

Здесь ω_0 — базовая частота л.а.х., $\omega_{\text{ср}}$ — частота среза. В простейшем случае, когда рассматривается система с астатизмом второго порядка, имеет место равенство $\omega_0 = \sqrt{K_2}$, где K_2 — добротность по ускорению.

Вид л.а.х. в низкочастотной ее части значения практически не имеет, если участок с двойным наклоном имеет достаточную протяженность (не менее 8–10).

Таким образом, в районе средних частот передаточная функция разомкнутой системы может быть представлена в виде

$$W(p) = \frac{\omega_0^2(1 + T_2p)}{p^2(1 + T_3p)(1 + T_4p) \dots}. \quad (5.51)$$

Полное выражение для передаточной функции разомкнутой системы может быть получено после "сшивания" низкочастотной части л.а.х., построенной по условиям требуемой части, и ее среднечастотной части (5.48) или (5.51).

Заметим, что все рассуждения по поводу обеспечения требуемого запаса устойчивости, которые были сделаны выше, относятся лишь к так называемым минимально-фазовым системам. Последнее предполагает, что все нули и полюсы передаточной функции находятся в левой половине или в начале координат плоскости комплексной величины p . Другие случаи рассмотрены в работах [22, 23].

Условия получения заданного запаса устойчивости могут быть распространены и на цифровые системы. При этом должно выполняться очевидное условие для периода дискретности $(T/2) < 1/\omega_{\text{ср}}$. Здесь $\omega_{\text{ср}}$ — частота среза, т.е. частота, при которой л.а.х. пересекает ось нуля децибел. При невыполнении последнего неравенства в системе управления вообще не может быть получен устойчивый режим.

При выполнении условия $(T/2) < 1/\omega_{\text{ср}}$ можно положить $\lambda \approx \omega$ и строить всю л.а.х. в функции псевдочастоты. При этом формулы, ограничивающие сумму постоянных времени, (5.47), (5.49) и (5.50) сохраняют свое значение, однако в этой сумме следует учесть дополнительную постоянную времени $T/2$.

Более подробно построение цифровых систем с заданным запасом устойчивости рассмотрено в [23]. В этой работе кроме построения л.а.х. с заданным запасом устойчивости и экстраполятором нулевого порядка рассмотрен также случай использования экстраполятора первого порядка.

§ 5.4. ПРОГРАММЫ ДЛЯ МИКРОЭВМ, РЕАЛИЗУЮЩИЕ АЛГОРИТМЫ УПРАВЛЕНИЯ

Цифровая коррекция. В цифровых системах могут использоваться как непрерывные, так и цифровые корректирующие средства. Выбор непрерывных корректирующих средств в цифровых системах может делаться известными методами, разработанными для непрерывных систем, например, на основе использования л.а.х.

Цифровые корректирующие средства могут быть реализованы на ЦВМ, включенной в контур управления (рис. 3.8). Это усложняет алгоритм работы ЦВМ, но позволяет избавиться от необходимости использовать непрерывные корректирующие устройства.

Требуемый алгоритм работы ЦВМ определяется передаточной функцией $D(z)$. Дискретные корректирующие средства могут быть также осуществлены на дискретных фильтрах, построенных на различных ячейках памяти.

Пусть тем или иным путем найдем желаемая дискретная передаточная функция разомкнутой системы

$$W_{\text{ж}}(z) = \frac{H_{\text{ж}}(z)}{1 - H_{\text{ж}}(z)} = D(z) W_{\text{н}}(z), \quad (5.52)$$

где $H_{\text{ж}}(z)$ — желаемая передаточная функция замкнутой системы, а $W_{\text{н}}(z)$ — передаточная функция исходной нескорректированной системы. Тогда искомая передаточная функция ЦВМ или дискретного фильтра имеет вид

$$D(z) = \frac{W_{\text{ж}}(z)}{W_{\text{н}}(z)} = \frac{H_{\text{ж}}(z)}{1 - H_{\text{ж}}(z)} \cdot \frac{1}{W_{\text{н}}(z)}. \quad (5.53)$$

Формирование желаемой функции $H_{\text{ж}}(z)$ должно производиться с учетом некоторых ограничений. Необходимо, чтобы передаточная функция $H_{\text{ж}}(z)$ содержала в качестве своих нулей все те нули передаточной функции $W_{\text{н}}(z)$, модуль которых равен или больше единицы. Кроме того, необходимо, чтобы выражение $1 - H_{\text{ж}}(z)$ содержало в качестве своих нулей все те полюсы $W_{\text{н}}(z)$, модуль которых равен или больше единицы.

В полученном выражении для $D(z)$ не должно быть полюсов лежащих вне круга единичного радиуса или на самом круге, кроме полюсов, равных $+1$. Последнее соответствует введению в программу ЦВМ интегрирования входного сигнала.

Невыполнение этих условий вызывает нарушение требований к грубости системы и ее неустойчивость, так как приводит к неустойчивым линейным программам ЦВМ, которые должны реализовать получающуюся по формуле (5.53) передаточную функцию $D(z)$. Кроме того, получающаяся дробно-рациональная передаточная

функция $D(z)$ не должна иметь степень числителя выше, чем степень знаменателя, так как это приводит к необходимости знания будущего значения входного сигнала, что не может быть реализовано.

Вместо формулы (5.53) может применяться соотношение, связывающее дискретные частотные передаточные функции желаемой и исходной систем

$$D^*(j\lambda) = W_{\text{ж}}^*(j\lambda)/W_{\text{н}}^*(j\lambda) \quad (5.54)$$

или соответствующие им логарифмические частотные характеристики

$$20 \lg |D^*(j\lambda)| = L_{\text{ж}}^*(\lambda) - L_{\text{н}}^*(j\lambda). \quad (5.55)$$

После определения $D^*(j\lambda)$ подстановкой $j\lambda = 2(z-1)/(z+1)T$ можно получить передаточную функцию $D(z)$.

Сформулированные выше ограничения по отношению к выражению (5.54) имеют следующий характер. Необходимо, чтобы передаточная функция $W_{\text{ж}}^*(j\lambda)$ содержала в качестве своих нулей и полюсов по переменной $j\lambda$ все те нули и полюсы передаточной функции $W_{\text{н}}^*(j\lambda)$, которые лежат в правой полуплоскости. Кроме того, необходимо, чтобы получающаяся дробно-рациональная функция $D^*(j\lambda)$ имела степень числителя не больше, чем степень знаменателя.

Реализация на ЦВМ передаточной функции $D(z)$ должна выполняться в соответствии с рассмотренными в § 4.2 условиями построения цифровых фильтров.

Определим в качестве иллюстрации передаточную функцию ЦВМ для рассмотренного в § 5.2 примера оптимального выделения полезного сигнала из его смеси с помехой. При этом примем, что непрерывная часть системы имеет передаточную функцию $W_0(p) = k_0/(1 + T_0p)$, соответствующую аperiodическому звену первого порядка. Дискретная передаточная функция непрерывной части в соответствии с (3.45)

$$W_{\text{н}}(z) = \frac{\delta}{\delta_1} \frac{z-1}{z} Z \left\{ \frac{k_0}{p(1+T_0p)} \right\} = \frac{\delta k_0}{\delta_1} \frac{1-d}{z-d},$$

где $d = \exp(-T_0/T)$.

В § 5.2 была получена желаемая частотная передаточная функция замкнутой системы, обеспечивающая оптимальное решение задачи фильтрации

$$H_{\text{ж}}^*(j\lambda) = \frac{1 + j\lambda T/2}{1 + j\lambda \tau}.$$

Для разомкнутой системы имеем

$$W_{\text{ж}}^*(j\lambda) = \frac{H^*(j\lambda)}{1 - H^*(j\lambda)} = \frac{1 + j\lambda T/2}{(\tau - T/2)j\lambda}.$$

Желаемая дискретная передаточная функция разомкнутой системы будет

$$W_{\text{ж}}(z) = W_{\text{ж}}^* \left(\frac{2}{T} \frac{z-1}{z+1} \right) = \frac{T}{\tau - T/2} \frac{1}{z-1}.$$

В соответствии с формулой (5.53) может быть получена передаточная функция ЦВМ

$$D(z) = \frac{W_{\text{ж}}(z)}{W_{\text{н}}(z)} = \frac{\delta_1 T}{\delta k_0 (\tau - T/2) (1-d)} \frac{z-d}{z-1} = A \frac{1-dz^{-1}}{1-z^{-1}}.$$

Если коэффициент передачи непрерывной части выбрать так, чтобы $A = 1$, или $k_0 = \delta_1 T / \delta (\tau - T/2) (1-d)$, то последнее выражение упростится:

$$D(z) = \frac{1-dz^{-1}}{1-z^{-1}}.$$

Отсюда может быть получена программа работы ЦВМ

$$x_0[n] = e_0[n] - de_0[n-1] + x_0[n-1],$$

где $e_0[n]$ и $x_0[n]$ – входная и выходная решетчатые функции ЦВМ (рис. 3.8), причем $e_0[n] = g_0[n] - y_0[n]$.

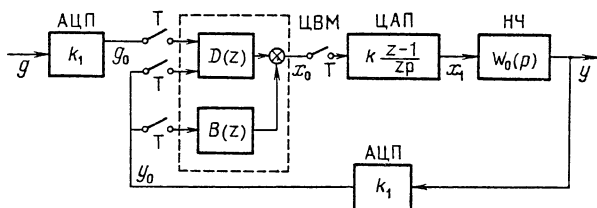
Можно получить частотную передаточную функцию ЦВМ

$$D^*(j\lambda) = D \left(\frac{1 + j\lambda T/2}{1 - j\lambda T/2} \right) = \frac{1-d}{T} \frac{1 + j\lambda \tau_0}{j\lambda},$$

где $\tau_0 = T(1+d)/[(1-d)2]$. Из полученного выражения видно, что дискретный фильтр, реализуемый на ЦВМ, осуществляет интегрирование и подъем верхних частот (дифференцирующее действие).

Цифровые корректирующие обратные связи. Дискретная коррекция может применяться и при использовании обратных связей в соответствии с рис. 5.12. Передаточная функция $B(z)$ такой обратной связи может быть найдена из известной желаемой передаточной функции разомкнутого канала. Если дискретная коррекция в прямой цепи отсутствует, т.е. $D(z) = 1$, то

$$W_{\text{ж}}(z) = \frac{W_{\text{н}}(z)}{1 + B(z)W_{\text{н}}(z)}. \quad (5.56)$$



Р и с. 5.12. Использование цифровой корректирующей обратной связи

Отсюда можно определить передаточную функцию цепи обратной связи

$$B(z) = \frac{W_{\text{н}}(z) - W_{\text{ж}}(z)}{W_{\text{ж}}(z)W_{\text{н}}(z)} = \frac{1}{W_{\text{ж}}(z)} - \frac{1}{W_{\text{н}}(z)}. \quad (5.57)$$

Аналогичным образом для частотной передаточной функции имеем

$$B^*(j\lambda) = \frac{1}{W_{\text{ж}}^*(j\lambda)} - \frac{1}{W_{\text{н}}^*(j\lambda)}. \quad (5.58)$$

Полученные значения передаточной функции по формулам (5.57) и (5.58) следует рассматривать как первое приближение искомым $B(z)$ или $B^*(j\lambda)$. Эти требования следующие.

1. Передаточной функции $B(z)$ должна соответствовать устойчивая программа вычислений на ЦВМ в том смысле, что полюсы этой передаточной функции должны находиться внутри круга единичного радиуса.

2. Степень числителя $B(z)$ не должна превышать степени знаменателя. Нарушение этого условия означает, что при вычислении необходимо будет знать будущие значения управляемой величины, поступающей на вход звена с передаточной функцией $B(z)$.

3. Желательно, чтобы степень числителя $B^*(j\lambda)$ была не больше степени знаменателя. Невыполнение этого условия может привести к возрастанию шумовой помехи на выходе звена с передаточной функцией $B^*(j\lambda)$ от квантования по уровню управляемой величины. Дисперсия этой помехи в канале ошибки ЦВМ может быть подсчитана по формуле

$$D_{\text{к}} = \frac{\delta_1^2}{12} \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{|B^*(j\lambda) [1 + W_{\text{н}}^*(j\lambda)]|^2 d\lambda}{|(1 + j\lambda T/2) \{1 + [1 + B^*(j\lambda)] W_{\text{н}}^*(j\lambda)\}|^2}, \quad (5.59)$$

где δ_1 — цена единицы младшего разряда входного преобразова-

теля. При инерционном объекте можно записать приближенно

$$D_{\kappa} \approx \frac{\delta_1^2}{12} \frac{T}{2\pi} \int_{-\infty}^{\infty} \frac{|B^*(j\lambda)|^2 d\lambda}{|1 + j\lambda T/2|^2}. \quad (5.60)$$

Из последнего выражения и вытекает сформулированное требование.

4. Для сохранения устойчивости по цепи корректирующей обратной связи необходимо, чтобы выполнялось требование нахождения внутри круга единичного радиуса корней характеристического уравнения $1 + B(z)W_n(z) = 0$.

Так как эти требования могут не выполняться в передаточной функции, то при определении окончательного вида $B(z)$ или $B^*(j\lambda)$ необходимо ввести коррективы.

При невыполнении первого требования (условия устойчивости программы вычисления) можно использовать отбрасывание множителей, соответствующих неустойчивым полюсам, или замену их близкими множителями, но с устойчивыми полюсами.

При невыполнении второго требования (условия использования в программе только текущих и предыдущих значений величин) возможно домножение передаточной функции $B(z)$ на $1/z$ в соответствующей степени.

При невыполнении третьего требования (условия ограничения полосы пропускания цепи обратной связи) возможно домножение $B^*(j\lambda)$ на множитель $1/(1 + j\lambda T/2)$ в соответствующей степени. Этому соответствует домножение $B(z)$ на множитель $(z + 1)/(2z)$ в той же степени.

Естественно, что эти изменения в полученной передаточной функции $B(z)$ или $B^*(j\lambda)$ приводят к изменению результирующей передаточной функции канала управления, полученной в результате синтеза. Поэтому из уточненного выражения передаточной функции обратной связи следует перейти к новым значениям $W_{ж}(z)$ или $W^*(j\lambda)$ и вновь просмотреть задачу формирования желаемых передаточных функций с учетом поправок, внесенных в цепь обратной связи.

При невыполнении принятых требований приходится возвращаться к вопросу уточнения вида $B(z)$ или принять новые требования и исходные данные к динамическим свойствам проектируемой системы (увеличить допустимый показатель колебательности, снизить имеющиеся в системе постоянные времени, увеличить допустимую ошибку и т.п.).

При решении такой задачи иногда удобнее пользоваться не формулами (5.56) и (5.57), а исходить из полученного желаемого выражения для передаточной функции канала управления, удовлетворяющей принятым требованиям. При выполнении принятых

требований уточненное значение $B(z)$ может быть принято как окончательное. По передаточной функции $B(z)$ или $B^*(j\lambda)$ может быть получена обычными способами программа работы ЦВМ по цепи обратной связи в виде разностного уравнения для величин $y_0[n]$ и $x_0[n]$.

Цифровыми методами может быть также решена задача построения системы комбинированного управления. Использование комбинированного управления позволяет существенно снизить требования к общему коэффициенту усиления основного канала управления, что значительно облегчает выбор последовательной цифровой коррекции или цифровой обратной связи. Более подробно см. работу [23].

ПРИЛОЖЕНИЕ 1

СПИСОК КОМАНД МИКРОЭВМ LSI-11

Мнемо-код	Машинный код	Функция команды	Коды условий			
			N	Z	V	C
1	2	3	4	5	6	7
<i>Команды пересылок</i>						
MOV	SD	<i>b1msmd</i>				
		переслать: $(src) \rightarrow (dst)$	+	+	0	-
<i>Арифметические команды</i>						
ADD	SD	<i>06msmd</i>				
		сложить: $(src) + (dst) \rightarrow (dst)$	+	+	+	+
SUB	SD	<i>16msmd</i>				
		вычесть: $(dst) - (src) \rightarrow (dst)$	+	+	+	+
CLR	D	<i>b050md</i>				
		очистить: $0 \rightarrow (dst)$	0	1	0	0
COM	D	<i>b051md</i>				
		получить обратный код: $(\bar{dst}) \rightarrow (dst)$	+	+	0	1
INC	D	<i>b052md</i>				
		увеличить на единицу: $(dst) + 1 \rightarrow (dst)$	+	+	+	-
DEC	D	<i>b053md</i>				
		уменьшить на единицу: $(dst) - 1 \rightarrow (dst)$	+	+	+	-
NEG	D	<i>b054md</i>				
		получить дополнительный код: $(\bar{dst}) + 1 \rightarrow (dst)$	+	+	+	+
ADC	D	<i>b055md</i>				
		прибавить перенос: $(dst) + c \rightarrow (dst)$	+	+	+	+
SBC	D	<i>b056md</i>				
		вычесть перенос: $(dst) - c \rightarrow (dst)$	+	+	+	+
TST	D	<i>b057md</i>				
		проверить байт: $(dst) \rightarrow (dst)$	+	+	0	0
MUL	S, <i>rd</i>	<i>070dms</i>				
		умножить: $(src) \times [rd] \rightarrow$	+	+	0	+

Приложение 1 (продолжение)

1	2	3	4	5	6	7
DIV <i>rd</i>	S, 071dms	→ [rd], [rd + 1], если <i>rd</i> четный; (src) × [rd] → [rd], если <i>rd</i> нечетный разделить: [rd], [rd + 1]/(src) → → [rd]; [rd + 1] – остаток; <i>rd</i> – всегда четный	+	+	+	+
<i>Логические команды</i>						
BIT	SD b3msmd	логическое умноже- ние: (src) ∧ (dst)	+	+	0	–
BIC	SD b4msmd	<u>по</u> битовый сброс: (src) ∧ (dst) → (dst)	+	+	0	–
BIS	SD b5msmd	логическое сложение: (src) ∨ (dst) → (dst)	+	+	0	–
XOR D	rs, 074smd	Исключающее ИЛИ: [rs] ⊕ (dst) → (dst)	+	+	–	0
CMP	SD b2msmd	сравнить: (src) – (dst)	+	+	+	+
<i>Команды переходов</i>						
BNE	C 001000 + XXX	перейти, если не равно нулю: [СК] + 2XXX → [СК]	–	–	–	–
BEQ	C 001400 + XXX	перейти, если равно нулю: [СК] + 2XXX → [СК]	–	–	–	–
BGE	C 002000 + XXX	перейти, если больше или равно нулю: [СК] + 2XXX → [СК]	–	–	–	–
BLT	C 002400 + XXX	перейти, если меньше нуля: [СК] + 2XXX → [СК]	–	–	–	–
BGT	C 003000 + XXX	перейти, если больше нуля: [СК] + 2XXX → [СК]	–	–	–	–
BLE	C 003400 + XXX	перейти, если меньше или равно нулю: [СК] + 2XXX → [СК]	–	–	–	–
BPL	C 100000 + XXX	перейти, если резуль- тат положителен: [СК] + 2XXX → [СК]	–	–	–	–

Приложение 1 (продолжение)

1	2	3	4	5	6	7	
BMI	C	100400 + XXX	перейти, если результат отрицателен: [CK] + 2XXX → [CK]	-	-	-	-
BNI	C	101000 + XXX	перейти, если больше: [CK] + 2XXX → [CK]	-	-	-	-
BLOS	C	101100 + XXX	перейти, если меньше или равно: [CK] + 2XXX → [CK]	-	-	-	-
BVC	C	102000 + XXX	перейти, если в бите переполнения ноль: [CK] + 2XXX → [CK]	-	-	-	-
BVS	C	102400 + XXX	перейти, если в бите переполнения единица: [CK] + 2XXX → [CK]	-	-	-	-
BCC	C	103000 + XXX	перейти, если в бите переноса ноль: [CK] + 2XXX → [CK]	-	-	-	-
BCS	C	103400 + XXX	перейти, если в бите переноса единица: [CK] + 2XXX → [CK]	-	-	-	-
BR	C	000400 + XXX	безусловный переход: [CK] + 2XXX → [CK]	-	-	-	-
JMP	D	0001md	безусловный переход по содержимому исполнительного адреса: (dst) → [CK]; m ≠ 0	-	-	-	-
SOB	rs, C	077smn	вычесть единицу и перейти: [rs] - 1 → [rs]; [rs] ≠ 0 ⇒ ⇒ [CK] - 2mn → [CK]; [rs] = 0 ⇒ [CK] → [CK]	-	-	-	-
<i>Специальные команды</i>							
JSR	rsd, D	004smd	перейти к подпрограмме: [rsd] → ↓ (УС); [CK] → [rsd]; (dst) → [CK]	-	-	-	-
RTS	rsd	00020d	выйти из подпрограммы: [rsd] → [CK]; (УС) ↑ → [rsd]	-	-	-	-
SWAB		0003md	поменять местами байты слова	+	+	0	0

Приложение 1 (продолжение)

1	2	3	4	5	6	7
HALT	000000	останов	-	-	-	-
WAIT	000001	ждать: процессор освобождает шину и ждет прерывания	-	-	-	-
RTI	000002	выйти из прерывания:	+	+	+	+
		(УС)† → [СК];				
		(УС)† → [РСП]				
RESET	000005	сброс периферийных устройств	-	-	-	-
NOP	000240	нет операции	-	-	-	-
MFPS D	1067md	прочсть слово состояния процессора:	+	+	0	-
MTPSS	1066ms	записать слово состояния процессора:	+	+	+	+
<i>Команды битовых операций</i>						
CLN	000250	очистить N: 0 → N	0	-	-	-
CLZ	000244	очистить Z: 0 → Z	-	0	-	-
CLV	000242	очистить V: 0 → V	-	-	0	-
CLC	000241	очистить C: 0 → C	-	-	-	0
CCC	000257	очистить N, Z, V, C: 0 → N, Z, V, C	0	0	0	0
SEN	000270	установить N: 1 → N	1	-	-	-
SEZ	000264	установить Z: 1 → Z	-	1	-	-
SEV	000262	установить V: 1 → V	-	-	1	-
SEC	000261	установить C: 1 → C	-	-	-	1
SCC	000277	установить N, Z, V, C: 1 → N, Z, V, C	1	1	1	1
SXT D	0067md	расширить знак: N = 0 ⇒ 0 → (dst) N = 1 ⇒ -1 → (dst)	-	+	-	-
<i>Команды сдвига</i>						
ROR D	b060md	выполнить циклический (логический) сдвиг вправо на 1 бит, включая бит C	+	+	+	+
ROL D	b061md	выполнить циклический (логический) сдвиг влево на 1 бит, включая разряд C	+	+	+	+

Приложение 1 (окончание)

1	2	3	4	5	6	7	
ASR	D	$b062m$	выполнить арифметический сдвиг вправо на 1 бит, включая разряд C, знаковый бит сохраняется	+	+	+	+
ASL	D	$b063m$	выполнить арифметический сдвиг влево на 1 бит, включая разряд C, знаковый бит сохраняется	+	+	+	+

Примечание. В Приложении 1 введены следующие обозначения: b – признак байтовой обработки (для слов длиной в 16 разрядов $b = 0$, для слов длиной в 8 разрядов $b = 1$); m – восьмеричный код метода адресации; SD, D – соответственно операнды SD и D – групп табл. 2.8; s, d – восьмеричные коды регистров общего назначения; rs, rd, rsd – имена регистров общего назначения; S, D – соответственно имена частей операндов, определяющих адрес источника и приемника; C – символическое имя адреса перехода; X, n – восьмеричное число (один разряд); СК – обозначение счетчика команд; УС – обозначение указателя стека; ↓, ↑ – соответственно символы операции записи информации в стек и чтения из стека; ССП – слово состояния процессора; РСР – регистр слова состояния процессора; N – бит знака результата операции; Z – бит нуля; V – бит арифметического переполнения; \underline{C} – бит переноса из старшего разряда; → – символ операции пересылки; (·) – символ операции получения обратного кода; Λ – логическое И; V – логическое ИЛИ; ∇ – исключающее ИЛИ.

ПРИЛОЖЕНИЕ 2

ИЗОБРАЖЕНИЯ РЕШЕТЧАТЫХ ФУНКЦИЙ

Производящая непрерывная функция		Несмещенная решетчатая функция
Оригинал	Преобразование Лапласа	
1	2	3
$f(t) = \begin{cases} 1 & \text{при } t = 0 \\ 0 & \text{при } t \neq 0 \end{cases}$	—	$\delta_0[n]$
$1(t) - 1(t - T)$	$\frac{1 - e^{-pT}}{p}$	$\nabla 1[n] = \Delta 1[n-1]$
$1(t)$	$\frac{1}{p}$	$1[n]$
t	$\frac{1}{p^2}$	nT
$\frac{t^2}{2!}$	$\frac{1}{p^3}$	$\frac{(nT)^2}{2!}$
$\frac{t^3}{3!}$	$\frac{1}{p^4}$	$\frac{(nT)^3}{3!}$
$\frac{t^k}{k!}$	$\frac{1}{p^{k+1}}$	$\frac{(nT)^k}{k!}$
$e^{-\alpha t}$	$\frac{1}{p + \alpha}$	$e^{-\alpha nT} = d^n$
$1 - e^{-\alpha t}$	$\frac{\alpha}{p(p + \alpha)}$	$1 - e^{-\alpha nT}$
$te^{-\alpha t}$	$\frac{1}{(p + \alpha)^2}$	$nTe^{-\alpha nT}$

z-преобразование	
Простое	Смещенное
4	5

1

0

1

1

$$\frac{z}{z-1}$$

$$\frac{z}{z-1}$$

$$\frac{Tz}{(z-1)^2}$$

$$Tz \left[\frac{\epsilon}{z-1} + \frac{1}{(z-1)^2} \right]$$

$$\frac{T^2 z (z+1)}{2!(z-1)^3}$$

$$\frac{T^2 z}{2!} \left[\frac{\epsilon^2}{z-1} + \frac{2\epsilon}{(z-1)^2} + \frac{z+1}{(z-1)^3} \right]$$

$$\frac{T^3 z (z^2 + 4z + 1)}{3!(z-1)^4}$$

$$\frac{T^3 z}{3!} \left[\frac{\epsilon^3}{z-1} + \frac{3\epsilon^2}{(z-1)^2} + \frac{3\epsilon(z+1)}{(z-1)^3} + \frac{z^2 + 4z + 1}{(z-1)^4} \right]$$

$$\frac{T^k z R_k(z)}{k!(z-1)^{k+1}}$$

$$\frac{T^k z}{k!} \sum_{\nu=0}^k C_k^\nu \frac{R_\nu(z)}{(z-1)^{\nu+1}} \cdot \epsilon^{k-\nu}$$

$$\frac{z}{z-d}, \quad d = e^{-\alpha T}$$

$$\frac{zd^\epsilon}{z-d}, \quad d = e^{-\alpha T}$$

$$\frac{(1-d)z}{(z-1)(z-d)}$$

$$\frac{z}{z-1} - \frac{zd^\epsilon}{z-d}$$

$$\frac{zd}{(z-d)^2}$$

$$\frac{zd^\epsilon \epsilon}{z-d} + \frac{zd^{\epsilon+1}}{(z-d)^2}$$

Приложение 2 (окончание)

1	2	3
$\frac{t^2}{2!} e^{-\alpha t}$	$\frac{1}{(p + \alpha)^3}$	$\frac{(nT)^2}{2!} e^{-\alpha nT}$
$\frac{t^k}{k!} e^{-\alpha t}$	$\frac{1}{(p + \alpha)^{k+1}}$	$\frac{(nT)^k}{k!} e^{-\alpha nT}$
$\sin \pi \frac{t}{T}$	$\frac{\pi T^{-1}}{p^2 + \pi^2 T^{-2}}$	$\sin \pi n = 0$
$\cos \pi \frac{t}{T}$	$\frac{p}{p^2 + \pi^2 / T^2}$	$\cos \pi n = (-1)^n$
$\sin \frac{\pi}{2} \frac{t}{T}$	$\frac{0,5 \pi / T}{p^2 + 0,25 \pi^2 / T^2}$	$\sin \frac{\pi}{2} n$
$\cos \frac{\pi}{2} \frac{t}{T}$	$\frac{p}{p^2 + 0,25 \pi^2 / T^2}$	$\cos \frac{\pi}{2} n$
$\sin \beta t$	$\frac{\beta}{p^2 + \beta^2}$	$\sin \beta n T$
$\cos \beta t$	$\frac{p}{p^2 + \beta^2}$	$\cos \beta n T$
$e^{-\alpha t} \sin \beta t$	$\frac{\beta}{(p + \alpha)^2 + \beta^2}$	$e^{-\alpha n T} \sin \beta n T$
$e^{-\alpha t} \cos \beta t$	$\frac{p + \alpha}{(p + \alpha)^2 + \beta^2}$	$e^{-\alpha n T} \cos \beta n T$

$$\frac{z(z+d)d^2}{2!(z-d)^3}$$

$$\frac{zR_k(d^{-1}z)d^k}{k!(z-d)^{k+1}}$$

0

$$\frac{z}{z+1}$$

$$\frac{z}{z^2+1}$$

$$\frac{z^2}{z^2+1}$$

$$\frac{z \sin \beta T}{z^2 - 2z \cos \beta T + 1}$$

$$\frac{z^2 - z \cos \beta T}{z^2 - 2z \cos \beta T + 1}$$

$$\frac{zd \sin \beta T}{z^2 - 2zd \cos \beta T + d^2}$$

$$\frac{z^2 - zd \cos \beta T}{z^2 - 2zd \cos \beta T + d^2}$$

$$\frac{zd^\epsilon \epsilon^2}{2!(z-d)} + \frac{zd^{\epsilon+1}}{2!(z-d)^2} +$$

$$+ \frac{z(z+d)d^{\epsilon+2}}{(z-d)^3}$$

$$\frac{z}{k!} \sum_{\nu=0}^k C_k^\nu \frac{R_\nu(d^{-1}z)d^{\epsilon+\nu} \epsilon^{k-\nu}}{(z-d)^{\nu+1}}$$

$$\frac{z \sin \pi \epsilon}{z+1}$$

$$\frac{z \cos \pi \epsilon}{z+1}$$

$$\frac{z^2 \sin \pi/2 \epsilon + z \cos \pi/2 \epsilon}{z^2 + 1}$$

$$\frac{z^2 \cos \pi/2 \epsilon - z \sin \pi/2 \epsilon}{z^2 + 1}$$

$$\frac{z^2 \sin \epsilon \beta T + z \sin(1-\epsilon)\beta T}{z^2 - 2z \cos \beta T + 1}$$

$$\frac{z^2 \cos \epsilon \beta T - z \cos(1-\epsilon)\beta T}{z^2 - 2z \cos \beta T + 1}$$

$$zd^\epsilon \frac{z \sin \epsilon \beta T + d \sin(1-\epsilon)\beta T}{z^2 - 2zd \cos \beta T + d^2}$$

$$zd^\epsilon \frac{z \cos \epsilon \beta T - d \cos(1-\epsilon)\beta T}{z^2 - 2zd \cos \beta T + d^2}$$

ПРИЛОЖЕНИЕ 3

ТАБЛИЦА ИНТЕГРАЛОВ

$$I_n = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{G(j\omega)d\omega}{A(j\omega)A(-j\omega)},$$

где

$$A(j\omega) = a_0(j\omega)^n + a_1(j\omega)^{n-1} + \dots + a_n,$$

$$G(j\omega) = b_0(j\omega)^{2n-2} + b_1(j\omega)^{2n-4} + \dots + b_{n-1}$$

и все корни $A(j\omega)$ расположены в верхней полуплоскости;

$$I_1 = \frac{b_0}{2a_0a_1},$$

$$I_2 = \frac{-b_0 + \frac{a_0b_1}{a_2}}{2a_0a_1},$$

$$I_3 = \frac{-a_2b_0 + a_0b_1 - \frac{a_0a_1b_2}{a_3}}{2a_0(a_0a_3 - a_1a_2)},$$

$$I_4 = \frac{b_0(-a_1a_4 + a_2a_3) - a_0a_3b_1 + a_0a_1b_2 + \frac{a_0b_3}{a_4}(a_0a_3 - a_1a_2)}{2a_0(a_0a_3^2 + a_1^2a_4 - a_1a_2a_3)},$$

$$I_5 = \frac{M_5}{2a_0\Delta_5},$$

$$M_5 = b_0(-a_0a_4a_5 + a_1a_4^2 + a_2^2a_5 - a_2a_3a_4) + a_0b_1(-a_2a_5 + a_3a_4) +$$

$$+ a_0b_2(a_0a_5 - a_1a_4) + a_0b_3(-a_0a_3 + a_1a_2) +$$

$$+ \frac{a_0b_4}{a_5}(-a_0a_1a_5 + a_0a_3^2 + a_1^2a_4 - a_1a_2a_3),$$

$$\Delta_5 = a_0^2a_5^2 - 2a_0a_1a_4a_5 - a_0a_2a_3a_5 + a_0a_3^2a_4 + a_1^2a_4^2 + a_1a_2^2a_5 - a_1a_2a_3a_4,$$

$$I_6 = \frac{M_6}{2a_0 \Delta_6},$$

$$\begin{aligned} M_6 = & b_0(-a_0 a_3 a_5 a_6 + a_0 a_4 a_5^2 - a_1^2 a_6^2 + 2a_1 a_2 a_5 a_6 + a_1 a_3 a_4 a_6 - \\ & - a_1 a_4^2 a_5 - a_2^2 a_5^2 - a_2 a_3^2 a_6 + a_2 a_3 a_4 a_5) + a_0 b_1(-a_1 a_5 a_6 + a_2 a_5^2 + a_2^2 a_6 - a_3 a_4 a_5) + \\ & + a_0 b_2(-a_0 a_5^2 - a_1 a_3 a_6 + a_1 a_4 a_5) + a_0 b_3(a_0 a_3 a_5 + a_1^2 a_6 - a_1 a_2 a_5) + \\ & + a_0 b_4(a_0 a_1 a_5 - a_0 a_3^2 - a_2^2 a_4 + a_1 a_2 a_3) + \frac{a_0 b_5}{a_6}(a_0^2 a_5^2 + a_0 a_1 a_3 a_6 - \\ & - 2a_0 a_1 a_4 a_5 - a_0 a_2 a_3 a_5 + a_0 a_3^2 a_4 - a_1^2 a_2 a_6 + a_1^2 a_4^2 + a_1 a_2^2 a_5 - a_1 a_2 a_3 a_4), \\ \Delta_6 = & a_0^2 a_5^3 + 3a_0 a_1 a_3 a_5 a_6 - 2a_0 a_1 a_4 a_5^2 - a_0 a_2 a_3 a_5^2 - a_0 a_2^2 a_6 + a_0 a_3^2 a_4 a_5 + \\ & + a_1^3 a_6^2 - 2a_1^2 a_2 a_5 a_6 - a_1^2 a_3 a_4 a_6 + a_1^2 a_4^2 a_5 + a_1 a_2^2 a_5^2 + a_1 a_2 a_3^2 a_6 - a_1 a_2 a_3 a_4 a_5, \end{aligned}$$

$$I_7 = \frac{M_7}{2a_0 \Delta_7},$$

где

$$\begin{aligned} M_7 = & b_0 m_0 + a_0 b_1 m_1 + a_0 b_2 m_2 + \dots + a_0 b_6 m_6, \\ m_0 = & a_0^2 a_6 a_7^2 - 2a_0 a_1 a_6^2 a_7 - 2a_0 a_2 a_4 a_7^2 + a_0 a_2 a_5 a_6 a_7 + a_0 a_3 a_5 a_6^2 + a_0 a_4^2 a_5 a_7 - \\ & - a_0 a_4 a_5^2 a_6 + a_1^2 a_6^3 + 3a_1 a_2 a_4 a_6 a_7 - \\ & - 2a_1 a_2 a_5 a_6^2 - a_1 a_3 a_4 a_6^2 - a_1 a_4^2 a_7 + a_1 a_4^2 a_5 a_6 + \\ & + a_2^2 a_7^2 - 2a_2^2 a_3 a_6 a_7 - a_2^2 a_4 a_5 a_7 + a_2^2 a_5^2 a_6 + a_2 a_3 a_4^2 a_7 - a_2 a_3 a_4 a_5 a_6 - a_2 a_3^2 a_6^2, \\ m_1 = & a_0 a_4 a_7^2 - a_0 a_5 a_6 a_7 - a_1 a_4 a_6 a_7 + a_1 a_5 a_6^2 - a_2^2 a_7^2 + 2a_2 a_3 a_6 a_7 + \\ & + a_2 a_4 a_5 a_7 - a_2 a_5^2 a_6 - a_3^2 a_6^2 - a_3 a_4^2 a_7 + a_3 a_4 a_5 a_6, \\ m_2 = & a_0 a_2 a_7^2 - a_0 a_3 a_6 a_7 - a_0 a_4 a_5 a_7 + a_0 a_5^2 a_6 - a_1 a_2 a_6 a_7 + \\ & + a_1 a_3 a_6^2 + a_1 a_4^2 a_7 - a_1 a_4 a_5 a_6, \\ m_3 = & -a_2^2 a_7^2 + 2a_0 a_1 a_6 a_7 + a_0 a_3 a_4 a_7 - a_0 a_3 a_5 a_6 - a_1^2 a_6^2 - a_1 a_2 a_4 a_7 + a_1 a_2 a_5 a_6, \\ m_4 = & a_0^2 a_5 a_7 - a_0 a_1 a_4 a_7 - a_0 a_1 a_5 a_6 - a_0 a_2 a_3 a_7 + \\ & + a_0 a_3^2 a_6 + a_1^2 a_4 a_6 + a_1 a_2^2 a_7 - a_1 a_2 a_3 a_6, \\ m_5 = & a_0^2 a_3 a_7 - a_0^2 a_5^2 - a_0 a_1 a_2 a_7 - a_0 a_1 a_3 a_6 + 2a_0 a_1 a_4 a_5 + a_0 a_2 a_3 a_5 - \\ & - a_0 a_3^2 a_4 + a_1^2 a_2 a_6 - a_1^2 a_4^2 - a_1 a_2^2 a_5 + a_1 a_2 a_3 a_4, \\ m_6 = & \frac{1}{a_7}(a_2^2 a_1 a_7^2 - 2a_2^2 a_3 a_5 a_7 + a_0^2 a_5^3 - 2a_0 a_1^2 a_6 a_7 + a_0 a_1 a_2 a_5 a_7 + 3a_0 a_1 a_3 a_5 a_6 - \\ & - 2a_0 a_1 a_4 a_5^2 + a_0 a_2 a_3^2 a_7 - a_0 a_2 a_3 a_5^2 - a_0 a_2^2 a_6 + a_0 a_3^2 a_4 a_5 + a_1^3 a_6^2 + a_1^2 a_2 a_4 a_7 - \\ & - 2a_1^2 a_2 a_5 a_6 - a_1^2 a_3 a_4 a_6 + a_1^2 a_4^2 a_5 - a_1 a_2 a_3 a_7 + a_1 a_2^2 a_5^2 + a_1 a_2 a_3^2 a_6 - a_1 a_2 a_3 a_4 a_5), \\ \Delta_7 = & -a_0^3 a_7^3 + 3a_0 a_1^2 a_6 a_7^2 + a_0 a_2 a_5 a_7^2 + 2a_0^2 a_3 a_4 a_7^2 - 3a_0^2 a_3 a_5 a_6 a_7 - a_0^2 a_4 a_5^2 a_7 + \\ & + a_0^2 a_5^3 a_6 - 3a_0 a_1^2 a_6^2 a_7 - 3a_0 a_1 a_2 a_4 a_7^2 + a_0 a_1 a_2 a_5 a_6 a_7 + 3a_0 a_1 a_3 a_5 a_6^2 - \\ & - a_0 a_1 a_3 a_4 a_6 a_7 + 2a_0 a_1 a_4^2 a_5 a_7 - 2a_0 a_1 a_4 a_5^2 a_6 - a_0 a_2^2 a_3 a_7^2 + 2a_0 a_2 a_3^2 a_6 a_7 + \\ & + a_0 a_2 a_3 a_4 a_5 a_7 - a_0 a_2 a_3 a_5^2 a_6 - a_0 a_3^3 a_6^2 - a_0 a_3^2 a_4^2 a_7 + a_0 a_3^2 a_4 a_5 a_6 + \\ & + a_1^3 a_6^3 + 3a_1^2 a_2 a_4 a_6 a_7 - 2a_1^2 a_2 a_5 a_6^2 - \\ & - a_1^2 a_3 a_4 a_6^2 - a_1^2 a_4^2 a_7 + a_1^2 a_4^2 a_5 a_6 + a_1 a_3^2 a_7^2 - 2a_1 a_2^2 a_3 a_6 a_7 - a_1 a_2^2 a_4 a_5 a_7 + \\ & + a_1 a_2^2 a_5^2 a_6 + a_1 a_2 a_3^2 a_6^2 + a_1 a_2 a_3 a_4^2 a_7 - a_1 a_2 a_3 a_4 a_5 a_6. \end{aligned}$$

СПИСОК ЛИТЕРАТУРЫ

1. *Аламада, Манн.* Применение микроЭВМ для формирования последовательностей высоковольтных импульсов и контроля температуры при исследованиях эффекта Керра//Приборы для научных исследований. — 1978. — № 11. — С. 17 — 22.
2. *Англо-русско-немецко-французский толковый словарь по вычислительной технике и обработке данных/Под ред. А.А. Дородицына.* — М.: Русский язык, 1981.
3. *Аналоговые и цифровые интегральные схемы/Под ред. С.В. Якубовского.* — М.: Советское радио, 1979.
4. *Аппаратное обеспечение микропроцессоров // Э.И. Приборы и элементы автоматики и вычислительной техники.* — 1977. — № 46. — С. 31—42.
5. *Арсенин В.П., Дервянко В.П.* Мультивычислительная система на базе микроЭВМ "Электроника-60"/"Электронная промышленность. — 1983. — № 9. — С. 62 — 66.
6. *Архитектура многопроцессорных вычислительных систем/Под ред. В.И. Тимохина.* — Л.: Изд-во ЛГУ, 1981.
7. *Балашов Е.П., Пузанков Д.В.* Микропроцессоры и микропроцессорные системы. — М.: Радио и связь, 1981
8. *Барабаш Ю.Л., Варский Б.В., Зиновьев В.Т.* и др. Вопросы статистической теории распознавания. — М.: Советское радио, 1967.
9. *Баранов С.Н., Кириллин В.А., Клубович А.А.* и др. Кросс-система К580/ЛГУ//Управляющие системы и машины. — 1983. — № 3. — С.19—21.
10. *Барашенков В.В., Бахарева Н.Ц., Гильгаева А.Ф.* и др. Система программирования с интерпретатором для цифровых приборов с встроенными микропроцессорами (микроЭВМ)//Управляющие системы и машины. — 1982. — № 2. — С. 76 — 78.
11. *Баумс А.К., Гутовцев А.Л., Зазнова Н.Е.* Микропроцессорные средства. — Рига: Зинатне, 1978.
12. *Баумс А.К., Зазнова Н.Е.* Применение микропроцессоров в системах автоматизации научных исследований//Изв. АН Латв. ССР. — 1978. — № 7. — С. 60 — 76.
13. *Бахтияров Г.Д., Тищенко А.Ю.* Реализация устройств цифровой обработки сигналов на основе алгоритмов БПФ//Зарубежная радиоэлектроника. — 1975. — № 9. — С. 71 — 97.
14. *Бащин О.И., Буевич В.В., Капителян В.Е.* и др. Микропроцессоры в энергетике. — Л.: Наука, 1982.
15. *Бедрековский М.А., Волга В.В., Кручинкин Н.С.* Микропроцессоры. — М.: Радио и связь, 1981.
16. *Бедрековский М.А., Кручинкин Н.С., Поделян В.А.* Микропроцессоры. — М.: Радио и связь, 1981.
17. *Беляев В.Н., Булгаков С.С., Глебов С.С.* и др. Микропроцессорный комплект БИС серии К1804//Электронная промышленность. — 1983. — № 9. — С.3 — 7.

18. Березенко А.И. МикроЭВМ на микропроцессорном наборе серии К589//Электронная промышленность. – 1978. – Вып. 6 (66). – С. 49–54.
19. Березенко А.И., Казанцев П.Н., Калинин С.Е. и др. Архитектурно-структурные особенности микропроцессорного комплекта БИС серии КР1802//Электронная промышленность. – 1983. – № 4. – С.19–23.
20. Березенко А.И., Корягин А.Н., Назарьян А.Р. Микропроцессорные комплекты повышенного быстродействия. – М.: Радио и связь, 1982.
21. Берисфорд Р. Параллельные АЦП и быстродействующие ЦАП для цифровой обработки сигналов//Электроника. – 1982. – Т. 55, № 9.
22. Бесекерский В.А. Динамический синтез систем автоматического регулирования. – М.: Наука, 1970.
23. Бесекерский В.А. Цифровые автоматические системы. – М.: Наука, 1976.
24. Бесекерский В.А., Изранцев В.В., Мишура О.В. Организация микропроцессорной вычислительной системы для численного интегрирования линейных дифференциальных уравнений//Радиотехника. – 1983. – № 6. – С. 60 – 61.
25. Бесекерский В.А., Небылов А.В. Робастные системы автоматического управления. – М.: Наука, 1983.
26. Бесекерский В.А., Попов Е.П. Теория систем автоматического регулирования. – М.: Наука, 1975.
27. Бойков В.И., Нельга А.Т., Тихобаев А.П. Устройство ввода технологической информации в микрокалькулятор "Электроника БЗ-21"//Приборы и системы управления. – 1983. – № 11. – С. 33 – 34.
28. Бокс Дж., Дженкинс Г. Анализ временных рядов. – М.: Мир, 1974. – Вып. 1.
29. Бондаренко П.А., Коштаев В.В., Кулиджанов Б.К. Устройство связи микроЭВМ с периферийными устройствами//Приборы и системы управления. – 1982. – № 3. – С. 31 – 32.
30. Борисенко В.Д. МикроЭВМ "Электроника-60"//Электронная промышленность. – 1978. – № 10. – С. 20 – 21.
31. Борисенко В.Д., Лопатин А.С., Плотников В.В. и др. Возможности построения систем на базе микроЭВМ "Электроника-60"//Электронная промышленность. – 1979. – № 11 – 12. С. 17 – 19.
32. Борисов В.С. Микропроцессорные комплекты интегральных схем. – М.: Радио и связь, 1982.
33. Борисов В.С., Васенков А.А., Малашев Б.М. и др. Микропроцессорные комплекты интегральных схем: Состав и структура. Справочник. – М.: Радио и связь, 1982.
34. Брусенцов Н.П. Миникомпьютеры. – М.: Наука, 1979.
35. Бузунов Ю.А., Буренков И.Г. Об одном способе реализации табличного метода умножения чисел//Управляющие системы и машины. – 1983. – № 6. – С. 12 – 16.
36. Вайда Ф., Чакань А. МикроЭВМ. – М.: Энергия, 1980.
37. Вариант языка ПЛ/1 для микропроцессоров//Электроника, – 1980. – № 10. – С. 6 – 7.
38. Васенков А.А. Развитие микропроцессоров и микроЭВМ семейства "Электроника НЦ" на основе комплексных целевых программ//Электронная промышленность. – 1979. – № 11 – 12. – С. 13 – 19.
39. Васенков А.А., Воробьев Н.М., Дихунян В.М. и др. Микропроцессорные БИС и микроЭВМ: Построение и применение. – М.: Советское радио, 1980.
40. Васенков А.А., Коночкин Э.И., Малашев Б.М. Терминология в технике микропроцессорных интегральных схем и микроЭВМ//Микроэлект-

- роника и полупроводниковые приборы. — М.: Советское радио, 1979. — Вып. 4. — С. 17 — 28.
41. *Вейцман К.* Распределенные системы мини- и микроЭВМ. — М.: Финансы и статистика, 1983.
 42. *Вешняков В.И., Кардашук М.О., Коваль В.Ф.* Комплексирование цифровых БИС для БПФ-процессоров//Управляющие системы и машины. — 1983. — № 5. — С. 12 — 17.
 43. Внедрение СБИС в область специализированных приложений//Электроника. — 1982. — № 3. — С. 34.
 44. *Воробьев Н.М., Васенков А.А., Поликанов М.Ф.* и др. Микропроцессорные наборы БИС и серия ЭВМ "Электроника ИЦ"/"Электронная промышленность. — 1978. — № 5. — С. 9 — 14.
 45. *Вторушина О.П., Изранцев В.В.* Оценка точности идентификации при статистическом прогнозе процессов//Тезисы докладов VIII Всесоюзного симпозиума по проблеме избыточности в информационных системах. Ч.4. — Л.: ЛИАП, 1983. — С. 141 — 142.
 46. Высокопроизводительная цифровая система обработки сигналов//Электроника. — 1982. — Т. 55, № 22. — С. 18 — 19.
 47. *Галушкин А.И., Зотов Ю.Н., Шикунов Ю.Н.* Оперативная обработка экспериментальной информации. — М.: Энергия, 1972.
 48. *Гальперин М.П., Масленников Ю.А., Нестеренко С.Н.* и др. Архитектура малой локальной сети//Электронная промышленность. — 1983. — № 9. — С. 66 — 68.
 49. *Гамильтон П.* Система в помощь парализованным, управляемая свистом//Электроника. — 1979. — № 7. — С. 8 — 9.
 50. *Гельман М.М.* Схемотехника микроАЦП системного применения//Электронная промышленность. — 1980. — № 7. — С. 18 — 23.
 51. *Герасимов И.В., Петров Г.А., Скворцов С.В.* Кросс-система разработки информационного обеспечения//Приборы и системы управления. — 1982. — № 7. — С. 10 — 13.
 52. *Гермес С.* Микропроцессоры — база для управляющих и вычислительных систем в автоматической измерительной технике//Зарубежная радиоэлектроника. — 1980. — № 3. — С. 86 — 90.
 53. *Гибсон Г., Лю Ю-Ч.* Аппаратные и программные средства микроЭВМ/Пер. с англ. — М.: Финансы и статистика, 1983.
 54. *Гигис Э.И.* Преобразователи информации для электронных цифровых вычислительных устройств. — М.: Энергия, 1973.
 55. *Гладков А.М., Хохлов Ю.В.* Оборудование для отладки программ микропроцессора К580ИК80//Приборы и системы управления. — 1982. — № 10. — С. 31 — 32.
 56. *Глазер С.Ф., Маликов А.Г.* Средства сокращения времени разработки микропроцессорных систем//Электронная промышленность. — 1983. — № 3 — С. 49—53.
 57. *Глазов Г.Я., Дризовский Л.М., Кореннов Е.Ф.* Опыт разработки и применения микропроцессоров//Приборы и системы управления. — 1977. — № 9. — С. 15 — 18.
 58. *Глазов Г.Я., Митяева Е.Н.* Программное обеспечение микроЭВМ//Обзорная информация/ЦНИИТЭИ приборостроения. — М., 1980. — Вып. 5.
 59. *Глушко О.В., Осинский Л.М.* Микропроцессорные средства для построения систем цифровой фильтрации сигналов//Управляющие системы и машины. — 1982. — № 1. — С. 73 — 76.
 60. *Голец Н.Т., Захаров В.П., Польский Ю.М.* и др. Автономные управляющие системы на основе микроконтроллеров//Электронная промышленность. — 1983. — № 3. — С. 59.

61. Горбунов В.Л., Николаев В.Т., Шахнов В.А. и др. Микропроцессорная техника в учебном процессе в высших учебных заведениях. — М.: НИИВШ, 1983.
62. Горн Л.С., Хазанов Б.И., Хазанов Д.Б. Микропроцессоры в приборах для радиационных измерений. — М.: Атомиздат, 1979.
63. Гош Дж. Микропроцессор со средствами для многозадачной работы в реальном времени//Электроника. — 1982. — № 22. — С. 15 — 17.
64. Грендбуа Г., Фримэн У. Многоканальные информационные системы с микропроцессором//Электроника. — 1981. — № 2. — С. 56—62.
65. Григорьев В.Л. Программное обеспечение микропроцессорных систем. — М.: Энергоатомиздат, 1983.
66. Гришин Ю.П., Казаринов Ю.М., Катиков В.М. Микропроцессоры в радиотехнических системах/Под ред. Ю.М. Казаринова. — М.: Радио и связь, 1982.
67. Громов Г.Н., Дроздов Р.В., Криворучко Ю.Т. Вычислитель на микропроцессорных БИС серии К588//Электронная промышленность. — 1983. — № 9. — С. 22 — 23.
68. ГОСТ 19. 402—78 ЕСПД. Описание программ//Единая система программной документации. — М.: Госком СССР по стандартам, 1982.
69. ГОСТ 19.003—80 ЕСПД. Схемы алгоритмов и программ. Обозначения условные графические//Единая система программной документации. — М.: Госком СССР по стандартам, 1982.
70. ГОСТ 19.002—80 ЕСПД. Схемы алгоритмов и программ. Правила выполнения//Единая система программной документации. — М.: Госком СССР по стандартам, 1982.
71. Грэндбойз Г., Фримең У. Упрощенный вариант сопряжения АЦП с процессором//Электроника. — 1983. — № 12. — С. 64 — 67.
72. Гусев В.Г. Методы исследования точности цифровых автоматических систем. — М.: Наука, 1973.
73. Демарк. Микропроцессорное распределенное управление технологическими процессами//Электроника. — 1976. — № 8. — С. 96 — 99.
74. Дешиц Е.Ф. Использование микропроцессорного комплекта для реализации каналов ввода-вывода//Электронная промышленность. — 1983. — № 9. — С. 29 — 30.
75. Джайн А.К.: Успехи в области математических моделей для обработки изображений//ТИИЭР. — 1981. — № 5. — С.9 — 39.
76. Джерри Л., Гудрич. Очень эффективная программа умножения и деления для микропроцессора 8080//Электроника. —1982.—№4.—С.74—75.
77. Джонсон К. Применение в микросистемах методов управления памятью, апробированных на крупных ЭВМ//Электроника. — 1981. — № 16. — С.44 — 57.
78. Джонсон Р.К. Операционная система с полным набором средств для работы с 16-разрядными микропроцессорами//Электроника. — 1982. — № 6. — С. 34 — 45.
79. Джонсон Р.К. ИС в качестве устройства сопряжения для бортовых систем информационного обмена//Электроника. —1981.— № 25. — С. 4 — 6.
80. Джури Э., Цыпкин Я.З. Теория дискретных автоматических систем (обзор)// А и Т. — 1970. — № 6.
81. Добринский Э.Е., Левин Ю.Г., Пенькин В.Н. и др. Стенд для отладки программ и аппаратуры микропроцессорных устройств//Приборы и системы управления. — 1981. — № 10. — С. 32 — 34.
82. Драгунов В.Л., Изранцев В.В. К вопросу образования и извлечения сведений об объекте в задаче распознавания образов// Труды. — Л.: ЛИАП, 1974. — Вып. 88. — С. 180 — 182.

83. Драгунов В.Л., Изранцев В.В., Чурочкин Е.Ф. Эффективность лингвистического метода распознавания образов//Труды. – Л.: ЛИАП, 1974. – Вып. 92. – С. 129 – 131.
84. Дризовский Л.М., Киселева Э.В., Буторина Т.С. Состояние и перспективы развития САПР//Приборы и системы управления. – 1983. – № 11. – С. 15 – 17.
85. Дихунян В.Л., Машевич П.Р., Коваленко С.С. и др. Архитектура однокристального микропроцессорного элемента К1801ВЕ1//Микроэлектроника: Сб. статей. – М.: Советское радио, 1980. – Вып. 6.
86. Елагин В.М., Неизвестнов А.А. Электронный регулятор отопления "Электроника Р-1М"//Электронная промышленность. – 1982. – № 9. – С. 18.
87. Елисеев А.А., Изранцев В.В., Мишура О.В. Показатели эффективности классификации локационных сигналов адаптивными лингвистическими структурами//Тезисы докладов Всесоюзной конференции "Теория адаптивных систем и ее применение". – М.; Л.: ВИНТИ, 1983.
88. Елисеев А.А., Изранцев В.В., Мишура О.В. Оценка параметров классификации дискретных сигналов на основе статистического эксперимента//Электронное моделирование. – 1985. – № 3. – С. 86 – 88.
89. Жабин В.И., Корнейчук В.И., Тарасенко В.П., Щербина А.А. Быстродействующая мультиплексорная вычислительная система для решения систем обыкновенных дифференциальных уравнений//Управляющие машины и системы. – 1980. – № 6 – С. 38 – 42.
90. Зазу В. Быстродействующий К/МОП АЦП со структурой кремний-на-сапфире//Электроника. – 1982. – № 9. – С. 52 – 55.
91. Зазнова Н.Е. Моделирование микропроцессорных систем с помощью средств языка Фортран-IV//Цифровые устройства и микропроцессоры. – 1979. – № 3. – С. 59 – 69.
92. Зарнеску С. Сопряжение 10-разрядного АЦП с 16-разрядным микропроцессором//Электроника. – 1982. – № 8. – С. 75 – 77.
93. Захаров В.П. Программируемые однокристальные микроконтроллеры серии К145//Электронная промышленность. – 1983. – № 3. – С. 27 – 30.
94. Зварич В.К., Кобылинский А.В., Сабадаш Н.Г. Универсальный программируемый контроллер "Электроника К1-20"//Электронная промышленность. – 1983. – № 3. – С. 24 – 27.
95. Зубашевич В.Ф., Кобылинский А.В., Темченко В.А. и др. Микропроцессорный комплект серии К580. Семейство микроЭВМ "Электроника К1" //Электронная промышленность. – 1979. – № 11 – 12. – С. 19 – 22.
96. Зурахинский В.И. Операционная система реального времени для микроЭВМ//Управляющие системы и машины. – 1983 – № 3. – С. 52 – 54.
97. Иванов А.Д., Старых В.А. Проектирование микропроцессорных систем//Зарубежная электронная техника. – М.: ЦНИИ "Электроника", 1980. – № 11.
98. Иванов Ю.В., Колосовский Л.И., Чебыкин Л.Е. и др. Кросс-средства разработки и отладки программ для микроЭВМ // Электронная промышленность. – 1978. – № 5.
99. Иверси Р.У. 16-разрядный процессор для управления двигателями на будущих моделях концерна FORD // Электроника. – 1981. – № 20. – С. 12 – 13.
100. Изранцев В.В. Архитектура структурно-избыточных микропроцессорных систем управления // Тезисы докладов VIII Всесоюзного симпозиума по проблеме избыточности в информационных системах. Ч. 4. – Л.: ЛИАП, 1983. – С. 89 – 92.

101. *Израңцев В.В.* Линейная дискриминантная функция сигналов с неравными матрицами ковариаций // Комплексные радиоэлектронные системы управления: Межвузовский сборник. – Л.: ЛИАП, 1977. – Вып. 118. – С. 32 – 34.
102. *Израңцев В.В.* Минимизация описания сигналов с неравными матрицами ковариаций коррелированных параметров // Тезисы докладов VII Всесоюзного симпозиума по проблеме избыточности в информационных системах. – Л.: ВИР, 1977. – С. 159 – 161.
103. *Израңцев В.В., Евлашев Н.Д., Яговой В.П.* Математическое и программное обеспечение микропроцессорной системы измерения текущих угловых координат подвижного объекта // Проблемы метрологического обеспечения систем обработки измерительной информации. – М.: ВНИИФТРИ, 1980. – С. 87 – 88.
104. *Израңцев В.В., Рот М.* Особенности проектирования микропроцессорных адаптивных классификаторов // Приборы и устройства электронных систем управления: Межвузовский сборник. – Л.: ЛИАП, 1980. – Вып. 143. – С. 66 – 69.
105. *Иордаң Г.Г., Курносое Н.М., Козлов М.Г.* и др. Микропроцессорные контроллеры в системах автоматического регулирования // Приборы и системы управления. – 1981. – № 1. – С. 50 – 54.
106. *Израелин В.К.* Комплексование бортового радиоэлектронного оборудования на основе мультиплексной шины // Зарубежная радиоэлектроника. – 1983. – № 9. – С. 52 – 63.
107. *Каган Б.М., Сташин В.В.* Микропроцессоры в цифровых системах. – М.: Энергия, 1979.
108. *Каллахэн Д., Пэйтел К.Н., Стивенсон Д.* Реализация виртуальной памяти в микросистемах // Электроника. – 1981. – № 13. – С. 37 – 43.
109. *Каляев А.В.* Многопроцессорные вычислительные системы с программируемой архитектурой // Многопроцессорные вычислительные структуры. – Таганрог: ТРТИ, 1981. – Вып. 3. – С. 5 – 12.
110. *Каляев А.В., Мелихов А.Н., Иванов Г.И.* и др. Система автоматизации программной настройки микропроцессорных управляющих автоматов // Управляющие системы и машины. – 1983. – № 5. – С. 27 – 31.
111. *Каньел А.* Аналого-цифровой преобразователь с микропроцессором // Электроника. – 1983. – № 10. – С. 60 – 62.
112. *Каппелини В., Константи́нидис А., Эмилиани П.* Цифровые фильтры и их применение. – М.: Энергоатомиздат, 1983.
113. *Кармаркар В.* Схема программного управления тремя электродвигателями // Электроника. – 1980. – № 18. – С. 69 – 71.
114. *Картов В.Д., Горбунов В.Т.* Цифровое телевидение // Зарубежная электронная техника. – 1980. – № 8. – С. 31 – 53.
115. *Като К.* Интегральная схема сбора данных // Электроника. – 1983. – № 9. – С. 45 – 50.
116. *Кибиткин В.В.* Единая операционная система для ЭВМ "Электроника-60", СМ-3 и СМ-4 // Управляющие системы и машины. – 1983. – № 5. – С. 65 – 69.
117. *Климашадаска К.Ю., Монставичюс Т.А., Мрщанкявичус А.* – И.К. Быстродействующий аналого-цифровой преобразователь К1107ПВ1 // Электронная промышленность. – 1983. – № 6. – С. 10 – 11.
118. *Клингман Э.* Проектирование микропроцессорных систем / Пер. с англ. – М.: Мир, 1980.
119. *Кобылинский А.В., Сабадаш Н.Г., Тесленко А.К.* Программное обеспечение микроЭВМ на базе однокристалльного микропроцессора // Микроэлектроника и полупроводниковые приборы. – М.: Советское радио, 1979. – Вып. 4. – С. 82 – 84.

120. *Коев М.* Преимущество ЭСЛ ЦАП: низкий уровень помех, повышение точности при уменьшении стоимости // *Электроника*. – 1982. – № 9. – С. 56 – 60.
121. Кольцо микропроцессоров в центральном обрабатывающем устройстве // *Электроника*. – 1981. – № 17. – С. 18 – 19.
122. Коммерческие микропроцессоры в военных программах // *Электроника*. – 1978. – № 5. – С. 9 – 10.
123. *Коневцов В.А.* Программная система цифрового управления // *Приборы и системы управления*. – 1983. – № 9. – С. 5 – 6.
124. *Конипов А.И., Кузнецов С.Н.* Использование микропроцессоров для повышения точности ЦАП // *Вопросы радиоэлектроники. Серия ЭВТ*. – 1978. – № 2. – С. 98 – 104.
125. Контроллер для увеличения огневой мощи самолетовылета // *Электроника*. – 1980. – № 16. – С. 10.
126. Контроллер для периферийных устройств микропроцессорных систем // *Э.И. Приборы и элементы автоматики и вычислительной техники*. – 1981. – № 18. – С. 5 – 13.
127. *Котельников В.А.* О пропускной способности эфира и проволоки в радиосвязи. – М.: Изд-во МГУ, 1933.
128. *Кофрон Дж.* Технические средства микропроцессорных систем / Пер. с англ. – М.: Мир, 1983.
129. *Коча В.М., Ланнэ А.А.* Аппаратная реализация цифровых фильтров // *Зарубежная радиоэлектроника*. – 1979. – № 9. – С. 49 – 57.
130. *Кричевский Б.М., Мустафаева А.А., Якуба А.А.* Некоторые проблемы разработки конвейерного управления в микропрограммной ЭВМ // *Управляющие системы и машины*. – 1981. – № 3. – С. 20 – 24.
131. *Крук Е.К., Александриди Т.М., Дилигенский С.Н.* Цифровые регуляторы. – М.: Госэнергониздат, 1966.
132. *Кузнецов В.Я., Масленников Ю.А., Никитин Э.А.* и др. Развитие микроЭВМ семейства "Электроника С5" и систем на их основе // *Электронная промышленность*. – 1979. – № 11 – 12. – С. 9 – 12.
133. *Кузьмин И.В., Кедрус В.А.* Основы теории информации и кодирования. – Киев: Вища школа, 1977.
134. *Кузьмин С.З.* Основы теории цифровой обработки радиолокационной информации. – М.: Советское радио, 1974.
135. *Купе Б.* Сверхэффективные программы умножения для микропроцессоров 8080 и Z80 // *Электроника*. – 1983. – № 6. – С. 74 – 76.
136. *Ламмерт М., Олсен Р.К.* Сверхскоростной параллельный АЦП на основе однокрипной технологии // *Электроника*. – 1982. – № 9. – С. 48 – 52.
137. *Лебедева С.И.* Микропроцессорный комплект быстройдействующих БИС серии K1800//*Электронная промышленность*, – 1983. – № 9. – С. 7–11.
138. *Левин Б.Р.* Теоретические основы статистической радиотехники. – М.: Советское радио, 1976. – Кн. 3.
139. *Лихарев В.А.* Цифровые методы и устройства в радиолокации. – М.: Советское радио, 1973.
140. *Лукошкин А.П., Оводенко А.А.* Перспективы построения робастных чувствительных элементов систем управления // *Межвузовский сборник*. – Л.: ЛИАП, 1981. – Вып. 151. – С. 6 – 11.
141. *Лэнтон С.* Гибридный преобразователь сельсин – код с большими интегральными схемами // *Электроника*. – 1981. – № 13. – С. 43 – 48.
142. *Макс Ж.* Методы и техника обработки сигналов при физических измерениях. Т. 1 / Пер. с англ. – М.: Мир, 1983.
143. *Малашевич Б.М.* Ассоциативные параллельные процессоры и микропроцессоры // *Микроэлектроника и полупроводниковые приборы*. – М.: Советское радио, 1979. – Вып. 4. – С. 70 – 82.

144. Мамзелев И.А., Русаков М.Ю., Часовников Е.Д. и др. Отказоустойчивые вычислительные системы // Зарубежная радиоэлектроника. – 1983. – № 11. – С. 3 – 28.
145. Мануэль Т. Усовершенствованные параллельные архитектуры как способ ускорения вычислений // Электроника. – 1982. – № 12. – С. 25 – 27.
146. Маркел Дж., Грей А.Х. Линейное предсказание речи / Пер. с англ. – М.: Связь, 1980.
147. Матесон, Хиггинс. Использование вспомогательного микропроцессора для проведения гармонического анализа в реальном времени при помощи миниЭВМ // Приборы для научных исследований. – 1978. – № 12. – С. 92 – 96.
148. Матричный процессор, содержащий 96 × 96 ячеек, для обработки изображений в реальном времени // Электроника. – 1980. – № 10. – С. 14 – 15.
149. Мейсон Б. Система управления на базе нескольких микропроцессоров модели 8048 // Электроника. – 1979. – № 25. – С. 6 – 7.
150. Меркулов О.Н., Петров И.М., Сергутин В.В. Интерфейс быстродействующего АЦП в системе связи с объектом микроЭВМ "Электроника-60" // Приборы и системы управления. – 1983. – № 3. – С. 29 – 30.
151. Микрокомпьютерные медицинские системы: Проектирование и применение / Под ред. У. Томпкинса, Дж. Уэбстера. – М.: Мир, 1983.
152. Микропроцессор военного назначения // Электроника. – 1982. – № 2. – С. 7.
153. Микропроцессор Z80 в системах контроля // Электроника. – 1981. – № 6. – С. 24 – 25.
154. Микропроцессор в усовершенствованной системе наведения тактических ракет // Электроника. – 1979. – № 19. – С. 5 – 6.
155. Микропроцессорное управляющее устройство // Э.И. Приборы и элементы автоматики и вычислительной техники. – 1979. – № 1. – С. 19 – 20.
156. Микропроцессоры // Сер. 3. Микроэлектроника. – М.: ЦНИИ "Электроника", 1982. – Вып. 1 (328).
157. МикроЭВМ / Под ред. А. Дирксена. – М.: Энергоиздат, 1982.
158. МикроЭВМ в системах управления // Э.И. Приборы и элементы автоматики и вычислительной техники. – 1980. – № 16. – С. 16 – 19.
159. Мультипроцессорные системы: Обзор практической реализации // ТИИЭР. – 1978. – № 2. – С. 135 – 150.
160. Мэрион Л. Процессор руки, распознающий различные сигналы организма // Электроника. – 1979. – № 8. – С. 4 – 5.
161. Нагорный Л.Я. Методы распараллеливания систем уравнений большой размерности для решения их на многопроцессорных структурах // Электронное моделирование. – 1980. – № 1. – С. 28 – 32.
162. Науман Г., Майлинг В., Шербина А. Стандартные интерфейсы для измерительной техники. – М.: Мир, 1982.
163. Неуен Ч.Д. Взаимодействие процессора с периферийными устройствами, использующими прямой доступ к памяти // Электроника. – 1982. – № 11. – С. 77 – 80.
164. Небылов А.В. Синтез аналого-цифровых радиотехнических следящих систем. – Л.: ЛЭТИ; ЛИАП, 1977.
165. Николаев Ю.А., Петухов В.П., Феклисов Г.И., Чемоданов Б.К. Динамика цифровых следящих систем. – М.: Энергия, 1970.
166. Образцова В.А., Топоров Д.И. Резидентная система автоматизации программирования для микроЭВМ "Электроника С5" // Электронная промышленность. – 1982. – № 2. – С. 60.
167. Основы проектирования следящих систем / Под ред. Н.А. Лакоты. – М.: Машиностроение, 1978.

168. *Отнес Р., Энкосон Л.* Прикладной анализ временных рядов/Пер. с англ. — М.: Мир, 1982.
169. *Пендер, Уингл.* Применение микропроцессора для исследования низкочастотных электрических шумов//Приборы для научных исследований. — 1978. — № 12. — С. 96 — 100.
170. *Петушин А.А., Похлебкин В.Е., Трояновский В.М.* и др. Система оперативно-диспетчерского управления "Электроника НЦ-26"//Электронная промышленность. — 1983. — № 9. — С. 71 — 72.
171. Повышение возможностей микропроцессоров с помощью схем поддержки//Э.И. Вычислительная техника. — 1981. — № 37. — С. 18 — 22.
172. *Погорелый С.Д., Решетников А.М., Сепалов В.И.* и др. Системное программное обеспечение модульного набора средств микропроцессорной техники//Средства связи. Сб. статей. — М.: Советское радио, 1981. — Вып. 1. — С. 16 — 20.
173. *Попов Е.П.* Прикладная теория процессов управления в нелинейных системах. — М.: Наука, 1973.
174. *Прангишвили И.В., Виленкин С.Я., Медведев И.Л.* Параллельные вычислительные системы с общим управлением. — М.: Энергоатомиздат, 1983.
175. *Прангишвили И.В.* Микропроцессоры и микроЭВМ. — М.: Энергия, 1979.
176. *Прангишвили И.В.* Применение микропроцессоров в приборостроении//Приборы и системы управления. — 1981. — № 2. — С. 48 — 50.
177. *Прангишвили И.В., Стецюра Г.Г.* Микропроцессорные системы. — М.: Наука, 1980.
178. Применение цифровой обработки сигналов/Под ред. Э. Опенгейма. — М.: Мир, 1980.
179. Программируемая система управления экспериментом и накопление данных на основе микропроцессора//Приборы для научных исследований. — 1978. — № 11. — С. 46 — 53.
180. *Пугачев В.С.* Теория случайных функций и ее применение к задачам автоматического управления. — М.: Физматгиз, 1962.
181. *Пушняк В.А., Цветов Э.И.* Автоматизация научных исследований//Приборы и системы управления. — 1983. — № 4. — С. 39 — 40.
182. *Рабинер Л., Гоулд Б.* Теория и применение цифровой обработки сигналов/Пер. с англ. — М.: Мир, 1978.
183. Разработка адаптивного контроллера//Электроника. — 1978. — № 13. — С. 16 — 17.
184. Разработка 16-разрядных микропроцессоров для бортовых авиационных ЭВМ//Электроника. — 1982. — № 25. — С. 8 — 9.
185. "Разумный" буфер, согласующий быстрые процессы и медленные периферийные устройства//Электроника. — 1980. — № 20. — С. 52.
186. *Ралланали К., Крегер Д.* Вспомогательные процессоры для быстрого выполнения арифметических операций в микропроцессорных системах//Электроника. — 1980. — № 10. — С. 43 — 49.
187. *Родригес-Искиердо.* Необычный способ прерываний, ускоряющий передачу данных в микропроцессорных системах//Электроника. — 1980. — № 8. — С. 69 — 70.
188. *Розин Б.Б.* Теория распознавания образов в экономических исследованиях. — М.: Статистика, 1973.
189. *Ролофф Д.* Управление памятью, обеспечивающее полное использование возможностей микропроцессоров//Электроника. — 1980. — № 8. — С. 53 — 59.
190. *Ростокин Б.И.* Операционные системы реального времени УВК СМ-3 и СМ-4//Приборы и системы управления. — 1977. — № 12. — С. 1 — 2.

191. *Русаков А.А.* Спутниковая навигационная система Navstar// Зарубежная радиоэлектроника. – 1978. – № 1. – С. 4 – 24.
192. *Саверченко В.Г.* Микропроцессоры в конвейерных вычислительных системах//Автоматика и электроприборостроение. – Киев: КПИ, 1983. – № 20. – С. 44 – 47.
193. *Саркисова А.Г., Соснина Н.С.* Цифровые приборы с микропроцессорами за рубежом//Приборы и системы управления. – 1982. – № 6. – С. 14 – 16.
194. Серийная микросхема цифрового фильтра//Электроника. – 1982. – № 6. – С. 21 – 22.
195. *Сигалов Г.Г., Мадорский Л.С.* Основы теории дискретных систем управления. – М.: Высшая школа, 1973.
196. *Смолов В.Б., Узрюмов Е.П., Фомичев В.С.* и др. Микроэлектронные цифро-аналоговые и аналого-цифровые преобразователи информации. – Л.: Энергия, 1976.
197. *Соботка З., Стары Я.* Микропроцессорные системы. – М.: Энергоиздат, 1981.
198. *Солдатенко Л.М., Захаров В.П.* Микрокалькулятор "Электроника МК-46" для автоматизации ТП//Приборы и системы управления. – 1983. – № 11. – С. 32 – 33.
199. Сопряжение 12-рядного аналого-цифрового преобразователя с микропроцессорами//Э.И. Приборы и элементы автоматизации и вычислительной техники. – 1982. – № 2. – С. 18 – 31.
200. *Соучек Б.* Микропроцессоры и микроЭВМ/Пер. с англ. – М.: Советское радио, 1979.
201. *Танкелевич Р.Л.* Моделирующие микропроцессорные системы. – М.: Энергия, 1979.
202. Теория автоматического регулирования. Т. 2: Анализ и синтез непрерывных и дискретных систем автоматического регулирования//Под ред. В.В. Солодовникова. – М.: Машиностроение, 1967.
203. *Тихонов В.С., Гильман Г.И., Альгиуль С.Д.* Применение микроЭВМ в системах автоматического контроля и управления//Приборы и системы управления. – 1979. – № 5. – С. 9 – 10.
204. *Томас Д.Э.* Автоматизированный синтез цифровых фильтров//ТИИЭР. – 1981. – № 10. – С. 20 – 24.
205. Три микропроцессора на борту спутника//Электроника. – 1978. – № 1. – С. 20 – 24.
206. *Трохименко Я.К., Любич Ф.Д.* Радиотехнические расчеты на микрокалькуляторах. – М.: Радио и связь, 1983.
207. *Тузов Г.И., Евдин В.В.* Обработка сигналов в системах радиосвязи с помощью микропроцессорных систем//Зарубежная радиоэлектроника. – 1981. – № 11. – С. 3 – 20.
208. *Турута Е.Н.* Мультипроцессорные системы//Зарубежная радиоэлектроника. – 1979. – № 3. – С. 3 – 18.
209. *Хилбурн Дж., Джулич П.* МикроЭВМ и микропроцессоры/Пер. с англ. – М.: Мир, 1979.
210. *Хьюз Ф., Дьюген Т., Хэнди Д., Паскуалини Р.* Однокристалльный микрокомпьютер, предназначенный для многопроцессорной работы//Электроника. – 1980. – № 22. – С. 66 – 71.
211. Цифро-аналоговые преобразователи, сопрягающиеся с микропроцессором//Электроника. – 1980. – № 14. – С. 56.
212. Цифровая коррекция погрешностей в системах сбора данных//Электроника. – 1979. – № 24. – С. 47.
213. Цифровые устройства обработки сигналов в системах управления//Под ред. В.А. Бесекерского. – Л.: ЛИАП, 1982.

214. Цифровые фильтры в электросвязи и радиотехнике/Под ред. Л.М. Гольденберга. — М.: Радио и связь 1982.
215. *Полис.* МР-16 в радионавигационной системе малых самолетов//Электроника. — 1977. — № 12. — С. 58 — 63.
216. *Цыпкин Я.З.* Теория линейных импульсных систем. — М.: Физматгиз, 1963.
217. *Цыпкин Я.З., Попков Ю.С.* Теория нелинейных импульсных систем. — М.: Наука, 1973.
218. *Чернов В.Г.* Принципы организации устройств аналогового ввода для микропроцессорных систем сбора и обработки данных (обзор)//Приборы и системы управления. — 1980. — № 12. — С. 9 — 12.
219. *Шарашенидзе Т.Э., Ревазшвили Г.Г., Тодуа Д.А.* Электронная вычислительная машина ПС-315//Приборы и системы управления. — 1982. — № 8. — С. 33.
220. *Шпилевский Э.К.* Динамическая классификация стохастических процессов и систем в дискретном времени//А и Т. — 1980. — № 11. — С. 46 — 53.
221. *Шпилевский Э.К.* Динамическая классификация стохастических процессов и систем в дискретном времени. 2//А и Т. — 1980. — № 2. — С. 45—54.
222. *Щетинин Ю.И., Березенко А.И., Базанов В.И.* и др. Схемотехника микропроцессорного комплекса БИС TTL с диодами Шоттки//Микроэлектроника и полупроводниковые приборы. — М.: Советское радио, 1979. — Вып. 4. — С. 56 — 62.
223. *Эванчук С.* Операционные системы реального времени. Обзор//Электроника. — 1983. — № 6. — С. 26.
224. Электрические обогревательные приборы с цифровым управлением//Электроника. — 1980. — № 3. — С. 13.
225. *Эрини И.* Методы и средства проектирования цифровых устройств на базе микропроцессоров//Приборы и системы управления. — 1980. — № 8. — С. 1 — 3.
226. *Юсифов С.И.* Особенности архитектуры и структуры микропроцессоров и микроЭВМ и их классификации//Управляющие системы и машины. — 1981. — № 6. — С. 60 — 64.
227. *Якубайтис Э.А., Баумс А.К.* Микропроцессорные средства. Достижения и проблемы//Приборы и системы управления. — 1978. — № 6. — С. 1 — 3.
228. *Fritzsch W.* Anlagenstrukturen und Steuerungs Konzepte mit Mikroprozessorrechnern//Messen—Steuern—Regeln. — 1977. — № 12. — S. 662—667.
229. *Parasuraman B.* Hardware multiplication techniques for microprocessor systems. — Compet. Des., 1977, April, 75—82.
230. *Pritti D.W.* A continuous path microprocessor numerical contr. — Glasgow: Department of Computer Science University of Strathclyde, Libingstone Tower, 1975.
231. *Queyssac D.* Comprendre les microprocesseurs//Mesures—Regulation — Automatisme. — 1977. — V. 42, № 5. — P. 47—52.
232. *Roth M.* Mikroprozessoren. Wissenschaftliche Zeitschrift Technische Hochschule Ilmenau. — Ilmenau: THI, 1979. — 256 S.
233. *Stengel R.F.* Digital flight control research using microprocessor technology//IEE Trans. an Aerospace and Electronic Systems. — 1979. — HES-15, № 3. — P. 397—403.
234. *Takashi R., Osati O., Yosiaki O.* Метод синтеза и разработки программного обеспечения для микропроцессора цифровой системы управления ориентацией КА//Э.-И. Астронавтика и ракетодинамика. — 1979. — № 30. — С. 25—34.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

- Автокод 80
Автоматизация обработки данных 29, 44
– программирования 157
– проектирования 157
– управления экспериментом 29
Адаптация 18, 169
Адресация автодекрементная 128
– автоинкрементная 128
– индексная 128
– косвенная 80, 129
– непосредственная 80, 129
– относительная 129
– прямая 80
– – регистровая 128
Аккумулятор 87, 89
Алгоритм 7
– выбора 41
– дифференцирования 201
– интегрирования точного 222
– – цифрового 219
– классификации 43, 253
– Ремеза 234
– универсальный 69
Анализ гармонический 31
Аппарат космический 25
Архитектура 7
– вычислительной системы 49
Ассемблер 8, 48, 124, 142
– перемещающий 147
- Байт 120
Библиотека программ 23
БИС 6, 39
Бит 8, 120
БИХ-фильтр 234
Буй звуколокაციонный 29
Буфер высокоскоростной 66
Быстродействие микроЭВМ 44, 66, 79
– преобразователя 114
- Ввод-вывод аппаратно управляемый 102
– – с квитиowaniem 100, 111
– данных асинхронный 98, 113
Вероятность ложной тревоги 43
– отказов 48
– правильной классификации 43
Восстановление сигналов 37
Время установления 119
Выбор архитектуры 45
– микропроцессорных средств 38, 73
– операционной системы 50
– периода дискретизации 44
– по быстродействию 39
– средств проектирования 38
– числа уровней квантования 42
Выявление рассогласований 37
- Генератор тактовый 29
Гиббса явление 230
Гибкость 16, 29, 40, 58, 70
Глитчер 119
Головка самонаведения 28
Графопостроитель 30
Грубость системы 283
- Данные 8
Датчик 36
– интеллектуальный 25, 36, 44
Деглитчирование 119
Декодирование команд 86
Децентрализация управления 19
Дешифратор адреса 68
Диагностирование 17, 30
Диалог начальный 147
Диз-ассемблирование 157
Директива ассемблера 148
Дискретизатор кривых 30
Дискретизация по времени 42
Дисперсия ошибки 198, 276
– – дополнительной 211
– – интегрирования 225

- Дисперсия ошибки округления 209
 - шумовой 210
 - процесса 193, 195
- Диспетчер 46
- Длина разрядной сетки 42, 75
 - – наращиваемая 90
 - слова 73, 76
- Длительность такта 79
- Добротность 192
- Доступ к памяти прямой 46, 97

- Загрузчик 80
 - абсолютный 145
- Задание программных функций 37
- Задача идентификации 33
 - распределения функций 45
 - сбора информации 20
- Задачи микроЭВМ в САУ, 48, 169
 - сопряжения 47
- Закон Симпсона 209
- Запас устойчивости 42, 192, 261
- Запятая плавающая 122
 - фиксированная 122
- Звено ограничено линейное 173

- Идентификация сигналов 37, 59
- Избыточность аппаратурная 18, 40
 - программная 18, 40
- Интеграл вероятности 255
- Интегратор цифровой 219
- Интегрирование 221
 - двойное 224
- Интеллектуализация систем 16
- Интерпретатор 30
- Интерфейс 7, 56, 97, 109
 - аналоговый 28
 - ввода-вывода 95, 99, 107
 - последовательный 29
 - программируемый 107, 109

- Качество управления 42
- Квантование по времени 169, 267
 - по уровню 42, 114, 170, 196
- Квитирование 96
- КИХ-фильтр 229
- Классификатор 43
- Классификация динамическая 246
 - сигналов 37, 245
- Кодер линейный 115
- Кодирование 15, 37
- Код дополнительный 121
 - маски 92
 - машинный 123
 - операнда 120
- Код операции 120
 - параллельный 94
 - последовательный 94
- Команда 8, 131
- Команды арифметические 131
 - битовых операций 131
 - логические 131, 134
 - пересылки 132
 - переходов 131
 - сдвига 131
 - специальные 134
- Коммутатор 58
- Коммутация перекрестная 58
- Компаратор амплитудный 115
- Компилятор 8
- Комплект БИС 6
- Конструкция модульная 18
- Контроллер ПДП 102
- Коррекция неисправностей 243
 - цифровая 283
- Коэффициент готовности 12
 - передачи интегратора 226
 - – линеаризованный 175
 - производительности 49
 - системных затрат 51
 - увеличения периода дискретности 227
- Коэффициент усиления предельный 274
- Коэффициенты ошибок 192
- Критерий быстродействия 262
 - Неймана – Пирсона 43, 255
 - оптимальности Чебышева 233
 - сообразности 244
- Кросс-ассемблер 161
 - диз-ассемблер 161
 - компилятор 8, 161
- Кэш-память 65

- Линеаризация гармоническая 170
 - характеристик 37
- Линейность преобразования 47
- Листинг 148
- Логика аппаратная 18, 49
 - жесткая 18, 49, 51
 - программируемая 18, 49, 56, 70
 - флажков 91, 140

- Масштабирование сигналов 37
- Матрица компараторов 115
 - циклическая 55
- Метка 143
- Метод адресации 125
 - билинейного преобразования 235

- Метод взвешивания 229, 230
 - инвариантного преобразования 235
 - отображения дифференциалов 235
 - умножения 53.
- частотной выборки 230
- Методы численного интегрирования 60**
- Механизм исполнительный 22
- Микроассемблер 138, 142
- Микроинструкция 92
- Микрокоманда 92
- Микрокомпьютер 82
- Микроконтроллер 11
 - адаптивный 44
 - периферийный 25
 - регулирующий 23
 - специализированный 24
- Микрооперация 92
- Микропрограмма 92
- Микропрограммирование внешнее 83
 - внутреннее 86
- Микропроцессор 6, 70
 - многокристальный 44, 91
 - однокристалльный 44, 76, 85, 91
- Микросхема 5
 - цифровая 5
- МикроЭВМ 7
 - автономная 11
 - виртуальная 7
 - лабораторная 82
 - проблемно ориентированная 74
 - серийная 74
 - с перестраиваемой структурой 18
 - специализированная 7, 74
 - универсальная 7, 74
 - управляющая 7, 152
 - встраиваемая 7
- Мини-ЭВМ 7
- Мнемоника 8
 - управления центральная 9
- Моделирование 170
- Модель системы ориентации 25
- Модуль базовый 20
 - загрузочный 35
 - основного времени 35
 - резервного времени 35
 - связи 20
- Монитор 30, 160
- МОП-структура 55
 - технология 70
- Мощность вычислительная 49
- Мультиплексор 21
- Набор команд 120**
- Наведение ракеты 28
- Навигация самолета 26
- Надежность систем 12, 48
 - технологическая 18
- Обеспечение дистанционного управления 34**
 - заданных показателей 42, 44
 - математическое 17
 - программное 17, 23, 28
 - – прикладное 49, 68, 154
 - – системное 50, 144
 - – – резидентное 80, 144
- Обмен данными асинхронный 96, 100
 - – по прерыванию 103
 - – программно управляемый 96, 113
 - – синхронный 96, 109
 - – с квитированием 118
 - – циклический 104
- Обнаружение неисправностей 243
- Обработка байтовая 123
 - информации конвейерная 12, 59
 - – параллельная 12
 - результатов статистическая 32
- Объем памяти 40, 75
- Операнд 7, 72, 123, 125
- Оператор дифференцирования 199
 - интегрирования 219
 - преобразования Лапласа 177
- Операция взвешенного суммирования 39
 - умножения 39
 - элементарная 79
- Оптимизация вторичная 269
 - процесса дифференцирования 211
- Отказ деградационный 48
 - кратковременный 243
- Оценивание корреляционной функции 15
 - статистическое 43
- Ошибка вычислений 244
 - интегратора шумовая 225
 - интегрирования 220, 223
 - – общая 220
 - квантования 208
 - максимальная 213
 - округления 196
 - прогноза 251
- Пакет данных 102**
- Память 6, 71
- Память блокнотного типа 46
 - виртуальная 66
 - внешняя 73
 - матричная 46

- Память микропрограмм 24
 - многоходовая 58
 - основная 73
 - программ 73
 - регистровая 73
 - сверх оперативная 87
 - стековая 39, 88
- Пеленгование цели 29
- Передача данных 94
 - информации 34
- Период дискретности 15, 168, 176
 - квантования по времени 176
- Планировщик задач 66
- Плотность спектральная 194
 - упаковки элементов 76, 115
- Повышение помехоустойчивости 33, 56
- Подпрограмма работы 28
- Поиск табличный 55
- Показатель эффективности 63
- Полинг 106
- Порог принятия решений 43, 255
- Порт 73, 105, 109
 - многофункциональный 66
- Правило решающее 255
- Преимущества ЭВМ 8
- Преобразование Фурье 29, 49, 194
- Преобразователь аналоговых сигналов 114, 117
 - аналого-цифровой 15, 21, 47, 94
 - угол – код 117
 - цифровых сигналов 119
- Прерывание 8
 - векторное 106
 - многоуровневое 104
 - одноуровневое 104
- Признаки 130
- Принцип децентрализованного управления 12.
 - управления аппаратный 75, 86
 - микропрограммный 76
- Принципы выбора средств 40
- Приоритет 35
- Проверка формата 30
- Прогнозирование параметров 37
 - траекторий 37
- Прогноз оптимальный 250
- Программа 7
 - ассемблирования 147
 - ВЕРТЕКС 67
 - диагностики 23
 - диспетчера 23
 - исполнительная 30
 - исходная 144
- Программа микроассемблер 144
 - монитор 82, 160
 - обслуживания гибкая 168
 - – жесткая 168
 - – объектная 150
 - отладки 150
 - перфорации 150
- Программа прикладная 156
 - рабочая 144
 - редактора текста 146
 - системная 30
- Проектирование фильтров 229
- Производительность управляющих микроЭВМ 7, 13, 48
 - управляющих микроЭВМ 13
- Производство гибкое (ГАП) 25
- Протез руки 32
- Процесс передачи данных 68
 - переходный 189
- Процессор 70, 168
 - центральный 71
- Нуль оператора 23
 - отладки 159
- Радиомаяк 26
- Радионавигация 26
- Разделение гальваническое 22
- Разряд 8
- Ракета тактическая 28
- Распараллеливание задач 44, 57
- Распознавание объектов 25
- Реализация алгоритмов 45
- Регистр 6, 35, 73
 - буферный 106
 - защелка 111
 - источник 97, 125
 - накопительный 87
 - общего назначения 6, 86
 - прерывания 104
- Регистр-приемник 97, 125
 - связи 134
- состояния 87, 103
- управляющего слова 109
- Регулятор одноконтурный 21
- Режим мультиплексный 80
 - прерывания 104, 118
 - прямого доступа 96, 102
 - реального времени 47, 82
 - редактирования 146
 - селекторный 80
- Резервирование 17
- Ремиконт 21, 23
- Ремонтопригодность 18
- Рентабельность ремиконта 23

- Робот 25
- Ряд микроЭВМ 81
- Самоконтроль 34, 37
- Самолет ПЛЮ 28
- Сбой деградационный 48
- Свертка функций 181
- Сглаживание 264
- Сжатие информации 37
- Сигнал каузальный 249
 - переноса 56
 - управления внешний 86
 - – внутренний 86
- Синтез динамический 260
 - цифровых фильтров 229
- Синхронизация работы микроЭВМ 101
- Система автоматизированная 33
 - – проектирования (САПР) 17, 154
 - – – комбинированная 48
 - – – резидентная 47
 - адаптивная 15
 - вычислительная 7, 29
 - – ассоциативная 59
 - – двухпроцессорная 52
 - команд 46, 77
 - микроконтроллерная 11
 - многомашинная 13
 - многомерная несвязанная 9, 168
 - – связанная 9, 168
 - многомикропроцессорная 12, 40, 57
 - наведения инерциальная 28
 - одноконтурная 9, 15, 168
 - операционная 50, 57, 145
 - перфоленточная 82
 - позиционного управления 24
 - прерываний 51, 104
 - радионавигационная 28
 - с перестраиваемой структурой 15, 45
 - управления ориентацией 25
 - – робастная 271
 - управляемая свистом 33
 - ЧПУ станками 84
- Словарь терминов 5
- Слово 73, 120
 - данных 121
 - машинное 6
 - состояния процессора 130
- Слово управляющее 103
- Сопряжение программное 47
 - схемотехническое 47
- Состав микроЭВМ 71
- Спецификация системы 150
- Спецпроцессор 51, 53
- Список команд базовый 122
- Способность пропускная 7
 - разрешающая 47, 114, 119
- Средства аппаратные 49, 152
 - – комбинированные 51
 - – прикладные 49
 - – системные 49
 - кроссовые 81, 83
 - микропроцессорные 74
 - планирования задач 66
 - программные 152
 - резидентные 81, 83
- Станция измерительная 20
- Стек 87, 88
- Степень интеграции 5, 83
 - функциональной сложности 6
- Стоимость программирования 17
 - систем 12
- Строб 101, 113
- Структура 7
 - векторная 59
 - виртуальная 9, 15
 - матричная 59
 - ремиконта 23, 35
 - цифровой САУ 12, 45
- Схема алгоритма машинно ориентированная 154
 - – структурная 154
 - – функциональная 154
 - интегральная 5
 - – большая (БИС) 6
 - – – заказная 6
 - – – унифицированная 6
 - интерфейса 97
 - функциональная 151
- Счетчик команд 87
 - комбинированный 54
 - программный 24
 - слов 103
 - указатель адреса 103
- Таймер 31
- Теорема Котельникова 114
- Тестирование 17, 157
- Техника авиационно-космическая 25
 - медицинская 32
- Технология биполярная 76, 115
 - – модифицированная 115
 - И²Л 83
 - интегральная 17
 - ТТЛШ 80
- Топология сетевая 15
 - цифровых САУ 14
- Точность вычисления потенциальная 217

- Точность преобразования 119
 – системы установившаяся 192
- Транслятор 8, 80
- Трансляция 142
- Требования к САУ 16
- Тревога ложная 255
- Управление мажоритарное 12**
 – микропрограммное 90
 – многоконтурное 20
 – обогревом 33
 – полетом самолета 28
 – по прерываниям 67
 – протезом 32
 – сложными объектами 9
 – технологическими процессами 19
 – энергетикой страны 34
- Упреждение 264
- Уровень квантования 42, 114
 – параллелизма 66
 – шума квантования 42
- Усилитель буферный 102
- Условия оптимума 217
- Устойчивость программы вычислений 287
 – системы 191
- Устройство арифметико-логическое 72, 86
 – внешнее 94
 – запоминающее (ЗУ) 6
 – – оперативное (ОЗУ) 6, 73
 – – постоянное (ПЗУ) 6, 73
 – – – перепрограммируемое 6, 73
 – – сверхоперативное 6
 – классификации адаптивное 256
 – навигационное 28
 – управления (УУ) 72, 86
 – фотосчитывающее 145
- Уход интегратора 226
- Файл 8**
- Фильтрация адаптивная 36**
 – винеровская 263
 – сигналов 36
 – цифровая 42
- Фильтр Баттлерворта 236**
 – Винера 263
 – – цифровой 266
 – импульсный 182
 – оптимальный по Чебышеву 234
 – физически реализуемый 230
 – цифровой 51, 169, 227
- Флаг (флажок) 88**
- Формат команды 80, 130**
 – микроинструкции 134
 – слова 109
- Формат управляющего слова 120, 122**
- Формирование порогового уровня 43**
- Формула обращения 181, 194**
 – разложения 189
 – Рэлея 182
- Функции интеллектуальные 37**
 – микроЭВМ нетрадиционные 34
 – – основные 36
 – – традиционные 36
 – сервисные 37
- Функция весовая приведенная 183**
 – дискриминантная 255
 – корреляционная 193
- Функция передаточная фильтра 183, 228**
 – решетчатая 176
 – управления адресом 140
- Характеристика вычислительной мощности 49**
 – линеаризованная 172
 – преобразователя статическая 171
 – частотная 187
 – – логарифмическая 187
- Частота дискретизации 47, 114**
 – среза 282
 – тактовая 77
- Число машинных циклов 62**
- Шаг интегрирования 61**
 – квантования 43
- Шина 57, 71**
 – адреса 89
 – данных внешняя 89
 – – внутренняя 88
- Шифратор приоритетов 106**
- Шум дискретный белый 195**
 – квантования 42, 170, 199, 263
 – округления 42
- Экономичность 16, 58**
- Эксперимент полунатурный 47**
- Экстраполятор 184**
- Элемент импульсный 169**
 – процессорный центральный 92
- Эмулятор 161**
- Эмуляция 8, 163**
- Эталон динамический 246**
- Эффективность вычислительной системы 63**
 – эксплуатации 18
- Явление Гиббса 230**

ОГЛАВЛЕНИЕ

Предисловие.	3
Глава 1. Общие сведения о системах автоматического управления с микроЭВМ.	5
§ 1.1. Структура цифровых систем управления.	5
Терминология (5). Влияние структуры цифровых систем на показатели стоимости и надежности (8). Два класса структур цифровых САУ (12). Автоматические системы с перестраиваемой структурой (15).	
§ 1.2. Задачи, решаемые микроЭВМ в системах автоматического управления.	16
Основные требования, предъявляемые к САУ с микроЭВМ (16). МикроЭВМ в промышленных системах автоматического управления и регулирования (19). Применение микроЭВМ в авиационно-космической технике (25). МикроЭВМ в автоматизированных системах научных исследований (29). Другие области применения микроЭВМ в автоматических системах (32). Обобщенные функции микроЭВМ в системах автоматического управления (36).	
§ 1.3. Проблемы разработки и реализации систем автоматического управления с микроЭВМ.	38
Особенности использования микропроцессоров и микроЭВМ в САУ (38). Проблема выбора микропроцессорных средств (39). Обеспечение заданных показателей качества управления (42). Рациональное распределение функций САУ между аппаратными и программными средствами (44). Выбор архитектуры управляющей микроЭВМ (45). Сопряжение цифровой и аналоговой частей САУ (47). Схемотехническое проектирование, разработка программного обеспечения и отладка САУ с микроЭВМ (47). Новые акценты в проблеме надежности микропроцессорных систем (48).	
§ 1.4. Повышение производительности микроЭВМ в САУ.	48
Архитектура и производительность микроЭВМ в САУ (48). Аппаратные средства реализации прикладных алгоритмов (52). Вычислительные структуры с программируемой логикой (56). Аппаратные средства планирования задач (66). Прикладное программное обеспечение (68).	
Глава 2. Построение систем управления с микроЭВМ.	70
§ 2.1. МикроЭВМ (состав, назначение, выбор и оценка).	70
Особенности подхода к проектированию цифровых авто-	

	матических систем (70). Состав и назначение отдельных частей микроЭВМ (71). Основные характеристики, принципы выбора и оценка микроЭВМ (73). Микропроцессор с аппаратным принципом управления (86). Микропроцессор с микропрограммным управлением (90).	
§ 2.2.	Ввод-вывод микроЭВМ в структуре систем автоматического управления Предварительные замечания (94). Типовая схема БИС интерфейса (97). Синхронный обмен данными (98). Асинхронный программно управляемый обмен данными (ввод-вывод с квити́рованием) (100). Асинхронный аппаратно управляемый ввод-вывод. Режим прямого доступа к памяти (102). Обмен по прерыванию (103). Организация схем ввода-вывода (107). Быстродействующие преобразователи аналоговых сигналов. Сопряжение с микроЭВМ (114). Быстродействующие преобразователи цифровых сигналов (119).	94
§ 2.3.	Программное обеспечение микроЭВМ Представление информации в микроЭВМ (120). Программирование микроЭВМ с постоянным набором команд (122). Функции команд микроЭВМ (131). Программирование микроЭВМ с изменяемым набором команд (134) Структура команд языков Ассемблер и Микроассемблер (142). Системное программное обеспечение (144).	120
§ 2.4.	Этапы разработки и отладки микроЭВМ для САУ Порядок разработки САУ с микроЭВМ (150). Этапы разработки управляющих микроЭВМ (152). Разработка схем алгоритмов прикладных задач (154). Функции и классификация систем автоматизированного проектирования (САПР) управляющих микроЭВМ (156). Проектирование с использованием пульта отладки (159). Кроссовые САПР (161). Резидентные системы автоматизированного проектирования (162). Комбинированные САПР (165).	150
Глава 3.	Методы исследования систем управления с микроЭВМ.	168
§ 3.1.	Вводные замечания Структура одномерной ЦАС (168). Задачи, решаемые ЦВМ (169). Методы исследования ЦАС (169). Линеаризация ЦАС (171).	168
§ 3.2.	Математический аппарат Основные понятия (176). Использование z-преобразования (177). Импульсные фильтры (182). Передаточные функции непрерывной части ЦАС (183). Передаточные функции ЦВМ (185). Передаточные функции ЦАС (187). Частотные характеристики (187).	176
§ 3.3.	Исследование детерминированных процессов Построение переходных процессов (189). Устойчивость ЦАС (191). Оценка качественных показателей (192).	189
§ 3.4.	Исследование случайных процессов Статистические характеристики случайных процессов (193). Типовые случайные процессы (195). Расчет ошибок ЦАС (197).	193

Глава 4. Реализация на микроЭВМ элементов и устройств систем управления	199
§ 4.1. Дифференцирование цифровых последовательностей	199
Общие сведения (199). Алгоритмы дифференцирования (201). Методические ошибки дифференцирования (202). Влияние шумов квантования (208). Оптимизация процесса дифференцирования (211).	
§ 4.2. Цифровые интеграторы	219
Общие сведения (219). Алгоритмы точного интегрирования (222). Двойное интегрирование (224). Влияние шумов квантования (225). Использование интеграторов в законах управления (226).	
§ 4.3. Цифровые фильтры	227
Два класса цифровых фильтров (227). Элементы теории фильтров с конечной импульсной характеристикой (229). Общая характеристика методов расчета БИХ-фильтров (234). Виртуальные структуры некоторых цифровых фильтров (238). Требования к точности задания коэффициентов цифровых фильтров (240).	
§ 4.4. Обнаружение и коррекция неисправностей	243
Кратковременные отказы (243). Ошибки вычислений (244).	
§ 4.5. Устройства классификации случайных сигналов	245
Постановка задачи (245). Математические модели классифицируемых сигналов (247). Статистические характеристики каузальных сигналов (249). Алгоритмы классификации (253). Реализация алгоритмов классификации на микроЭВМ (258).	
Глава 5. Динамический синтез цифровых систем автоматического управления	259
§ 5.1. Общие сведения о синтезе систем управления	259
Статические и динамические требования (259). Другие требования при синтезе (261). Критерии запаса устойчивости (261). Критерии быстродействия (262). Особенности использования микроЭВМ (263).	
§ 5.2. Использование цифровых винеровских фильтров	263
Основные соотношения для винеровских фильтров (263). Цифровые винеровские фильтры (266). Определение периода дискретности и вторичная оптимизация (269).	
§ 5.3. Робастные цифровые системы автоматического управления. Общие сведения (271). Обеспечение требуемой точности (271). Использование типовых передаточных функций (275). Обеспечение запаса устойчивости (281).	271
§ 5.4. Программы для микроЭВМ, реализующие алгоритмы управления	283
Цифровая коррекция (283). Цифровые корректирующие обратные связи (285).	
Приложение 1: Список команд микроЭВМ LSI-11	289
Приложение 2: Изображения решетчатых функций	294
Приложение 3: Таблица интегралов	298
Список литературы	300
Предметный указатель	311
	319

Виктор Антонович Бесекерский
Виталий Васильевич Изранцев

**СИСТЕМЫ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ
С МИКРОЭВМ**

Серия "Теоретические основы технической кибернетики", № 88

Редактор *Ф.С. Петров*
Художественный редактор *Т.Н. Кольченко*
Технический редактор *С.В. Геворкян*
Корректоры *Н.П. Круглова, Т.В. Обод*

Набор осуществлен в издательстве
на наборно-печатающих автоматах

ИБ № 12600

Сдано в набор 12.09.86. Подписано к печати 24.12.86. Т-23998
Формат 84 × 108 1/32. Бумага офсетная
Гарнитура Пресс-Роман. Печать офсетная
Усл.печл. 16,80. Усл.кр.-отт. 16,80. Уч.-издл. 19,47
Тираж 18200 экз. Тип. зак. Цена 1 р. 60 к.

Ордена Трудового Красного Знамени
издательство "Наука"
Главная редакция физико-математической литературы
117071 Москва В-71, Ленинский проспект, 15

4-я типография издательства "Наука"
630077 г. Новосибирск-77, ул. Станиславского, 25

