

Е. А. Дроздов,
В. А. Комарницкий,
А. П. Пятибратов

ЭЛЕКТРОННЫЕ
ВЫЧИСЛИТЕЛЬНЫЕ
МАШИНЫ
ЕДИНОЙ СИСТЕМЫ



Москва * Машиностроение * 1976

6Ф7.3
Д72
УДК 681.3

Редактор инж. Е. Е. Скворцов
Рецензент канд. техн. наук П. А. Чукреев

Д72

Дроздов Е. А., Комарницкий В. А., Пятибратов А. П.

Электронные вычислительные машины Единой системы. М., «Машиностроение», 1976.
672 с. с ил.

В книге изложены основы выполнения операций, принципы построения и функционирования ЭВМ Единой системы. Рассмотрены общие схемы машин Единой системы, их элементная база, кодирование информации и машинные алгоритмы выполнения различных операций; особенности построения, основные функциональные и структурные схемы узлов и устройств машин, а также унифицированного набора внешних устройств, структура и возможности математического обеспечения, способы и аппаратура организации совместной работы нескольких машин.

Материал базируется на схемах, организации и возможностях использования моделей ЕС-1050, ЕС-1030, ЕС-1020.

Книга предназначена для тех, кто готовится к эксплуатации машин Единой системы, а также будет полезна для специалистов, уже работающих на таких машинах.

Д $\frac{30502-001}{038(01)-76}$ 1-76

6Ф7.3

ПРЕДИСЛОВИЕ

Развитие средств вычислительной техники привело к созданию электронных вычислительных машин, которые по линии машин общего назначения образуют Единую систему ЭВМ (ЕС ЭВМ). Машины Единой системы как семейство ЭВМ с быстродействием от нескольких десятков тысяч до полутора миллионов операций в секунду могут эффективно использоваться не только автономно или в составе автоматизированных систем управления, но и при организации многомашинных и многопроцессорных комплексов. В создании этих машин принимали активное участие коллективы ученых, инженеров и рабочих стран социалистического содружества — Болгарии, Венгрии, ГДР, Польши, СССР и Чехословакии.

Различные модели ЭВМ Единой системы отличаются по своей производительности, но имеют общую конструктивную и элементную базу, построены по аналогичным общим схемам и являются программно-совместимыми. Это положение принято в качестве базового при построении материала настоящей книги, посвященной основам устройства и организации функционирования машин Единой системы.

В книге изложены основы выполнения операций, принципы построения и функционирования всех основных устройств и машин в целом, а также особенности организации и возможности их математического обеспечения. Материал книги базируется главным образом на схемах и структурах машин ЕС-1020, ЕС-1030, ЕС-1050; кроме того, рассматриваются вопросы, относящиеся ко всем моделям ЭВМ Единой системы, в том числе вопросы телеобработки, контроля, построения вычислительных систем и организации их функционирования.

При написании книги авторы стремились дать общие основы построения, функционирования и использования моделей ЭВМ

Единой системы, раскрыть их особенности, выявить и представить в общем виде их специфические узлы, блоки и устройства. В то же время материал книги не представляет собой справочник по всем моделям или по какой-либо одной модели ЭВМ Единой системы, так как назначение книги — ввести читателя в круг идей и решений, принятых при создании ЕС ЭВМ, с тем, чтобы облегчить углубленное изучение отдельных конкретных вопросов или отдельных моделей ЭВМ.

Книга рассчитана на читателей, знакомых с основами построения и применения цифровых вычислительных машин и систем. Она может использоваться не только при подготовке кадров для эксплуатации моделей ЭВМ, но и в учебных заведениях и научно-исследовательских организациях.

Авторы выражают искреннюю признательность авторам проектов ЕС ЭВМ, и особенно В. С. Антонову, С. П. Соловьеву, И. С. Храмцову, С. А. Цареву за полезные консультации и советы, рецензенту кандидату технических наук П. А. Чукрееву и редактору Е. Е. Скворцову, в значительной степени способствовавшим обработке материала книги.

Введение, гл. 1, 5, 11, 14, 17, 18 написаны А. П. Пятибратовым, гл. 2, 3, 4, 6, 15, 16 — В. А. Комарницким, гл. 7, 8, 9, 10, 12 и 13 — Е. А. Дроздовым. В приложении табл. 1, 2, 3, 4, 5 и 6 составлены А. П. Пятибратовым, табл. 7, 8, 9 и 10 — В. А. Комарницким.

ВВЕДЕНИЕ

Научно-технический прогресс, развитие современной научной мысли, развитие практически всех отраслей народного хозяйства тесно связано с использованием электронных вычислительных машин (ЭВМ) и вычислительных систем (ВС). Можно утверждать, что успешное разрешение современных научных и технических проблем в значительной степени зависит от уровня развития электронной вычислительной техники. В связи с этим в нашей стране уделяется большое внимание развитию и совершенствованию средств вычислительной техники и их математическому обеспечению. Из года в год расширяется парк ЭВМ, совершенствуется их техническая база — микроэлектроника, создаются все новые и новые вычислительные центры и автоматизированные системы управления (АСУ) на основе ЭВМ.

На пути развития электронной вычислительной техники можно выделить четыре поколения ЭВМ, отличающиеся элементной базой, конструктивно-технологическим исполнением, логической организацией, математическим обеспечением, техническими характеристиками, степенью доступа к ЭВМ со стороны пользователей (абонентов).

Смене поколений сопутствовало изменение основных технико-экономических показателей ЭВМ и в первую очередь таких, как быстродействие, надежность и стоимость. При этом одной из основных тенденций развития было и есть стремление уменьшить трудоемкость подготовки программ решаемых задач, облегчить связь операторов с машинами, повысить эффективность использования последних. Это диктовалось и диктуется постоянным ростом сложности и трудоемкости задач, решение которых возлагается на ЭВМ в различных сферах их применения.

Возможности улучшения технико-экономических показателей ЭВМ в значительной степени зависят от элементов, используемых для построения их электронных схем. Поэтому при рассмотрении этапов развития электронных цифровых вычислительных машин каждое поколение обычно в первую очередь характеризуется используемой элементной базой.

Основным активным элементом ЭВМ первого поколения является электронная лампа, остальные компоненты электронной аппаратуры — это обычные резисторы, конденсаторы, трансфор-

маторы. Для построения оперативной памяти ЭВМ уже с середины 50-х годов начали использоваться специально разработанные для этой цели элементы — ферритовые сердечники с прямоугольной петлей гистерезиса. В качестве устройств ввода-вывода сначала использовалась стандартная телеграфная аппаратура (телеграфы, ленточные перфораторы, трансмиттеры, аппаратура счетно-перфорационных машин), а затем специально для ЭВМ были разработаны электромеханические запоминающие устройства на магнитных лентах, барабанах, дисках и быстродействующие печатающие устройства.

К первому поколению машин относятся созданные советскими учеными и инженерами машины БЭСМ-2, «Стрела», М-3, «Минск-1», «Урал-1», «Урал-2», М-20 и др. Они имеют значительные размеры, потребляют большую мощность, имеют сравнительно малое быстродействие, малую емкость оперативной памяти, невысокую надежность работы и недостаточное математическое обеспечение. Несмотря на это первое поколение ЭВМ заложило основы логического построения машин и продемонстрировало грандиозные возможности цифровой вычислительной техники.

На смену ламп в машинах второго поколения (с 1953 г.) пришли транзисторы. В отличие от ламповых ЭВМ транзисторные машины обладают большими быстродействием, емкостью оперативной памяти, надежностью. Существенно уменьшены размеры, масса и потребляемая мощность. Большим достижением явилось применение печатного монтажа. Повысилась надежность электромеханических устройств ввода-вывода, удельный вес которых увеличился. Машины второго поколения обладают большими вычислительными и логическими возможностями.

Особенностью машин второго поколения является их дифференциация по применению. Появились машины для решения научно-технических задач, экономических задач, для управления производственными процессами и различными объектами (управляющие машины).

Наряду с техническим совершенствованием ЭВМ развиваются методы и приемы программирования вычислений, высшей степенью которых является автоматическое программирование, требующее минимальной затраты труда математиков-программистов. Большое развитие и применение получили алгоритмические языки, существенно упрощающие процесс подготовки задач к решению на ЭВМ. С появлением алгоритмических языков резко сократились штаты «чистых» программистов, поскольку составление программ на этих языках стало под силу самим пользователям.

В период развития и совершенствования машин второго поколения, наряду с однопрограммными, появились многопрограммные ЭВМ. В отличие от однопрограммных машин, в которых программы выполняются только поочередно, в многопрограммных ЭВМ возможна одновременная реализация нескольких программ за счет организации параллельной работы основных устройств машины.

С появлением машин второго поколения, особенно многопрограммных машин, значительно расширилась сфера применения электронной вычислительной техники. Машины стали использоваться в качестве управляющего и вычислительного органа в автоматизированных и автоматических системах управления, а также в системах передачи информации.

К ЭВМ второго поколения относятся машины отечественного производства «Минск-2», «Раздан-2», «Раздан-3», М-220, БЭСМ-4, БЭСМ-6, «МИР», «НАИРИ», «Минск-22», «Минск-32», «Урал-14» и др.

Третье поколение ЭВМ (с 1962 г.) характеризуется широким применением интегральных схем, заменивших большинство транзисторов и различных деталей. Интегральная схема представляет собой законченный логический функциональный блок, соответствующий достаточно сложной транзисторной схеме. Благодаря интегральным схемам удалось существенно улучшить технические и эксплуатационные характеристики машин. Этому способствовало также применение многослойного печатного монтажа.

В машинах третьего поколения значительно расширился набор различных электромеханических устройств для ввода и вывода информации. Развитие этих устройств носит эволюционный характер: их характеристики улучшаются гораздо медленнее, чем характеристики электронного оборудования.

Математическое обеспечение машин третьего поколения получило дальнейшее развитие, особенно это касается операционных систем. Развитые операционные системы многопрограммных машин, снабженных периферийными устройствами ввода-вывода с автономными пультами абонентов, обеспечивают управление работой ЭВМ в различных режимах: в режиме пакетной обработки, в режиме деления времени, в режиме запрос—ответ и др.

В машинах третьего поколения, по сравнению с ЭВМ первых двух поколений, существенно расширены возможности по обеспечению непосредственного доступа к ним со стороны абонентов, находящихся на различных, в том числе и значительных (десяtkи и сотни километров) расстояниях. Обеспечение удобства общения абонентов с машиной достигается как за счет развитой сети абонентских пунктов, связанных с ЭВМ информационными каналами связи, так и благодаря соответствующему математическому обеспечению. Например, в режиме деления времени многим абонентам предоставляется возможность одновременного, непосредственного и оперативного доступа к ЭВМ. Вследствие большого различия инерционности человека и машины у каждого из одновременно работающих абонентов создается впечатление, будто ему одному предоставлено машинное время.

При разработке машин третьего поколения применяются различные методы автоматизации проектирования. Основной объем документации, необходимой для монтажа, разрабатывается с помощью ЭВМ.

Четвертое поколение машин начало развиваться с 1970 г. Для них характерно применение больших интегральных схем (БИС), или систем, изготавливаемых на кремниевых пластинах. Высокая степень интеграции способствует увеличению плотности компоновки электронной аппаратуры, повышению ее надежности и быстродействия, снижению стоимости. Это, в свою очередь, оказывает существенное воздействие на логическую структуру (архитектуру) ЭВМ и на ее математическое обеспечение. Более тесной становится связь структуры машины и ее математического обеспечения, особенно операционной системы.

В настоящее время в серийном производстве находятся ЭВМ третьего поколения, которые по сравнению с машинами первого поколения имеют улучшенные характеристики (по быстродействию на 2—3 порядка, по емкости оперативной памяти на 2 порядка, по надежности на 3 порядка). Одновременно на 2—3 порядка уменьшились потребляемая мощность, занимаемая площадь, масса. Доля электронного оборудования в ЭВМ по стоимости снизилась до 40%.

В последние годы весьма отчетливо проявляется тенденция к унификации ЭВМ, к созданию машин, представляющих собой единую систему. Ярким выражением этой тенденции является создание ЕС ЭВМ — Единой системы электронных вычислительных машин.

Единая система электронных вычислительных машин представляет собой семейство (ряд) программно-совместимых машин, построенных на единой элементной базе, на единой конструктивно-технологической основе, с единой структурой, единой системой математического обеспечения, единым унифицированным набором внешних устройств.

Создание ЕС ЭВМ началось по решению шести социалистических стран — Болгарии, Венгрии, ГДР, Польши, СССР и Чехословакии (позже к ним присоединилась Куба). Промышленный выпуск первых моделей ЭВМ Единой системы был начат в 1972 г. По своей сложности, масштабам, целенаправленности и концентрации усилий — это один из крупнейших проектов в истории братского сотрудничества социалистических государств. При создании ЕС ЭВМ были использованы все современные достижения в области электронной вычислительной техники, технологии и конструирования ЭВМ, в области построения систем математического обеспечения. Объединение знаний и производственных мощностей стран—разработчиков ЕС ЭВМ позволило в довольно сжатые сроки решить сложную комплексную научно-техническую проблему.

Все машины Единой системы по своей элементной базе, по принципам функционирования и взаимодействия основных элементов, по составу периферийного оборудования, которое может быть подключено к центральному процессору ЭВМ, представляют собой программно-совместимые машины третьего поколения, предна-

значенные для решения широкого круга научно-технических, экономических и других задач. Важной областью применения ЕС ЭВМ является использование их в АСУ.

В настоящее время в состав ЕС ЭВМ входят семь машин;

- ЕС-1010, разработанная венгерскими специалистами;
- ЕС-1020, созданная в СССР при участии болгарских специалистов;
- ЕС-1021, разработанная ЧССР. Эта машина имеет некоторые конструктивные особенности, обусловленные тем, что она создавалась главным образом для управления технологическими процессами;
- ЕС-1030, разработанная в СССР при участии польских специалистов;
- ЕС-1040, созданная в ГДР;
- ЕС-1050, разработанная в СССР;
- ЕС-1060, разработка которой ведется в СССР.

В основу построения ЕС ЭВМ заложены такие принципы, как принцип мультипрограммирования, совместимость систем программирования на уровне машинных кодов, высокий уровень технической стандартизации и унификации, широкое применение интегральной микроэлектроники, возможность организации многоэтапной работы по созданию и совершенствованию технических и программных средств ЕС ЭВМ.

Одной из особенностей машин Единой системы является возможность работы в режиме разделения времени, при котором несколько абонентов, удаленных от ЭВМ на достаточно большие расстояния, могут общаться с ней непосредственно, оперативно и независимо друг от друга. Взаимодействие удаленных абонентов с ЭВМ осуществляется с помощью периферийной аппаратуры, подсоединенной к каналам через стандартную систему сопряжения. Благодаря стандартному сопряжению центральный процессор может работать с большим набором разнообразных периферийных устройств.

Широкая область использования ЕС ЭВМ обеспечена в значительной степени именно благодаря тому, что разработано много разнообразных периферийных устройств. Здесь речь идет не только о тех устройствах, которые необходимы для использования машин независимо от областей их применения (к ним относятся внешние запоминающие устройства на магнитных дисках, барабанах, лентах, традиционные устройства ввода и вывода информации на перфокарты, перфоленты, алфавитно-цифровые печатающие устройства (АЦПУ), электрифицированные пишущие машинки), но и о тех устройствах, которые расширяют возможность использования машин в самых различных областях. Сюда прежде всего относятся устройства, обеспечивающие телеобработку (дистанционную обработку) информации, например, абонентские пункты, устройства, облегчающие общение человека с машиной, например, различные операторские пульта со средствами наглядного ото-

бражения алфавитно-цифровой и графической информации на электронно-лучевых трубках — дисплеи.

Технические средства, входящие в состав ЕС ЭВМ, в дальнейшем могут существенно обновляться на основе более прогрессивных конструктивных и технологических решений, на основе новых принципов построения логической структуры машин.

Основные направления разработанной и реализуемой в настоящее время программы дальнейшего совершенствования, обновления и развития ЕС ЭВМ заключаются в следующем.

1. Улучшение технико-экономических показателей ЭВМ, оцениваемых главным образом соотношением между производительностью машины и ее стоимостью. Техничко-экономические показатели зависят от таких факторов, как структура машины, ее технологичность, состав математического обеспечения. Улучшение этих показателей предполагается достичь прежде всего за счет использования микросхем с более высоким уровнем интеграции, более широкого применения многослойного печатного монтажа с повышенной плотностью, совершенствования системы охлаждения, обеспечивающей уплотнение конструктивной компоновки.

2. Улучшение системных свойств ЭВМ, обеспечивающих создание с меньшими затратами систем коллективного пользования, систем, работающих в реальном масштабе времени, многопроцессорных и многомашинных вычислительных комплексов.

3. Совершенствование периферийного оборудования, улучшение его технических характеристик, создание новых типов внешних устройств.

В широких масштабах проводятся работы по развитию и расширению системы математического обеспечения ЕС ЭВМ. Проводится стандартизация инструкций и кодов, идет разработка подсистем математического обеспечения, ориентированных на широкие классы применений ЭВМ, — это и АСУ предприятий, объединений, отраслей министерств и сферы торговли, снабжения и другие области. В этой работе принимают деятельное участие все страны — разработчики ЕС ЭВМ. Так, в Болгарии созданы подсистемы «Техничко-экономическое управление для легкой и текстильной промышленности», «Техническая подготовка серийного производства»; в ГДР — «Техническая подготовка производства (определение сводных нормативов)», «Управление материально-техническим снабжением»; в ЧССР — «Управление трудом и заработной платой»; в Польше — «Управление себестоимостью»; в Венгрии — «Бухгалтерский учет»; в Советском Союзе — «Техничко-экономическое управление для машиностроения и приборостроения», «Оперативное управление основным производством» и ряд других подсистем. Активное участие социалистических стран в разработке и совершенствовании машин Единой системы и их математического обеспечения во многом способствует более полному решению задачи освоения и серийного выпуска современных средств электронной вычислительной техники.

ПРИНЦИПЫ ПОСТРОЕНИЯ
И ХАРАКТЕРИСТИКИ ЕС ЭВМ

1.1. ПРИНЦИПЫ ПОСТРОЕНИЯ ЭВМ ЕДИНОЙ СИСТЕМЫ

Единая система ЭВМ представляет собой комплекс типовых стационарных машин общего назначения с достаточно широким диапазоном по производительности, построенных на единой структурной и микроэлектронной базе и имеющих совместимые системы математического обеспечения, а также единую номенклатуру внешних запоминающих устройств и внешних устройств ввода-вывода. Машин Единой системы предназначены для решения широкого круга задач, перечень и характер которых заранее не определены. Это задачи научно-технические, экономические, специальные.

В основу построения ЕС ЭВМ заложены следующие принципы.

1. Принцип мультипрограммирования, при реализации которого на одной и той же ЭВМ одновременно выполняются несколько рабочих программ (программ абонентов, пользователей). При работе в мультипрограммном (или многопрограммном) режиме машина может одновременно выполнять разные команды одной и той же или различных независимых программ, которые хранятся в запоминающем устройстве (ЗУ) этой машины. При этом в каждый данный момент процессор выполняет только какую-то одну команду. Цель применения мультипрограммного режима работы машины — увеличить коэффициент использования ее оборудования за счет организации параллельной работы основных ее устройств. Машины, в которых реализован такой режим работы, называются мультипрограммными, или многопрограммными.

2. Совместимость систем программирования на уровне машинных кодов, что позволяет: использовать имеющиеся заделы программ при переходе к работе на более высокопроизводительных машинах (поскольку в этом есть необходимость); решать одну и ту же задачу за различное время в зависимости от конкретных требований; существенно экономить в силах и средствах при разработке системы стандартных программ математического обеспечения (МО) за счет унификации этих программ для машин различной производительности; создать необходимые предпосылки для технической стандартизации машин. Системы команд ЕС ЭВМ являются достаточно универсальными, в одинаковой степени подходящими для задач различного характера.

Состав команд процессоров включает три основные группы:

а) команды по обработке чисел с фиксированной запятой, имеющих два формата — полуслово (два байта) и слово (четыре

байта), команды для выполнения логических операций, команды управления, команды обмена;

б) команды для выполнения операций над числами с плавающей запятой, имеющих два формата — слово и двойное слово: (восемь байт);

в) команды для выполнения операций над десятичными числами и логических операций над массивами чисел переменной длины. Система адресации в командах обеспечивает формирование прямого адреса для оперативного запоминающего устройства (ОЗУ) емкостью до 16 млн. байт, причем данные могут выбираться различными форматами. Стандартизованы форматы в 2, 4, 8 байт, а также предусмотрена обработка данных с переменной длиной в пределах 1—256 байт.

3. Высокий уровень технической стандартизации, который выражается в единой для всех машин номенклатуре внешних запоминающих устройств (ВЗУ), устройств ввода-вывода (УВВ), абонентских пунктов (АП), в стандартизации связей этих устройств с центральными процессорами, в стандартизации конструктивного решения ряда узлов и блоков различных машин и устройств. Такой уровень стандартизации позволяет, с одной стороны, сэкономить в силах и средствах при разработке и массовом производстве машин Единой системы и, с другой стороны, за счет привлечения широкого круга квалифицированных специалистов увеличить состав и повысить качество технической документации. Стандартизация устройств и узлов сочетается с возможностью вариаций в достаточно широких пределах (обусловленных потребностями абонентов) комплектации машин по количеству ВЗУ, УВВ, АП и по объему оперативной памяти. Стандартизация связей внешних устройств с центральными процессорами позволяет создавать многомашинные вычислительные системы различной конфигурации.

4. Широкое применение интегральной микроэлектроники. Все машины Единой системы относятся к машинам третьего поколения, они строятся на основе интегральных микроэлектронных схем. Это позволяет удешевить аппаратуру, повысить ее надежность, улучшить габаритно-весовые и эксплуатационные характеристики машин, в частности, снизить потребляемую мощность, а также затраты на оборудование помещения, системы охлаждения, системы электросилового питания.

5. Возможность организации многоэтапной работы по созданию машин Единой системы различной производительности и их математического обеспечения. Создание и совершенствование ЕС ЭВМ, оснащенных развитой системой математического обеспечения, — многолетняя задача больших коллективов ученых инженеров, производственников как в нашей стране, так и в странах социалистического содружества, участвующих в решении этой задачи. На первом этапе создаются одномашинные (однопроцессорные) системы и соответствующее математическое обеспечение для работы в однопрограммном режиме (в режиме решения

одной задачи), на втором этапе — однопрограммные системы и математическое обеспечение для работы в многопрограммном режиме, когда на одной машине одновременно выполняются разные команды одной и той же или различных независимых программ и тем самым существенно повышается коэффициент использования основных устройств машины. Только после этого могут создаваться многопроцессорные (многопроцессорные) вычислительные системы для работы в различных режимах, включая режим пакетной обработки, режим разделения времени, режим запрос—ответ, комбинированные режимы.

Реализация указанных принципов построения ЕС ЭВМ связана с необходимостью решения ряда сложных задач. Так, программная совместимость на уровне машинных кодов требует реализации в сравнительно малопроизводительных машинах Единой системы той же развитой и довольно сложной логической структуры, которая характерна для высокопроизводительных машин типа ЕС-1050. Решение задачи программной совместимости стало возможным благодаря применению в малопроизводительных машинах (или в младших моделях машин) микропрограммного управления.

Достижение высокого уровня технической стандартизации потребовало уже на начальной стадии проектирования машин разработки системы нормативно-технической документации (системы стандартов), охватывающей наиболее принципиальные, наиболее общие вопросы по созданию ЕС ЭВМ. Ориентация на широкое применение микроразветренных схем в интегральном исполнении связана с разработкой новых конструкций для компоновки аппаратуры и новых технологических процессов для ее монтажа, в частности, разработки многослойного печатного монтажа.

1.2. ОБОБЩЕННАЯ СХЕМА ЭВМ ЕДИНОЙ СИСТЕМЫ

Как уже отмечалось, все машины Единой системы относятся к разряду многопрограммных ЭВМ. Многопрограммные ЭВМ являются дальнейшим развитием однопрограммных машин, в которых каждая последующая рабочая программа может выполняться только после реализации предыдущей.

Принципы построения многопрограммных и однопрограммных машин в основе своей общие, однако многопрограммные ЭВМ имеют и существенные отличия. Основные из них заключаются в следующем.

1. Машины имеют более сложную структуру. В многопрограммных ЭВМ имеются средства распределения памяти между различными программами, система защиты памяти, система прерываний и приоритетов, датчик времени. В ОЗУ таких машин хранятся несколько одновременно выполняемых программ. Для предотвращения искажения информации при одновременном размещении в ОЗУ нескольких программ предусматриваются меры по

защите программ от взаимного влияния. Реализация этих мер осуществляется системой защиты памяти, которая представляет собой комплекс аппаратурно-программных средств.

2. Большинство многопрограммных ЭВМ строится таким образом, чтобы можно было организовать различные режимы работы, отличающиеся порядком обслуживания программ абонентов и степенью их воздействия на вычислительный процесс: режим пакетной обработки, режим работы с разделением времени, режим запрос—ответ и др. Организация режима работы с разделением времени связана с необходимостью разветвленной сети внешних устройств ввода-вывода.

3. Для многопрограммных машин характерно широкое применение модульных структур. Это прежде всего относится к запоминающим устройствам и к устройствам ввода-вывода, состав которых (количество модулей) определяется нуждами потребителей.

4. Многопрограммные ЭВМ в большей степени приспособлены для построения на их базе многопроцессорных и многомашинных вычислительных систем.

5. Состав математического обеспечения многопрограммных машин значительно шире, чем однопрограммных. Это особенно касается тех машин, в которых предусмотрены различные режимы работы.

Кроме отмеченного, в многопрограммных ЭВМ более широко реализована возможность работы с машинными словами переменной длины, а также возможность выполнения операций с числами, представленными не только в двоичном, но и двоично десятичном коде.

Структура машин единой системы может быть представлена в виде обобщенной схемы (рис. 1.1). Основными устройствами машины являются процессор, оперативное запоминающее устройство (ОЗУ), включающее основную оперативную память (ООП) и большую оперативную память (БОП), мультиплексный канал (МК), селекторные каналы (СК), число которых для различных моделей ЭВМ различно, устройства ввода-вывода (УВВ), внешние запоминающие устройства (ВЗУ) и устройства управления внешними запоминающими устройствами (УУ ВЗУ).

Каждое устройство представляет собой функционально независимую по преобразованию информации часть машины. В конструктивном отношении не все устройства представлены в виде автономной, самостоятельной части машины. Устройства включают в себя следующие функциональные части: элементы, узлы, блоки.

Элемент — простейшая функциональная часть ЭВМ, обеспечивающая выполнение элементарной операции над одной или несколькими цифрами алфавита. К числу элементарных операций относятся: запоминание цифры алфавита, реализация одной из элементарных логических операций, формирование сигнала (кодového, управляющего, синхронизирующего) по форме, длитель-

ности, амплитуде. Примеры элементов: триггер, ферритовый сердечник с обмотками, логические элементы И, ИЛИ, НЕ, И—НЕ, ИЛИ—НЕ, И—ИЛИ—НЕ и др., усилитель, формирователь и т. д.

Узел — функциональная часть машины, составленная из элементов и обеспечивающая выполнение операции над одним словом. К узлам относятся регистры, счетчики, коммутаторы и т. д.

Блок — функциональная часть машины, составленная из элементов и узлов, обеспечивающая выполнение операций над словами или управление выполнением этих операций. Примеры блоков: блок сумматора, блок регистров, блок местного управления операциями и др. Иногда блоки выделяются как самостоятельные конструктивные части.

Процессор является главной частью ЭВМ. В нем сосредоточены средства выполнения арифметических и логических операций, средства управления выполнением заданной последовательности команд, средства обращения к оперативной памяти и организации начала обмена информацией между оперативной памятью и внешними устройствами. Координация работы устройств, блоков и узлов машины осуществляется с помощью синхронизирующих и управляющих сигналов, вырабатываемых в процессоре. Синхронизирующие сигналы обеспечивают совмещение срабатывания элементов ЭВМ во времени, а управляющие сигналы задают необходимую последовательность срабатывания этих элементов.

Все машины единой системы имеют общую логическую структуру процессора, однако ее аппаратная реализация различна в разных моделях системы. Этим и объясняется неодинаковая производительность процессоров ЕС ЭВМ.

Основная оперативная память служит для приема, хранения и выдачи информации (операндов и команд), участвующих в ближайшем ряде операций. Емкость ООП может достигать 1024К байт* (для модели ЕС-1050).

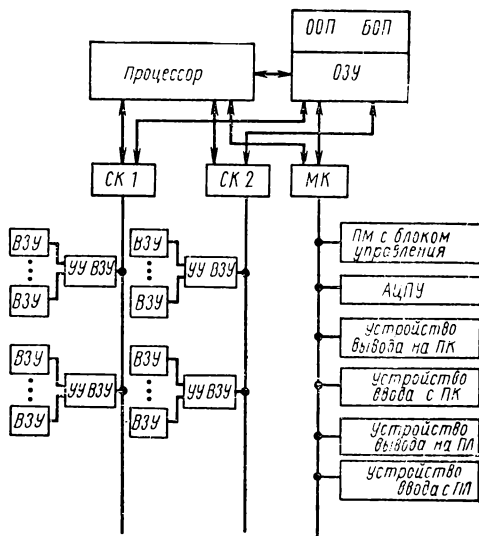


Рис. 1.1. Обобщенная схема ЭВМ единой системы

* Здесь и далее при определении емкости устройств памяти $K = 1024$.

Организация ООП выполнена таким образом, что возможна адресация хранящихся в ней данных к каждому байту. За одно обращение к памяти в разных моделях выбирается одновременно различное количество байтов, представляющее собой ширину выборки.

Быстродействие оперативной памяти характеризуется циклом обращения и временем выборки. Цикл обращения — это полное время, затрачиваемое на обращение к адресуемой ячейке памяти. За цикл обращения осуществляется дешифрация адреса, выборка содержимого ячейки, регенерация считанной информации или запись новой информации в ячейку. Время выборки — это время, затрачиваемое на дешифрацию адреса и выборку содержимого ячейки.

Необходимо отметить, что в младших моделях ЭВМ Единой системы основная оперативная память конструктивно объединена с процессором, что дает основание считать ее составной частью процессора.

Большая оперативная память является резервом для ООП. Она имеется только в старших моделях ЭВМ Единой системы. Емкость БОП может достигать по 16 мегабайт.

Мультиплексный канал представляет собой логически самостоятельный агрегат, осуществляющий организацию взаимосвязи между основной оперативной памятью машины и группой сравнительно медленнодействующих устройств ввода-вывода. Мультиплексный канал работает по методу уплотнения передач путем разделения во времени действий по обслуживанию нескольких одновременно работающих устройств ввода-вывода.

Мультиплексный канал разделяется на подканалы, число которых определяется числом внешних устройств (устройств ввода-вывода, периферийных абонентских пунктов), подключаемых к каналу. Подканал образуют средства, необходимые для выполнения операции ввода-вывода, связанной с одним внешним устройством.

Устройство ввода служит для восприятия вводимой информации (исходных данных и программ), преобразования ее в электрические кодовые сигналы и передачи в линию связи к мультиплексному каналу. В качестве устройств ввода применяются устройства ввода с перфокарт, устройства ввода с перфолент, пишущие машинки с блоками управления, графические устройства ввода (дисплеи, графопостроители).

Устройство вывода служит для преобразования выводимого из машины числового материала, представленного электрическими сигналами, в систему пробивок на перфокартах или перфолентах или для печатания результатов решения задач на бумажных лентах. В качестве устройств вывода применяются устройства вывода на перфокарты, устройства вывода на перфоленты, алфавитно-цифровые печатающие устройства, графические устройства вывода.

Селекторный канал — это логически самостоятельный агрегат, осуществляющий организацию взаимосвязи между основной оперативной памятью машины и группой внешних запоминающих устройств в режиме предпочтения, или в монопольном режиме. В селекторном канале исключена возможность одновременной работы нескольких ВЗУ, все средства канала монополизируются в течение некоторого времени одним ВЗУ.

В младших моделях ЭВМ Единой системы мультиплексный и селекторный каналы конструктивно объединены с процессором.

В качестве накопителей информации ВЗУ применяются накопители на магнитных лентах (НМЛ), на магнитных дисках (НМД), на магнитных барабанах (НМБ). Устройства управления (УУ) для нескольких однотипных накопителей являются общими.

Кроме указанных устройств, связанных между собой системой магистралей, в состав ЕС ЭВМ входят *устройства подготовки данных*, которые не имеют электрической связи с машиной. Эти устройства служат для подготовки исходных данных и программ к вводу в машину. С их помощью исходные данные и программы, записанные на стандартных бланках, переносятся на перфокарты или перфоленты, с которых в дальнейшем информация считывается в устройствах ввода и поступает в ЗУ. В состав устройств подготовки данных входят клавишные устройства, контрольно-считывающие устройства, перфораторы.

1.3. КОМПЛЕКТАЦИЯ МОДЕЛЕЙ ЭВМ ЕДИНОЙ СИСТЕМЫ

В состав ЕС ЭВМ входят различные комплекты технических средств (аппаратурная часть машин) и система математического обеспечения. Комплект технических средств включает одну или несколько моделей ЭВМ, набор устройств ввода-вывода, присоединенных к моделям через стандартный интерфейс ввода-вывода, набор внешних запоминающих устройств, связанных через селекторные каналы с центральным процессором (ЦП) и ОЗУ, набор автономных внешних устройств для подготовки исходных данных и программ к вводу в машину, вспомогательное оборудование (ЗИП, кондиционеры, системные кабели и т. п.) Стандартный интерфейс ввода-вывода представляет собой устройство, включающее логические и аппаратурные средства, которые обеспечивают стандартную систему линий связи и обмен сигналами, имеющими унифицированные электрические параметры. Интерфейс имеет средства для обеспечения мультиплексной работы и позволяет обслуживать до 256 одновременно работающих устройств.

Система математического обеспечения состоит из программ операционных систем, контролирующих и диагностических тест-программ технического обслуживания и набора прикладных программ. Модель ЭВМ вместе с мультиплексным и селекторными

каналами, набор устройств ввода-вывода, набор ВЗУ, система математического обеспечения образуют однопроцессорную вычислительную систему.

Характерной особенностью ЭВМ единой системы является модульность структуры их технических и программных средств, что обеспечивает большую приспособляемость ЭВМ к нуждам пользователей. Организации-пользователю представляется возможность выбора приемлемого комплекта технических средств, исходя из своих конкретных требований к производительности процессора, емкости ООП, составу и емкости ВЗУ, составу УВВ. Такая возможность обеспечивается тем, что ООП моделей ЕС-1020, ЕС-1030 и ЕС-1050 комплектуется поблочно, а подключение ВЗУ и УВВ к мультиплексному и селекторному каналам осуществляется унифицированным образом — через стандартный интерфейс ввода-вывода. Это позволяет включать в состав ЭВМ любое устройство номенклатурного перечня. Более того, по мере развития ЕС ЭВМ допускается включение в ее состав более совершенных устройств без изменения структуры системы в целом. Необходимо только, чтобы эти устройства, поскольку они являются нестандартными, имели автономные блоки управления, обеспечивающие выход на стандартный интерфейс ввода-вывода. Специфика управления этими устройствами должна быть также отражена в операционной системе ЭВМ.

В системе математического обеспечения ЕС ЭВМ предусматриваются три основных типа операционных систем (ОС):

а) ОС-1 — самый низкий уровень ОС с основной управляющей программой с последовательным планированием обработки заданий (рабочих программ) и основными средствами управления данными;

б) ОС-2—ОС с управляющей программой, обеспечивающей реализацию многопрограммного режима работы с фиксированным числом задач (выполняемых рабочих программ), с приоритетным планированием обработки заданий и расширенными средствами управления данными;

в) ОС-3 — самый высокий уровень ОС с управляющей программой, обеспечивающей реализацию многопрограммного режима работы с переменным числом задач, с приоритетным планированием обработки заданий, развитыми средствами ввода-вывода и возможностью вывода графической информации.

Математическое обеспечение любой модели ЭВМ Единой системы комплектуется на базе одной из этих операционных систем, набора прикладных программ, функционирующих под управлением ОС, и комплекта программ технического обслуживания. Пользователь имеет возможность включить в состав ОС свои программные модули, если это диктуется необходимостью. Такая необходимость возникает, например, при создании математического обеспечения новых, более совершенных устройств ввода-вывода.

Кроме ОС, на первом этапе разработки математического обеспечения младших моделей ЭВМ Единой системы предусматривается применение дисковой операционной системы (ДОС). Как и большинство операционных систем, ДОС предназначена для автоматизации процесса создания программ, сокращения промежутка времени между постановкой задачи для решения и получением результатов и для увеличения объема работы, выполняемой машиной в единицу времени.

Очевидно, что для каждого модуля ЕС ЭВМ существуют свои требования к планировке, подводке питания, системе вентиляции и охлаждения в зависимости от конкретных условий эксплуатации.

Номенклатура технических средств и возможности математического обеспечения позволяют создавать различные структуры (конфигурации) однопроцессорных вычислительных систем с учетом требований пользователей и особенностей эксплуатации. Операционные системы позволяют подсоединять к моделям ЭВМ до 748 различных УВВ и ВЗУ и 255 устройств управления ими.

Основными критериями при комплектации моделей ЭВМ Единой системы являются конкретные потребности пользователя: требования по производительности процессора, по емкости ЗУ, по составу математического обеспечения. Конфигурация вычислительной системы существенно зависит от того, для решения каких задач она предназначена (научно-технических, экономических, информационно-поисковых и т. п.). Номенклатурный перечень внешних запоминающих устройств и устройств ввода-вывода ЕС ЭВМ приведен в приложении (табл. 1).

Необходимо иметь в виду, что для функционирования операционных систем ДОС и ОС необходим определенный минимум технических средств, обязательный при комплектации модели. Минимально необходимый состав технических средств приведен в приложении (табл. 2). Кроме устройств, перечисленных в этой таблице, для работы с информационными массивами требуются по крайней мере два накопителя на магнитных лентах с соответствующим устройством управления. Структуры систем, построенных на базе моделей ЕС-1020 и ЕС-1030 (или ЕС-1050), с минимальным составом технических средств приведены на рис. 1.2 и 1.3. Такие структуры неудобны для использования, так как не реализуются полностью возможности операционных систем.

Вычислительная система, построенная на базе модели ЕС-1020 и предназначенная для решения научно-технических задач, может иметь структуру, приведенную на рис. 1.4. Для решения экономических задач может быть рекомендована система, схема которой показана на рис. 1.5. Наиболее полный состав устройств имеют информационно-поисковые системы (рис. 1.6), в которых может быть до 176 выносных абонентских пультов.

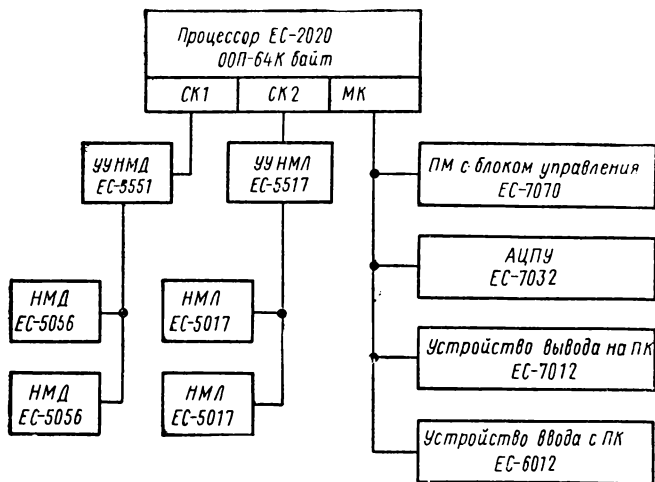


Рис. 1.2. Схема модели ЕС-1020 с минимальным составом технических средств

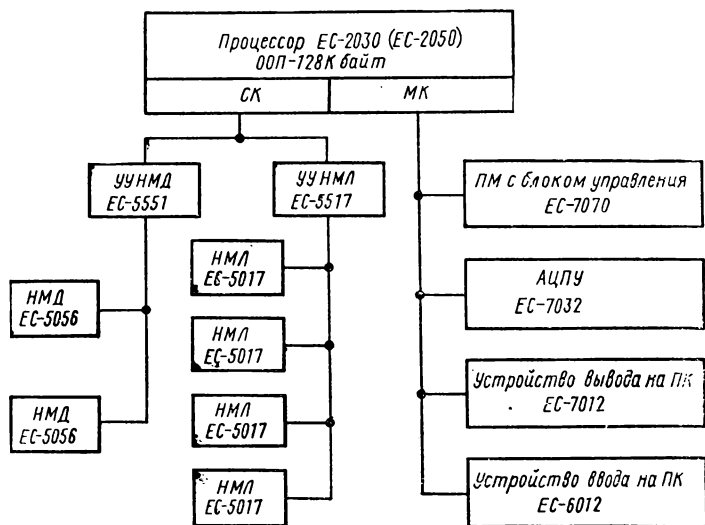


Рис. 1.3. Схема модели ЕС-1030 (ЕС-1050) с минимальным составом технических средств

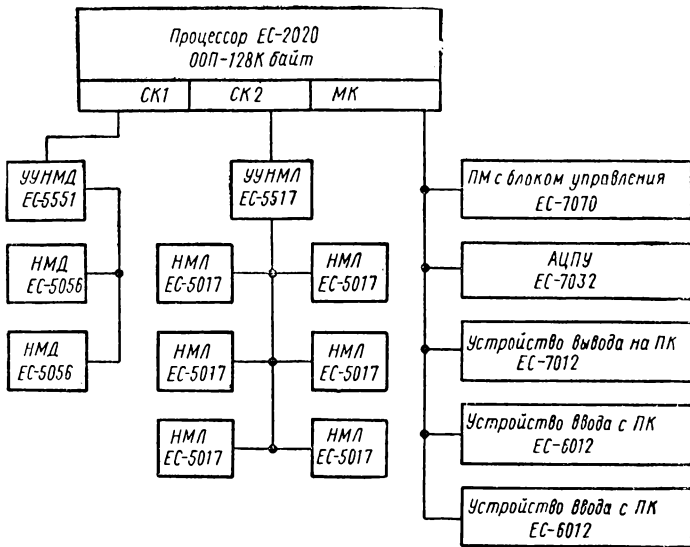


Рис. 1.4. Возможная структура модели EC-1020 для решения научно-технических задач

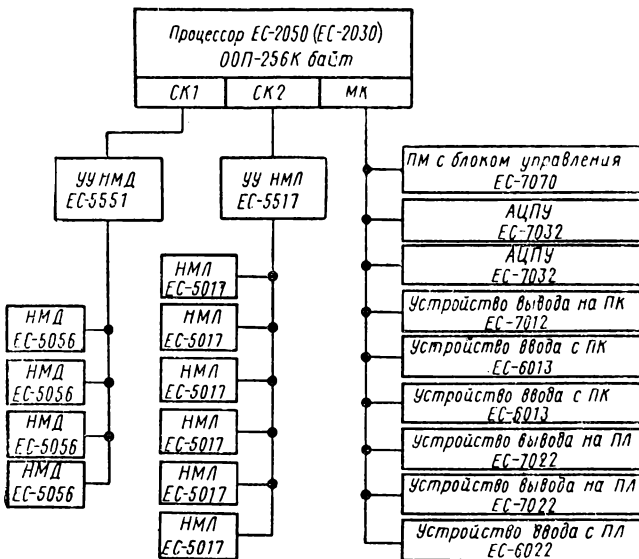


Рис. 1.5. Возможная структура модели EC-1050 (EC-1030) для решения экономических задач

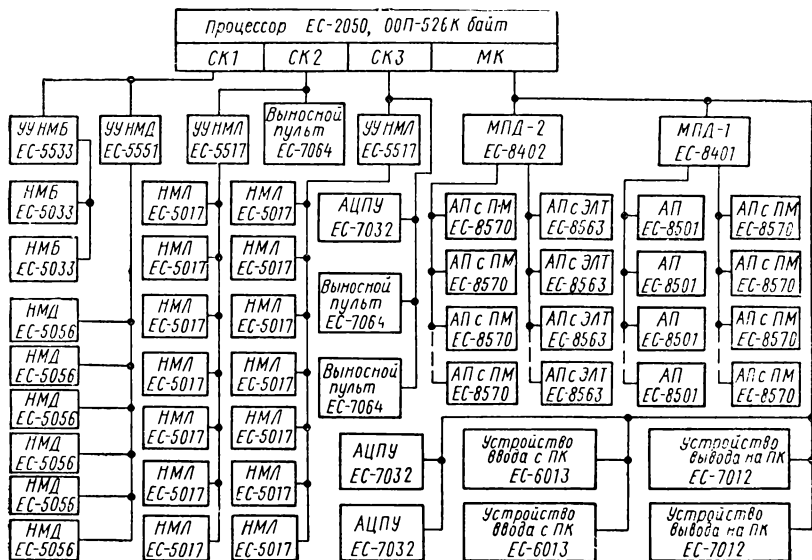


Рис. 1.6. Информационно-поисковая система на базе модели ЕС-1050

1.4. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ЭВМ ЕДИНОЙ СИСТЕМЫ

Характеристики машин Единой системы [11, 12, 37] приведены в приложении, где указаны основные характеристики моделей (табл. 3), УВВ (табл. 4), абонентских пунктов (табл. 5) и мультиплексоров передачи данных (табл. 6). Кроме данных, содержащихся в этих таблицах, приведем некоторые дополнительные сведения.

Модель ЕС-1010 предназначена для обработки данных механизированного делопроизводства, для управления технологическими процессами и решения сравнительно простых научно-технических задач. Она может быть также использована в качестве спутника старших моделей для предварительной обработки информации (для подготовки и преобразования данных).

В процессоре модели реализовано микропрограммное управление. Память для хранения микропрограмм построена на полупроводниковых интегральных схемах, емкость ее равна 512—1536 слов, длина слова 16 бит, время выборки 60 мкс. В арифметическом устройстве процессора операции выполняются над полусловами. Сверхоперативная регистровая память процессора имеет емкость 64×2 байт, цикл обращения 60 нс.

В типовом составе модели имеются один мультиплексный и один селекторный каналы, для технической реализации которых

требуется около 1000 модулей интегральных схем. Основная оперативная память блочной структуры, емкость одного блока равна 8К байт, цикл обращения 0,8—1,0 мкс, время выборки 0,4—0,5 мкс. Число блоков ООП может достигать 8, в типовом составе модели установлены два блока. В качестве ВЗУ используются два накопителя на магнитных лентах и один накопитель на сменных магнитных дисках. Устройство управления накопителями расположено в типовой стойке. Устройства ввода-вывода представлены фотосчитывающим устройством с перфолент, ленточным перфоратором и двумя электрифицированными пишущими машинками, одна из которых используется как пульт оператора. Кроме того, имеется выносной пульт для вывода алфавитно-цифровой информации на электронно-лучевую трубку (ЭЛТ).

Все электронное оборудование модели, в том числе и ООП, размещается в одной типовой стойке. Увеличение емкости оперативной памяти производится путем установки дополнительных блоков в электронную стойку. Модель ЕС-1010 в типовом комплекте потребляет мощность около 2,0 кВА.

Модель ЕС-1010 характеризуется последовательной обработкой байтов и реализацией перечня команд, принятых для ЕС ЭВМ, с помощью программной интерпретации. Она поставляется с операционной системой ОС-10.

Модель ЕС-1021 принадлежит к числу малых моделей Единой системы. Она предназначена для решения экономических и административно-управленческих задач в низшем звене АСУ. Кроме того, машина может применяться в качестве вспомогательной ЭВМ больших вычислительных комплексов.

Типовой комплект машины включает процессор с оперативной памятью и каналами, накопителя на сменных магнитных дисках, АЦПУ, устройства ввода с перфокарт и перфолент и пультовую пишущую машинку. Машина поставляется с малой операционной системой (МОС).

Оперативная память на ферритовых сердечниках имеет емкость 16К байт (в расширенном варианте 32 или 62К байт), цикл обращения 2 мкс, время выборки 1 мкс, ширину выборки 1 байт. Защита памяти не предусмотрена.

Устройство управления процессора построено по принципу микропрограммирования. Микропрограммы хранятся в постоянной памяти емкостью 3072 слова, длина слова 72 разряда, цикл обращения 250 нс.

В составе модели имеются один мультиплексный канал и два селекторных канала. Мультиплексный канал разделяется на 16 подканалов, он дает возможность подключить до 136 внешних устройств. Скорость обмена информацией в мультиплексном режиме равна 35 кбайт/с, в монопольном режиме — 220 кбайт/с. К каждому селекторному каналу через 10 устройств управления может быть подключено до 160 внешних устройств. Максимальная скорость обмена селекторного канала равна 250 кбайт/с.

Типовой комплект модели ЕС-1021 размещается на площади 50 м², потребляемая мощность 13 кВА. С другими моделями ЕС ЭВМ машина ЕС-1021 совместима на уровне алгоритмических языков и программ в АССЕМБЛЕРЕ и РПГ.

Модель ЕС-1020 (Р-20) предназначена для решения широкого круга научно-технических, информационно-логических и экономических задач в научных и проектных организациях, промышленных и торговых предприятиях, государственных учреждениях и ведомствах. Минимальный состав технических средств машины, (минимальная стандартная конфигурация) указывался выше (см. рис. 1.2).

Модель поставляется с ДОС, а при наличии расширенной емкости основной оперативной памяти возможно применение ОС.

Процессор модели ЕС-1020 характеризуется широким применением микропрограммного управления, наличием запоминающих регистров и однобайтового арифметического устройства. Следовательно, как и в процессоре модели ЕС-1010, здесь также имеет место последовательная обработка байтов. Микропрограммная логика процессора является основным средством управления выполнением всех инструкций, операций с пульта, процедур обмена данными с устройствами ввода-вывода. Структура микрокоманды обеспечивает задание и выполнение действий над двумя операндами в АУ с одновременным обращением к памяти и формированием адреса следующей микрокоманды. Количество различных команд процессора равно 142.

Время выполнения основных операций, мкс:

Сложение — вычитание с фиксированной запятой	20—30
Сложение — вычитание с плавающей запятой	50—70
Сложение — вычитание при двойных словах	50—110
Умножение с фиксированной запятой	350
Умножение с плавающей запятой	480
Умножение при двойных словах	1200
Деление с фиксированной запятой	400
Деление с плавающей запятой	400
Короткие операции	20—30

Процессор вместе с мультиплексным и двумя селекторными каналами содержит 8000 модулей интегральных схем. Максимальная скорость обмена в каждом из селекторных каналов равна 200 кбайт/с. Производительность мультиплексного канала характеризуется такими данными: скорость обмена в мультиплексном режиме 10—16 кбайт/с (число подканалов равно от 48 до 112 в зависимости от емкости ООП), в монопольном режиме скорость обмена в одном подканале составляет 100 кбайт/с. Конструктивно процессор с каналами размещается в двух типовых стойках.

Основная оперативная память может быть увеличена до 256К байт. Наименьшей адресуемой единицей информации является байт. Память имеет защиту по записи и считыванию.

В типовом (не минимальном) составе модели используются 4 ВЗУ на МЛ, подключаемых к селекторному каналу через УУ НМЛ, и один накопитель на сменных магнитных дисках, также подключаемый к каналу через отдельное устройство управления. Устройства ввода представлены фотосчитывающим устройством с перфолент и устройством ввода с перфокарт. В составе выходных устройств имеются устройство вывода данных на перфокарты, устройство вывода на перфоленты, а также алфавитно-цифровое печатающее устройство. Операторский пульт модели ЕС-1020 оборудован пишущим механизмом «Консул-260» со скоростью печати до 10 зн/с.

Для размещения оборудования комплекта модели ЕС-1020 необходимо помещение площадью 50 м², потребляемая мощность 21 кВА.

Модель ЕС-1030 (Р-30) предназначена для решения достаточно сложных научно-технических задач и обработки больших массивов информации, хранимой в ВЗУ. Она может использоваться в средних вычислительных центрах научно-исследовательских предприятий и вузов страны, а также в средних и старших звеньях АСУ.

В процессоре модели ЕС-1030 осуществляется параллельно-последовательная обработка четырехбайтовых двоичных слов и широко используется микропрограммное управление. Микропрограммы хранятся в постоянной памяти трансформаторного типа на П-образных магнитных сердечниках.

Основная оперативная память машины в типовом комплекте имеет емкость 256К байт с возможностью доведения ее до 512К байт. Цикл обращения к памяти — 1,25 мкс, время выборки — 0,75 мкс, ширина выборки — 4 байта. Защита памяти организована как по записи, так и по считыванию.

В типовом составе модели имеется один мультиплексный канал, включающий память емкостью 1024 32-битных слова, и два селекторных канала. Число подканалов мультиплексного канала — 256, скорость обмена в мультиплексном режиме — 40 кбайт/с, в монопольном режиме — 200—300 кбайт/с. Селекторный канал, работающий только в монопольном режиме, имеет скорость обмена 600—900 кбайт/с. Используются восемь накопителей на магнитной ленте, подключаемых к селекторному каналу через устройство управления, смонтированное в типовой стойке. Кроме того, имеются два накопителя на сменных магнитных дисках, которые подключаются к каналу через свое устройство управления. В качестве устройства ввода-вывода используются устройства ввода с перфокарт и перфолент, выходные ленточные и карточные перфораторы, АЦПУ. В состав модели включен также одиночный выносной пульт для ввода и вывода алфавитно-цифровой и графической информации на электронно-лучевую трубку.

Модель ЕС-1030 в типовом составе занимает площадь порядка 110 м² и потребляет мощность 25 кВА.

Модель ЕС-1040 относится к числу высокопроизводительных моделей Единой системы. Она предназначена для решения сложных научно-технических и экономических задач и может использоваться в достаточно крупных вычислительных центрах, в системах сбора и обработки данных, в средних и высших звеньях АСУ.

Типовой комплект машины включает процессор с оперативной памятью и каналами ввода-вывода, ВЗУ на сменных магнитных дисках и магнитных лентах, набор перфокарточного и перфоленточного оборудования ввода-вывода, печатающее устройство и пультовую пишущую машинку. В зависимости от области применения модель ЕС-1040 поставляется с ДОС или с ОС.

В состав процессора ЕС-2040 входят вычислительное устройство, устройство управления, оперативная память, каналы, а также вспомогательные устройства и блоки. Вычислительное устройство включает основной арифметический блок для обработки операндов и байтовый арифметический блок для обработки команд.

Емкость оперативной памяти, имеющей модульную структуру, может быть 256, 512 и 1024К байт. Цикл обращения к памяти — 1,35 мкс, время выборки — 0,45 мкс, выборка побайтовая. Имеется защита памяти по записи. Конструктивно оперативная память оформлена автономно, но функционально входит в состав процессора.

В комплекте модели имеется один мультиплексный и до шести селекторных каналов. Каждый из каналов конструктивно оформляется в виде самостоятельного блока. К каждому селекторному каналу подключается до десяти устройств управления внешними устройствами. Обмен информацией через селекторные каналы осуществляется с различной скоростью: для первого канала скорость обмена равна 1300 кбайт/с, для второго и третьего — 500 кбайт/с, для четвертого, пятого и шестого — 300 кбайт/с. В мультиплексном канале содержится 128 подканалов при емкости основной оперативной памяти, равной 256К байт, или 256 подканалов при емкости 512 и 1024К байт. Скорость обмена в мультиплексном режиме равна 20—25 кбайт/с, в монопольном режиме 180—720 кбайт/с (максимальная скорость достигается в случае, когда ни один из селекторных каналов не работает). К мультиплексному каналу подключается также до десяти устройств управления.

Устройство управления процессора имеет микропрограммную структуру. Постоянная память микропрограмм характеризуется следующими данными: емкость — 3072 слов, длина слова — 130 бит, цикл работы — 450 нс.

Размещение оборудования комплекта модели ЕС-1040 осуществляется на площади в 150 м², потребляемая мощность 65 кВА.

Модель ЕС-1050 (Р-50) предназначена для решения широкого круга сложных научно-технических, экономических, информационных задач в крупных центрах обработки информации. Она

может использоваться также в старших звеньях АСУ. Это машина наиболее высокой производительности по сравнению с другими моделями Единой системы. Для нее характерно выделение мультиплексного и селекторного каналов в конструктивно-самостоятельные устройства с полностью автономным управлением, наличие в большом объеме внутрисистемных и внешних средств комплексов, более широкое введение аппаратных средств контроля и диагностики. Машина поставляется с операционной системой ОС.

В процессоре модели осуществляется выполнение всех арифметических и логических операций по обработке данных, организуется обращение к основной оперативной памяти, иницируются процедуры ввода-вывода через каналы. Он характеризуется параллельной обработкой восьмибайтовых слов и аппаратной реализацией системы управления. Высокая скорость выполнения операций достигается за счет совмещения во времени этапа выполнения очередной команды с проведением подготовительных этапов для последующих команд. В модели предусмотрена защита памяти по ключам как по записи, так и по считыванию. Для хранения ключей защиты в процессоре имеется специальная память емкостью 512 байт и циклом обращения 320 нс. Конструктивно процессор размещается в двух типовых электронных стойках и содержит около 35 000 модулей интегральных схем.

Оперативная память модели ЕС-1050 имеет достаточно большую емкость: от 256 до 1205К байт (от 1 до 4 блоков емкостью 256К

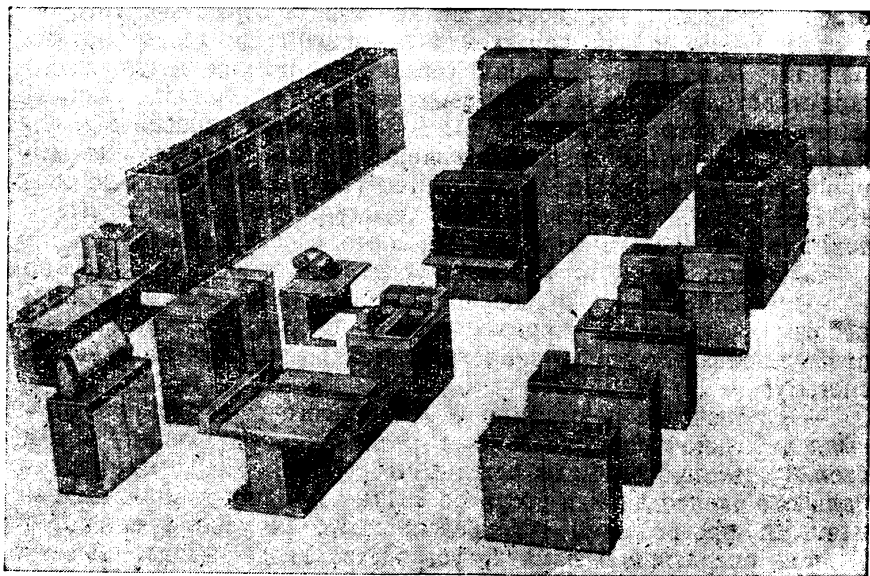


Рис. 1.7. Внешний вид модели ЕС-1050

байт). Полный цикл обращения — 1,25 мкс, время выборки 0,8 мкс, ширина выборки — 8 байт. Предусмотрена возможность одновременного обращения по двум смежным адресам ОПП.

Типовой комплект модели включает один мультиплексный канал и два или три селекторных.

В мультиплексном канале имеется 192 неразделимых подканала, образующих общий мультиплексный подканал, и 4 разделенных селекторных подканала. Селекторные подканалы работают только в монопольном режиме, скорость обмена одного подканала до 180 кбайт/с. Мультиплексный подканал работает как в мультиплексном, так и в монопольном режиме. Суммарная пропускная способность этого подканала составляет 110 кбайт/с. Суммарная пропускная способность всего мультиплексного канала не превышает 670 кбайт/с. Скорость обмена одного селекторного канала равна 1,3 Мбайт/с. К каждому селекторному каналу может подключаться до 8 устройств управления.

Основное электронное оборудование модели ЕС-1050 размещается в шести типовых стойках. В типовом составе модель занимает площадь 230 м² и потребляет мощность 70 кВА.

Внешний вид модели ЕС-1050 приведен на рис. 1.7.

Модели ЕС-1020; ЕС-1030; ЕС-1040 и ЕС-1050 совместимы на уровне машинного языка.

1.5. МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЕС ЭВМ

Математическим обеспечением или системой математического обеспечения (СМО) ЭВМ называется комплекс программных средств, предназначенных для повышения эффективности использования машины, облегчения ее эксплуатации и снижения трудоемкости подготовительной работы при решении задач на машине. Система математического обеспечения ЕС ЭВМ является основой для получения математического обеспечения однопроцессорных и многопроцессорных вычислительных систем (ВС), построенных на базе ЕС ЭВМ и представляющих собой совокупность соответствующих технических средств и математического обеспечения.

Основные функции математического обеспечения ЕС ЭВМ:

— управление прохождением заданий (независимых единиц работы, задаваемых программистом, оператором, абонентом, вообще пользователем машинного времени) через вычислительную систему;

— обеспечение повышения эффективности функционирования вычислительной системы за счет реализации различных, в том числе мультипрограммных режимов ее работы, обеспечение рационального распределения ресурсов ВС между одновременно выполняемыми программами (распределяемыми ресурсами являются области оперативной памяти, устройства ввода-вывода, области памяти ВЗУ, машинное время процессора, программы БСП, системные программные модули, как подпрограммы);

- обеспечение взаимодействия вычислительной системы и оператора в формах, удобных для оператора;
- предоставление стандартных программ для типичных приложений;
- обеспечение контроля надежности функционирования технических средств с помощью системы контролирующих и диагностических тестов;
- обеспечение сокращения времени на подготовку задач к решению на ЭВМ с помощью различных систем автоматизации программирования, являющихся составной частью СМО Единой системы.

В структуре системы математического обеспечения ЕС ЭВМ, представленной на рис. 1.8, можно выделить две группы — общее МО и специальное. Общее МО позволяет повысить эффективность использования машины, облегчает ее техническую эксплуатацию, упрощает работу программистов по подготовке задач к решению. Такое МО решает задачу применения ЭВМ как некоторой универсальной системы обработки информации. Специальное МО представляет собой комплекс программ, которые добавляются к общему МО и обеспечивают функционирование ЭВМ как некоторой специализированной системы обработки информации.

Общее МО состоит из четырех частей [28]: операционной системы, набора пакетов прикладных программ, комплекса программ технического обслуживания и системы документации на СМО Единой системы.

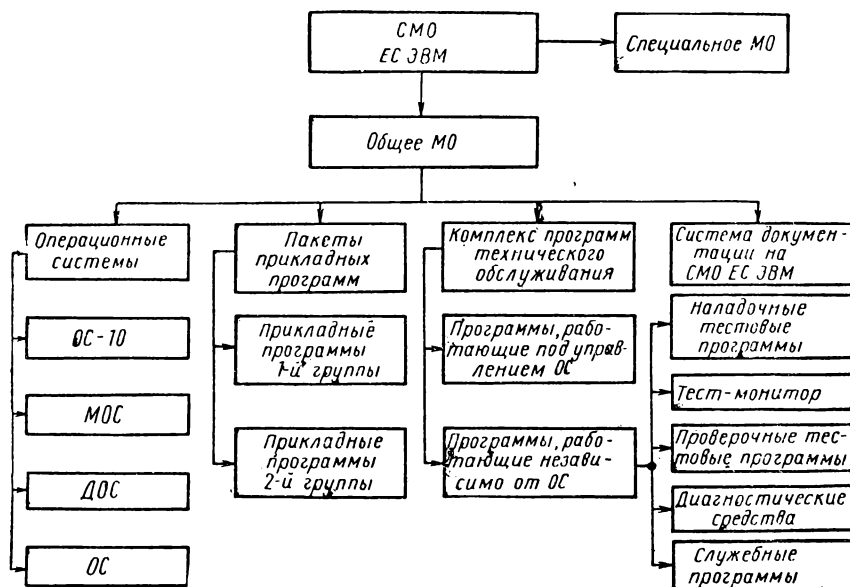


Рис. 1.8. Структура системы математического обеспечения ЕС ЭВМ

В ЕС ЭВМ операционная система объединяет средства организации и управления процессом функционирования машины в различных режимах.

Перечень и объем операций, выполняемых операционной системой, различен для различных типов ОС. В настоящее время разработаны четыре типа операционных систем:

- ОС-10 для модели ЕС-1010;
- МОС — для модели ЕС-1021;
- дисковая операционная система ДОС для всех других моделей ЭВМ Единой системы;
- основная операционная система ОС также для всех других моделей ЭВМ Единой системы.

Более подробно структура и функции ДОС и ОС рассмотрены в гл. 14. Эти типы операционных систем в наибольшей степени отвечают требованиям, главные из которых состоят в следующем:

— ОС должна обладать свойством адаптируемости, структурной гибкости, способностью изменяться и пополняться новыми программными компонентами. Это обуславливается разнообразием условий и областей применения вычислительных систем: состав активных абонентов и характер решаемых ими задач могут изменяться в достаточно широком диапазоне, могут изменяться также и периферийные технические средства (внешние устройства ввода-вывода). Обеспечение адаптируемости, расширения функций и возможностей операционной системы с целью использования ее с новой аппаратурой, для новых областей применения и с новыми программными компонентами по мере их появления достигается с наименьшими затратами при использовании принципа модульного программирования. Модульное программирование состоит в разбиении всего процесса обработки информации на ряд элементарных процессов или модулей. В применении к операционной системе модульное программирование сводится к разбиению ее программ на модули, каждый из которых специализирован на управлении определенной совокупности операций, выполняемых в вычислительной системе. При этом один и тот же модуль может использоваться для нескольких программ. Модули могут формироваться и отлаживаться независимо друг от друга. Благодаря модульной структуре программы операционной системы легко поддаются изменению либо заменой модулей, либо подключением новых, либо перестановкой модулей, определяющих процесс обработки информации;

— ОС должна обеспечивать предоставление абонентам максимума удобств в сочетании с наиболее эффективным использованием ЭВМ;

— ОС должна способствовать сведению к минимуму простоев оборудования (главным образом процессора), связанного с вмешательством оператора или программиста в процесс обработки информации машиной;

— потери времени на работу самой операционной системы должны быть минимальными.

Затраты машинного времени на работу ОС существенно зависят от соотношения между программной и аппаратурной частями этой системы (операционная система, строго говоря, представляет собой программно-аппаратурный комплекс средств организации и управления процессом функционирования ЭВМ). В общем случае чем больше удельный вес аппаратурной части, тем сложнее ЭВМ, но тем меньше затрачивается времени на работу ОС, и наоборот. В аппаратурную часть ОС входят переключатели режимов работы машины и ОС, схемы выработки сигналов прерываний, регистры прерываний, схемы управления прерыванием, счетчики времени для выработки периодических временных сигналов и т. д.

Программная часть операционной системы представляет собой иерархический набор управляющих и обслуживающих программ, верхний уровень которого определяет стратегию, а последующие — тактику прохождения задач через ЭВМ. Стратегия управления определяется на основании анализа характеристик задач и имеющихся ресурсов. Тактика в основном определяется текущим состоянием ЭВМ, временем нахождения в машине задач, подлежащих решению, количеством этих задач, временем на их обслуживание и т. д.

Среди управляющих и обслуживающих программ, входящих в состав операционных систем ЕС ЭВМ, можно выделить следующие наиболее важные функциональные группы:

— программы начального пуска, управляющие операцией начального ввода информации в основную память ЭВМ и настройку ОС на заданный режим работы;

— программы диспетчирования вычислений, управления выполнением потока заданий с организацией прерываний для передачи управления и мультипрограммной работы в различных режимах, программы планирования мультипрограммной работы;

— программы управления вводом-выводом информации, обменом информацией между ЭВМ и абонентами, между основной памятью и внешними запоминающими устройствами и устройствами ввода-вывода;

— программы управления процессом контроля работоспособности ЭВМ.

В состав операционных систем ДОС и ОС включены средства, обеспечивающие снижение трудоемкости подготовительной работы при решении задач на ЭВМ. Эти средства составляют систему автоматизации программирования. Системой автоматизации программирования (САП) называется часть математического обеспечения, обеспечивающая всестороннюю автоматизацию процесса получения рабочих программ абонентов. Развитые САП включают алгоритмические языки (проблемно ориентированные, машинно-ориентированные), трансляторы (программы перевода рабочих программ с алгоритмических языков на машинный язык), интер-

претирующие и компилирующие системы, библиотеки стандартных подпрограмм, программы сервиса и т. д. Объем и разнообразие средств, представленных в САП, в конечном счете определяют «квалификацию» машины, удобство использования ее абонентами и программистами.

Прикладные программы предназначены для решения типовых научно-технических, инженерно-экономических и специальных задач.

В системе математического обеспечения выделяются две группы прикладных программ: прикладные программы 1-й группы, обеспечивающие расширение возможностей операционной системы для специальных комплексов технических средств и способов применения; прикладные программы 2-й группы, составляющие комплексы программ для решения типовых задач.

К прикладным программам 1-й группы относятся:

- программные средства обеспечения телеобработки;
- программные средства обеспечения машинной графики;
- средства для построения обобщенных систем обработки данных;
- средства для обеспечения специфических внешних устройств;
- пакеты программ для обслуживания многомашинных комплексов;
- средства для обмена данными с машинами-спутниками;
- средства для обеспечения преемственности программ машин, не входящих в ЕС ЭВМ;

— системы программирования на базе проблемно-ориентированных языков, предназначенные для решения некоторых классов задач (задач моделирования, обработки символов матриц и др.).

Телеобработка включает широкий круг применений. В системе телеобработки осуществляется обмен информацией между абонентскими пунктами и центральным вычислительным органом (центральной системой) по телефонным и телеграфным каналам связи. Средства телеобработки обеспечивают обработку данных на расстоянии, обработку заданий на расстоянии, решение задач в оперативном режиме (в режиме разделения времени).

Обработка данных на расстоянии предусматривает три вида работ:

коммутацию сообщений, обработку запросов и сбор данных.

Коммутация сообщений это такой вид телеобработки, при котором сообщения, полученные центральным вычислительным органом от некоторого абонентского пункта, передаются одному или нескольким другим абонентским пунктам.

Обработка запросов представляет собой вид телеобработки, при котором в соответствии с запросами от абонентских пунктов центральная система посылает на эти пункты данные, содержащиеся в одном или нескольких основных файлах (массивах) данных. Обновление основных файлов производится в оперативном режиме в соответствии с данными, поступающими от абонентских

пунктов. Такой вид телеобработки может применяться для управления различными процессами в реальном масштабе времени. Характерным примером системы обработки запросов является система резервирования авиабилетов.

Сбор данных — вид телеобработки, при котором в центральной системе происходит накопление и последующая обработка данных, поступающих от абонентских пунктов. Примером системы сбора данных может служить система учета готовой продукции некоторого предприятия: данные о количестве выработанной продукции каждого наименования поступают с рабочих мест через абонентские пункты в течение смены и накапливаются в центральной системе. В конце смены производится обработка этих данных и подведение итогов.

Обработка заданий на расстоянии — вид телеобработки, при котором задания на обработку данных вводятся в центральную систему с одного или нескольких абонентских пунктов и выполняются системой в режиме пакетной обработки. Результаты обработки либо отправляются на соответствующие абонентские пункты, или засылаются в библиотеку на хранение. Абонент, находящийся на абонентском пункте, имеет доступ ко всем средствам центральной системы.

Решение задач в оперативном режиме, называемом также режимом разделения времени, представляет собой такой вид телеобработки, при котором ряд абонентов имеет возможность через свои абонентские пункты непосредственно, одновременно и оперативно использовать центральную вычислительную систему для решения своих задач. Работа в режиме разделений времени часто осуществляется в форме диалога между абонентом и программой, которая выполняется центральной системой в интересах абонента. Программа может запрашивать у абонента необходимую информацию или отвечать на его запросы и информировать об ошибках. При этом допускается использование языков программирования высокого уровня, автокодов и специального языка команд.

Программные средства обеспечения машинной графики или иначе средства графического программирования предназначены для организации и обеспечения взаимодействия абонента с вычислительной системой через графические и алфавитно-цифровые устройства графического отображения (дисплей). Эти средства упрощают поиск, обработку и воспроизведение информации в графической форме. Они включают в себя расширенный набор макрокоманд, управляющие программы, обрабатывающие программы, средства манипулирования данными, программы анализа прерываний от ввода-вывода, диагностические программы и программы восстановления при ошибках.

Средства обеспечения машинной графики включают также следующие системы:

— систему оперативной работы на алфавитно-цифровых дисплеях, представляющую собой набор программ, которые при ис-

пользовании алфавитно-цифровых дисплеев для оперативной связи с ЭВМ обеспечивают выполнение ряда общих функций, таких как обработка запросов от дисплеев, разделение времени между дисплеями и т. д.;

— систему графического анализа трехмерных данных, предназначенную для создания, анализа и преобразования трехмерных данных с использованием графического дисплея;

— систему машинной графики на проблемно-ориентированных языках, содержащую средства для описания изображений и задания формата данных, выводимых на графические дисплеи.

Удобным средством графического программирования является световое перо, с помощью которого оператор имеет возможность указывать выполняемой программе определенные точки, линии или знаки на воспроизводимом изображении. Совместно с клавиатурой световое перо используется для переупорядочения информации на экране, устранения ненужной информации, соединения линиями уже высвеченных на экране точек.

Под обобщенными системами обработки данных понимаются системы, предназначенные для обработки банков данных, т. е. больших массивов взаимосвязанных данных, не содержащих избыточной информации. Такие системы должны обеспечивать возможность создания, изменения и расширения автоматизированных систем обработки информации, включая АСУ. Создание программных средств для построения обобщенных систем обработки данных — задача весьма сложная и трудоемкая.

В состав прикладных программ 1-й группы входит математическое обеспечение графопостроителей и устройств ввода-вывода с перфолент.

Математическое обеспечение графопостроителей представляет собой набор программ, обеспечивающих формирование и вывод изображений на графопостроители под управлением операционной системы ОС.

Основные компоненты этого математического обеспечения следующие:

— совокупность программ, реализующих графический метод доступа. С помощью этих программ осуществляется управление данными в основной памяти, управление выводом информации на графопостроители, обработка ошибок, возникающих при работе графопостроителей;

— сервисные программы графопостроителей (набор макрокоманд), обеспечивающие простое программирование для графопостроителей на языке АССЕМБЛЕРА;

— транслятор, предназначенный для перевода графических программ, написанных на языке приказов графопостроителей, в последовательность команд;

— проблемно-ориентированные программы — это набор стандартных программ, обеспечивающих вывод на графопостроитель ряда часто используемых изображений;

— пакет графических программ графопостроителей — это совокупность стандартных программ, обеспечивающих формирование изображений и вывод на графопостроитель при программировании на одном из языков ФОРТРАН, КОБОЛ или ПЛ/1;

— обобщенные системы вывода изображений на графопостроители представляют собой пакет взаимосвязанных программ, обеспечивающих вывод на графопостроители сложных графических изображений, описанных на одном из проблемно-ориентированных языков, и обобщенных наборов данных.

Математическое обеспечение устройств ввода-вывода с перфо-лент — это совокупность программ, входящих в состав управляющей программы ОС и обеспечивающих обработку последовательных наборов данных с записями переменной или неопределенной длины. С помощью этих программ осуществляется преобразование данных, считанных с перфоленты и представленных в некотором коде, во внутренний код машины. При выводе данных на перфоленту производится обратное преобразование кодов. Предусматривается ввод информации с перфоленты и вывод на перфоленту без программного преобразования кодов.

На базе моделей ЭВМ Единой системы могут строиться много-машинные вычислительные комплексы с различными уровнями комплексирования. Например, могут строиться многомашинные комплексы с вспомогательным процессором или комплексы, составленные из двух однотипных машин. Математическое обеспечение таких комплексов включает пакеты программ, обеспечивающих работу комплексов в различных режимах. В частности, должны быть программы для распределения работ между машинами с целью эффективной загрузки всего комплекса.

Машины типа ЕС-1010 могут использоваться в качестве машин-спутников, связанных с машинами Единой системы более высокой производительности через каналы связи. Например, на базе машины ЕС-1010 создан абонентский пункт с переменным набором устройств ввода-вывода. Для машин-спутников разрабатываются специальные наборы программ, обеспечивающие использование ресурсов высокопроизводительных машин в интересах машин-спутников.

Преимственность программ машин, не входящих в ЕС ЭВМ, достигается за счет создания программных средств, обеспечивающих выполнение таких функций, как преобразование данных на картах и магнитных лентах в форматы данных на магнитных лентах ЕС ЭВМ, перевод программ, составленных на автокодах, на язык АССЕМБЛЕРА, трансляция программ, составленных на языке АЛМО, и др.

В состав прикладных программ 2-й группы входят:

- пакеты программ для научно-технических расчетов;
- система математического программирования;
- пакеты программ для моделирования;
- системы для обработки документов и текстовой информации;

- пакеты программ для планирования и управления;
- обобщенные информационные системы.

Пакеты прикладных программ для научно-технических расчетов составлены из стандартных программ, реализующих различные численные методы анализа, теории вероятности, математической статистики, теории графов.

Система математического обеспечения включает пакеты прикладных программ, предназначенных для решения широкого круга задач линейного и нелинейного программирования, сетевого анализа, теории игр, теории управления, теории запасов, теории массового обслуживания, задач дискретной оптимизации и т. д. Необходимость решения этих задач возникает в самых различных отраслях, таких как планирование производства, перевозок и снабжения, распределение ресурсов, управление кадрами и др.

Пакеты прикладных программ для моделирования используются при исследовании сложных систем. Например, имеются пакеты программ для оценки качества функционирования сложных дискретных систем, для моделирования процессов, описываемых системой линейных и нелинейных дифференциальных уравнений, и др.

Система обработки документов и текстовой информации представляет собой набор программ, предназначенных для организации информационно-поисковой системы. По запросу абонентов система организует поиск необходимых документов, причем на печать поступают либо библиографические данные о документе, либо его поисковый образ, либо полный текст.

Пакеты программ для планирования и управления предназначены для решения задач планирования и управления, таких как планирование материальных и людских ресурсов, управление различными процессами и т. д.

Обобщенные информационные системы предназначены для сбора, накопления и обработки различных данных и информационных запросов. Основное их приложение — это обработка массивов экономической и научно-технической информации и организация больших информационно-поисковых систем.

Комплекс программ технического обслуживания ЕС ЭВМ предназначен для профилактического контроля машин во время планового обслуживания, для контроля и локализации неисправностей в процессе эксплуатации ЭВМ (внеплановое обслуживание), для оперативной проверки работоспособности внешних устройств в мультипрограммном режиме, а также для наладки серийных образцов машин. В комплекс входят тестовые программы, которые разделяются на две группы: программы, работающие под управлением операционной системы, и программы, работающие независимо от операционной системы.

Программы, входящие в состав операционной системы и работающие под ее управлением, обеспечивают сокращение времени восстановления работоспособности ЭВМ после возникновения

сбоя в аппаратуре. К ним относятся программы сбора и редактирования информации о сбоях процессора и селекторных и мультиплексных каналов, программы восстановления ЭВМ после сбоя, программы оперативной проверки устройств.

Состав и назначение тестовых программ, работающих независимо от операционной системы.

1. Система наладочных тестовых программ предназначена для проверки и обнаружения неисправностей отдельных устройств и модели в целом при их наладке. Для каждого устройства имеется свой наладочный тест и выполняется он отдельно при проведении автономных проверок устройств. Часть автономных тестов объединяется в комплекс наладочных тестовых программ, обеспечивающих комплексную проверку модели. Состав системы наладочных тестов для каждого типа модели свой.

2. Управляющая программа тест-монитор — это самостоятельная программа, предназначенная для управления выполнением проверочных и диагностических тестов. Она обеспечивает автоматический (без участия оператора) прогон тестов модели, связь с оператором (через входные и выходные сообщения), загрузку заданного теста в основную память, назначение тестам соответствующих устройств, перемещение тестов в основной памяти.

3. Система проверочных тестовых программ предназначена для контроля правильности функционирования всех устройств модели. Проверочные тесты работают в основном под управлением тест-монитора и только небольшая их часть выполняется автономно. Большинство проверочных тестовых программ являются унифицированными, независимыми от типа модели, что важно для обеспечения совместимости моделей.

4. Диагностические средства предназначены для локализации места неисправности аппаратуры. Они включают диагностические программы и диагностические процедуры, которые реализуют оборудование, встроенное в процессор. С помощью диагностических средств реализуются такие методы диагностики, как метод диагностических таблиц, метод диагностических характеристик, метод списков подозреваемых элементов.

3. Служебные программы предназначены для генерации и обслуживания тестовых программ на магнитных носителях. Основные функции служебных программ: размещение тестовых программ на магнитных дисках и лентах, копирование тестов, включение новых тестов, исключение и исправление тестов.

Наконец, четвертая часть общего математического обеспечения ЕС ЭВМ — это система документации на СМО Единой системы. Она предназначена для обеспечения эффективного применения, изучения и обслуживания программных средств ЕС ЭВМ.

сбоя в аппаратуре. К ним относятся программы сбора и редактирования информации о сбоях процессора и селекторных и мультиплексных каналов, программы восстановления ЭВМ после сбоя, программы оперативной проверки устройств.

Состав и назначение тестовых программ, работающих независимо от операционной системы.

1. Система наладочных тестовых программ предназначена для проверки и обнаружения неисправностей отдельных устройств и модели в целом при их наладке. Для каждого устройства имеется свой наладочный тест и выполняется он отдельно при проведении автономных проверок устройств. Часть автономных тестов объединяется в комплекс наладочных тестовых программ, обеспечивающих комплексную проверку модели. Состав системы наладочных тестов для каждого типа модели свой.

2. Управляющая программа тест-монитор — это самостоятельная программа, предназначенная для управления выполнением проверочных и диагностических тестов. Она обеспечивает автоматический (без участия оператора) прогон тестов модели, связь с оператором (через входные и выходные сообщения), загрузку заданного теста в основную память, назначение тестам соответствующих устройств, перемещение тестов в основной памяти.

3. Система проверочных тестовых программ предназначена для контроля правильности функционирования всех устройств модели. Проверочные тесты работают в основном под управлением тест-монитора и только небольшая их часть выполняется автономно. Большинство проверочных тестовых программ являются унифицированными, независимыми от типа модели, что важно для обеспечения совместимости моделей.

4. Диагностические средства предназначены для локализации места неисправности аппаратуры. Они включают диагностические программы и диагностические процедуры, которые реализуют оборудование, встроенное в процессор. С помощью диагностических средств реализуются такие методы диагностики, как метод диагностических таблиц, метод диагностических характеристик, метод списков подозреваемых элементов.

3. Служебные программы предназначены для генерации и обслуживания тестовых программ на магнитных носителях. Основные функции служебных программ: размещение тестовых программ на магнитных дисках и лентах, копирование тестов, включение новых тестов, исключение и исправление тестов.

Наконец, четвертая часть общего математического обеспечения ЕС ЭВМ — это система документации на СМО Единой системы. Она предназначена для обеспечения эффективного применения, изучения и обслуживания программных средств ЕС ЭВМ.

представленные вначале в двоично-десятичной форме, по специальным командам или подпрограммам автоматически преобразуются в двоичные. При выдаче результатов осуществляется обратное преобразование в десятичную систему.

Перевод числа, в общем случае любого смешанного,

$$A_N = \pm (a_{m-1} N^{m-1} + a_{m-2} N^{m-2} + \dots + a_1 N^1 + a_0 N^0 + a_{-1} N^{-1} + \dots + a_{-l} N^{-l}), \quad (2.1)$$

из системы счисления с основанием N_1 в систему с основанием N_2 может быть выполнен по универсальному алгоритму, который нетрудно определить, приведя целую и дробную части выражения (2.1) к видам:

$$A_N^u = \pm ((\dots((a_{m-1} N + a_{m-2}) N + a_{m-3}) N + \dots) N + a_0) \quad (2.2)$$

и

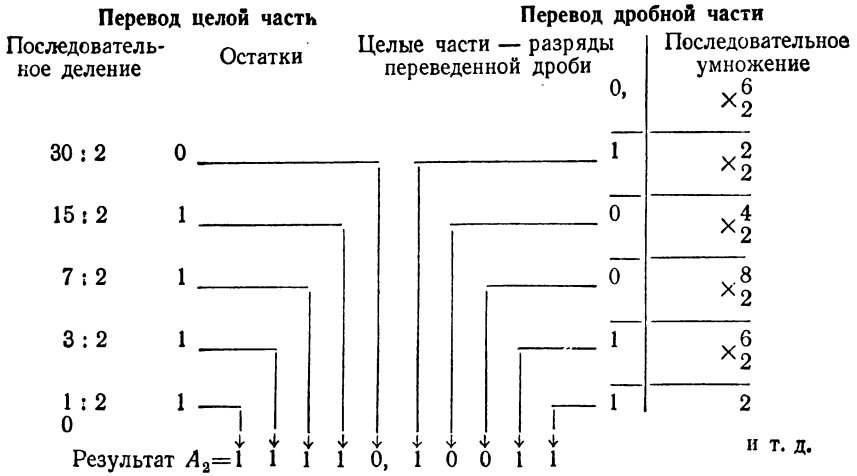
$$A_N^{др} = \pm N^{-1} (a_{-1} + N^{-1} (a_{-2} + \dots + N^{-1} (a_{-(l-1)} + a_{-l} N^{-1}) \dots)). \quad (2.3)$$

Согласно этому алгоритму перевод смешанного числа состоит из вычислительных процессов двух видов: 1) последовательного деления целой части и образующихся целых частных на основание новой системы счисления и 2) последовательного умножения дробной части и дробных частей получающихся произведений на то же новое основание, записанное, как и в первом случае, цифрами исходной системы счисления.

При переводе целой части *получающиеся* в результате процесса последовательного деления *остатки представляют цифры* a_0, a_1, \dots *целой части числа в новой системе счисления, записанные цифрами исходной системы счисления*. Последний остаток является старшей цифрой переведенного числа.

При переводе дробной части числа целые части, получающиеся при каждом умножении, не участвуют в последующих умножениях. Эти *целые части представляют цифры дробной части исходного числа в новой системе счисления, изображенные цифрами старшей системы*. Значение первой целой части является первой цифрой после запятой переведенного числа.

Пример 2.1. Требуется перевести десятичное число $A_{10} = 30,6$ в двоичную систему счисления ($A_2 = ?$). Согласно данному выше правилу, переводим отдельно целую $A_{10}^u = 30$ и дробную $A_{10}^{др} = 0,6$ части числа.

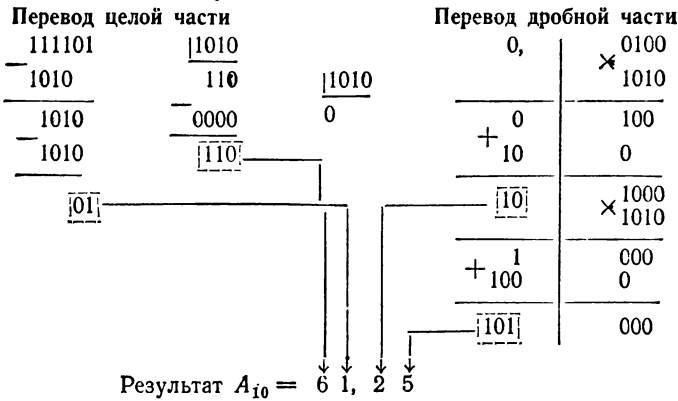


Если при переводе дробной части получается периодическая дробь, как в рассматриваемом примере, то производят округление, руководствуясь заданной точностью вычислений. В данном случае искомая дробь A_2 определена с точностью до пятого знака после запятой.

Аналогично рассмотренному выше можно осуществлять перевод чисел и из двоичной системы счисления в десятичную.

Пример 2.2. Двоичное число $A_2 = 111'01,01$ перевести в десятичную систему счисления.

Основание десятичной системы счисления в двоичной записи имеет вид 1010. Поэтому при переводе целую часть $A_2^ц$ будем последовательно делить, а дробную часть $A_2^д$ последовательно умножать на число 1010.



Десятичные эквиваленты разрядов искомого числа находим, пользуясь примером на стр. 38.

При переводах чисел по рассмотренному алгоритму необходимо выполнять последовательные преобразования, пользуясь таблицами сложения и умножения исходной системы счисления. Для машинного перевода это не имеет значения, а для перевода вручную при переходе к системе с большим основанием такой процесс оказывается неудобным. Поэтому, пользуясь формулами

(2.2) и (2.3), применяют и другой алгоритм перевода. Например, для целых чисел сначала находится произведение старшего разряда числа на основание исходной системы счисления, к которому прибавляется следующая цифра переводимого числа. Затем полученная сумма также умножается на основание, к полученному произведению прибавляется следующая цифра и т. д. до последнего цикла, в котором осуществляется только прибавление цифры младшего разряда без последующего умножения. При этом действия выполняются в новой системе счисления.

Пример 2.3. Перевести двоичное число $A_2^C = 111101$ в десятичную систему счисления.

$$\begin{array}{r}
 \times 111101 \\
 \times 2 \\
 \hline
 + 2 \leftarrow 1 \\
 \hline
 \times 3 \\
 \times 2 \\
 \hline
 + 6 \leftarrow 1 \\
 \hline
 \times 7 \\
 \times 2 \\
 \hline
 + 14 \leftarrow 1 \\
 \hline
 \times 15 \\
 \times 2 \\
 \hline
 + 30 \leftarrow 0 \\
 \hline
 \times 30 \\
 \times 2 \\
 \hline
 + 60 \leftarrow 1 \\
 \hline
 \end{array}$$

Результат $A_{10} = 61$

Аналогичная последовательность действий может быть применена и для перевода дробных частей, только для умножения здесь придется использовать $N_2^{-1} = 0,5$.

Пример 2.4. Перевести двоичное число $A_2^D = 0,101$ в десятичную систему счисления.

$$\begin{array}{r}
 0, 101 \\
 \times 0,5 \\
 \hline
 + 0,5 \\
 \hline
 \rightarrow 0 \\
 \times 0,5 \\
 \hline
 + 0,25 \\
 \hline
 \rightarrow 1 \\
 \times 1,25 \\
 \times 0,5 \\
 \hline
 \end{array}$$

Результат $A_{10} = 0,625$

Машинный перевод в десятичную систему счисления также обычно осуществляется аналогичным образом по циклической схеме, состоящей из последовательности умножений на основание десятичной системы в соответствующей степени и прибавлений значений очередных разрядов переводимого числа.

Кроме двоичной и десятичной систем счисления в ЭВМ Единой системы широко используется третья система — *шестнадцатиричная*. В единой системе она имеет такое же значение, какое имела восьмеричная система в ранее выпускавшихся цифровых машинах и, кроме того, используется для непосредственной организации некоторых видов вычислений.

Шестнадцатиричная система счисления имеет основание $N = 16$. Значение каждого разряда шестнадцатиричного числа может изменяться от 0 до 15, для изображения которых применяют десять арабских цифр: от 0 до 9, и шесть прописных букв латинского алфавита: A, B, C, D, E и F:

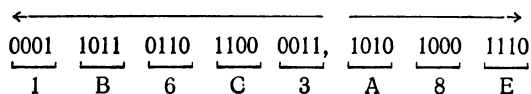
Шестнадцатиричная цифра	0	1	2	3	4	5	6	7
Десятичное значение . . .	0	1	2	3	4	5	6	7
Двоичный код	0000	0001	0010	0011	0100	0101	0110	0111

Шестнадцатиричная цифра	8	9	A	B	C	D	E	F
Десятичное значение . . .	8	9	10	11	12	13	14	15
Двоичный код	1000	1001	1010	1011	1100	1101	1110	1111

Так как основание шестнадцатиричной системы $16 = 2^4$, и, следовательно, показатель степени 4 относится к показателю степени основания двоичной системы 2^1 , как 4 : 1, то переход от шестнадцатиричной записи чисел к двоичной осуществляется очень просто. Каждая шестнадцатиричная цифра выражается четырьмя двоичными (тетрадами). Обратный перевод также прост. Для этого двоичное число разбивается влево и вправо от запятой на тетрады, и каждая из них заменяется шестнадцатиричной цифрой. Если правая и левая тетрады будут неполными, то двоичное число необходимо соответственно справа и слева дополнить нулями, например, как в следующем примере.

Пример 2.5. Требуется преобразовать двоичное число

$$A_2 = 11011011011000011, 10101000111.$$



Таким образом, $A_{16} = 1B6C3, A8E$.

Также проста в применении и восьмеричная система. Однако достоинство шестнадцатиричной системы счисления состоит в том,

что она позволяет делать более компактные записи двоичных чисел, что очень важно, поскольку диапазоны разрядности чисел, команд и специальных двоичных слов, которыми оперируют ЭВМ Единой системы, стали значительно шире. Например, адреса основной оперативной памяти имеют 24 двоичных разряда; специальные слова, отражающие состояния вычислительного процесса, имеют 64 двоичных разряда. Поэтому возможность делать более короткие записи больших двоичных чисел явилась одной из первых причин перехода к шестнадцатиричной системе счисления.

Кроме того, в ЕС ЭВМ в качестве основной единицы информации используется восьмиразрядный двоичный код — байт (от английского слова byte — слог). Переменные разрядности чисел и команд установлены кратными байту. Двоичный код байта при записи удобно представляется двухразрядным шестнадцатиричным числом.

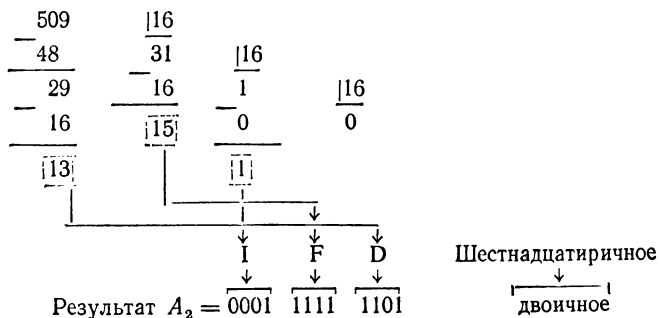
В шестнадцатиричной записи наглядно изображаются и десятичные числа.

Применение шестнадцатиричной системы оказалось удобным для реализации в машинах ускоренных сдвигов на 4 и 8 разрядов, что способствует ускорению выполнения арифметических и логических операций, содержащих сдвиги.

Как будет показано в п. 2.4, применение шестнадцатиричной системы счисления для выражения чисел в полулогарифмической форме с плавающей запятой, кроме ускорения выполнения операций, расширяет допустимый диапазон представления нормализованных чисел.

Применение шестнадцатиричной системы счисления позволяет быстрее переводить десятичные числа в двоичные и обратно. При этом вначале соответственно либо десятичное число переводится в шестнадцатиричное, либо шестнадцатиричным заменяется двоичное.

Пример 2.6. Перевести, пользуясь шестнадцатиричной системой счисления, десятичное число $A_{10} = 509$ в двоичную систему счисления.



Пример 2.7. Пользуясь шестнадцатиричной системой, перевести 24-разрядное число

$A_2 = 100010011100011111110011$ в десятичное.

1000	1001	1100	0111	1111	0011	← двоичное
9	C	7	F	3		← шестнадцатиричное
× 8						
16						
48						
+ 8						
128						
9 ←						
× 137						
16						
822						
+ 137						
2192						
12 ←						
× 2204						
16						
13224						
+ 2204						
35264						
7 ←						
× 35271						
16						
211626						
+ 35271						
564336						
15 ←						
× 564351						
16						
3386106						
+ 564351						
9029616						
3 ←						
9029619 ←						← десятичное

Для того чтобы оценить эффективность применения шестнадцатиричной системы счисления, можно попытаться то же самое 24-разрядное число перевести непосредственно из двоичной системы в десятичную.

2.2. КОДИРОВАНИЕ АЛФАВИТОВ ВХОДНЫХ ЯЗЫКОВ

Алфавит входного языка составляет совокупность элементарных символов, с помощью которых на языке записываются программы решаемых задач. Эти программы в машинах автоматически транслируются в программы на машинном языке, на котором они и выполняются.

Стандартный алфавит ЕС ЭВМ содержит:

а) графические символы, к которым относятся латинские прописные и строчные буквы ($A, B, C, \dots, Z, a, b, c, \dots, z$), арабские цифры (0, 1, 2, . . . , 9), ограничители (знаки арифметических и логических операций, разделители, скобки);

б) управляющие символы, являющиеся указателями специальных сигналов управления аппаратурой передачи данных, устройствами ввода-вывода, разделения информации на части (блоки) в соответствии с ее логическим содержанием.

Всего в стандартном алфавите имеется 94 графических и 33 управляющих символов. Этот набор соответствует латинской части международного 7-разрядного кода для передачи данных, утвержденного Международной организацией по стандартизации и Международным консультативным комитетом по телефонии и телеграфии. Он составил основу ГОСТ 13052—74 «Машины вычислительные и аппаратура передачи данных. Коды 7-битные для обмена информацией».

В составе символов стандартного кода, кроме латинских букв, имеются прописные и строчные буквы русского алфавита. На базе этого набора символов для ЕС ЭВМ разработан так называемый полный алфавит.

Полный алфавит по сути дела представляет собой расширение стандартного добавлением ряда дополнительных управляющих символов, букв русского алфавита и нескольких графических знаков. В настоящее время в полном алфавите имеется 190 символов.

Для того чтобы иметь возможность представлять в цифровых машинах информацию, выраженную символами алфавита, необходимо сами символы закодировать определенными наборами двоичных цифр. Существует несколько систем кодирования.

В ЕС ЭВМ в основном применяются три системы, одна из которых основана на представлении символов 7-разрядными кодами, а две — 8-разрядными.

Способ кодирования 7-разрядными кодами для обмена информацией (КОИ-7) иллюстрируется табл. 2.1. Он применяется в системе телеобработки данных для передачи информации по линиям связи, а также при нанесении ее на перфоленку. Таблица состоит из двух частей, различающихся столбцами: 4—7 и 12—15. Столбцы 8—11 повторяют содержимое столбцов 0—3. Левая часть (столбцы 0—7) в ЕС ЭВМ именуется первым регистром, правая — вторым (национальным) регистром*. В аппаратуре связи эти регистры обычно называются соответственно латинским и русским.

Согласно таблице каждому символу соответствует 7-разрядный двоичный код $\mathcal{A}_7\mathcal{A}_6\mathcal{A}_5\mathcal{A}_4\mathcal{A}_3\mathcal{A}_2\mathcal{A}_1$. Позиции в таблице обычно обозначают дробью, числитель которой указывает номер столбца, а знаменатель — номер строки. Например, букве A в первом

* Для того чтобы можно было кодировать большее количество служебных символов, таблица КОИ-7 имеет еще третий регистр.

Таблица 2.1

Таблица кодирования символов 7-разрядными кодами, используемыми для обмена информацией в ЕС ЭВМ (КОИ-7)

							Первый регистр							Второй регистр									
Э ₇	Э ₆	Э ₅	Э ₄	Э ₃	Э ₂	Э ₁	Э ₇	Э ₆	Э ₅	Э ₄	Э ₃	Э ₂	Э ₁	Э ₇	Э ₆	Э ₅	Э ₄	Э ₃	Э ₂	Э ₁			
							0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
			0	0	0	0	0	ПУС	АР1	0	@	P	\	p									
			0	0	0	1	1	НЗ	СУ1	!	1	A	Q	a	q					ю	п	Ю	П
			0	0	1	0	2	НТ	СУ2	»	2	B	R	b	r					б	р	Б	Р
			0	0	1	1	3	КТ	СУЗ	#	3	C	S	c	s					ц	с	Ц	С
			0	1	0	0	4	КП	СТП	д	4	D	T	d	t					д	т	Д	Т
			0	1	0	1	5	КТМ	НЕТ	%	5	E	U	e	u					е	у	Е	У
			0	1	1	0	6	ДА	СИН	&	6	F	V	f	v					ф	ж	Ф	Ж
			0	1	1	1	7	ЗВ	КБ	'	7	G	W	g	w					з	в	З	В
			1	0	0	0	8	ВШ	АН	(8	H	X	h	x					х	ь	Х	Ь
			1	0	0	1	9	ГТ	КН)	9	I	Y	i	y					и	ы	И	Ы
			1	0	1	0	10	ПС	ЗМ	*	:	J	Z	j	z					й	з	Й	З
			1	0	1	1	11	ВТ	АР2	+	;	K	[k	{					к	ш	К	Ш
			1	1	0	0	12	ПФ	РФ	,	<	L	\	l						л	э	Л	Э
			1	1	0	1	13	ВК	РГ	-	=	M]	m	}					м	щ	М	Щ
			1	1	1	0	14	ЛАТ	РЗ	.	>	A'	^	a	~					н	ч	Н	Ч
			1	1	1	1	15	РУС	РЭ	/	?	0	_	o	3б					о	_	О	3Б

регистре (позиция 4/1) будет соответствовать двоичный код 1000001, а во втором (позиция 6/1) — 1100001. Для различия символов обоих регистров перед посылаемой информацией передается соответственно либо символ ЛАТ, либо РУС (в новом стандарте они заменены соответственно символами ВХ и ВУХ).

Будучи хорошо согласованным с современной аппаратурой связи, 7-разрядный код является основным для аппаратуры передачи данных и устройств ввода-вывода абонентских пунктов. Однако необходимость посылать специальные символы, переключающие регистры, делает неудобным этот код для представления и произвольной обработки информации в машинах. Поэтому в качестве внутренних в ЕС ЭВМ приняты 8-разрядные коды. При таком кодировании каждому символу полного алфавита ставится в соответствие 8-разрядная машинная единица — байт. Существуют два способа 8-разрядного кодирования.

Один из них — 8-разрядный код для обмена информацией (КОИ-8), представленный табл. 2.2, строится на основе 7-разрядного кода КОИ-7. В КОИ-8 позиции двоичных цифр (бит) пронумерованы следующим образом: 87654321. При этом значения цифр в позициях 7, 6, 5, 4, 3, 2, 1 соответственно равны значениям Э₇, Э₆, Э₅, Э₄, Э₃, Э₂, Э₁ 7-разрядного кода. Значение же цифры

Таблица 2.2

Таблица кодирования символов 8-разрядными кодами, используемыми для обмена информацией в ЕС ЭВМ (КОИ-8)

								Зональная тетрада																		
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
7	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
6	0	0	1	1	0	0	1	1	0	1	1	0	0	1	1	0	0	1	0	0	1	1	1	1	1	1
5	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15										
0	0	0	0	0	0	0	0	0	ПУС	АР1	0	0	Р	Р	ВЦФ											
0	0	0	1	1	НЗ	СУ1	!	1	А	Q	q	НЗН										я		Я		
0	0	1	0	2	НТ	СУ2	"	2	В	Р	б	г	РП	УУК								б		5		
0	0	1	1	3	КТ	СУЗ	#	3	С	С	С	С										ц		Ц		
0	1	0	0	4	КП	СТП	х	4	Д	Т	а	т	БК	ВКП								д	Т	Д		
0	1	0	1	5	КТМ	НЕТ	%	5	Е	У	е	и	НС	ОУЧ										У		
0	1	1	0	6	ДА	СИН	&	6	F	V	f	v	СБ	ПБ								ф	Ж	Ф	Ж	
0	1	1	1	7	ЗВ	КБ	'	7	G	w	g	w	ОЖД									з	В	Г		
1	0	0	0	8	ВШ	АН	(8	Н	Х	н	х										ь		Ь		
1	0	0	1	9	ГТ	КН)	9	I	Y	i	y										и	Ы	И	Ы	
1	0	1	0	10	ПС	ЗМ	*	:	J	Z	j	z	УР									й	З	Й	З	
1	0	1	1	11	ВТ	АР2	+	; [k {														к	Ш	Ш		
1	1	0	0	12	ПФ	РФ	,	< L \ L					Вып									л	Э	Л	Э	
1	1	0	1	13	ВК	РГ	-	= M] m }					ВСТ									м	Щ	Щ		
1	1	1	0	14	РУС	РЗ	.	> N ^ n ~														н	Ч	Ч		
1	1	1	1	15	ЛАТ	РЭ	/	? 0 - o 3b														э		Э		
Цифровая тетрада																										

в позиции 8 зависит от того, какому регистру КОИ-7 соответствует столбец таблицы КОИ-8: если первому (латинскому), то в 8-й позиции будет 0, если второму (русскому), то — 1.

К преимуществам КОИ-8, по сравнению с КОИ-7, следует отнести отсутствие разделения кодовой таблицы на два регистра и возможность кодирования большего (до 256) количества символов. Последнее объясняется тем, что при 8-элементном кодировании нет необходимости повторять 64 символа, как это имеет место для содержимого столбцов 8—11 таблицы КОИ-7.

Другим внутренним кодом, на применение которого рассчитаны аппаратура и программное обеспечение ЕС ЭВМ, является двоичный код для обмена информацией (ДКОИ) *. Ему отдается предпочтение при нанесении цифровой информации на перфокарты, а также при обработке ее в машинах. Предусмотрены аппаратно-программные средства перекодирования данных в процессорах из КОИ-8 в ДКОИ и обратно. Кодирование символов согласно

* ДКОИ соответствует американскому расширенному двоично-кодированному десятичному коду для обмена информацией EBCDIC — Expanded Binary Coded Decimal Interchange Code.

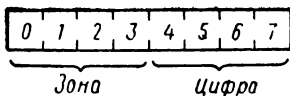


Рис. 2.1. Деление кодирующего байта на зональную и цифровую части

ДКОИ показано в табл. 2.3. Кодовая таблица содержит полный алфавит, включающий кроме стандартных управляющих символов ряд дополнительных. Наименование графических знаков, стандартных и дополнительных управляющих символов даны в приложении (табл. 7—9).

Двоичные разряды в 8-разрядных кодовых комбинациях ДКОИ нумеруются слева направо цифрами от 0 до 7: 01234567. В кодовой таблице символы определенного целевого назначения размещены группами по зонам. Для удобства указания двоичного номера зоны и двоичного номера любого из расположенных в группе символов кодирующий байт делится на две 4-разрядные части (тетрады): зональную и цифровую (рис. 2.1).

Двоичные цифры в 0-м и 1-м разрядах кодирующего байта ДКОИ определяют:

- 00 — зону служебных символов;
- 01 — зону графических знаков;
- 10 — зону строчных латинских и русских букв;
- 11 — зону прописных латинских и русских букв и цифр.

Таблица 2.3

Таблица двоичного кодирования символов 8-разрядными кодами, используемыми для обмена информацией в ЕС ЭВМ (ДКОИ)

		Зональная тетрада																					
		0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1						
	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1						
	1	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1						
	2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1						
	3	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1						
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F								
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
	0	0	0	1	1	НЗ	СУ1	НЗН		/	а	ј	ы	А	Ј	І	І						
	0	0	1	0	0	2	НТ	СУ2	РП	СИН				б	к	ѕ	з	В	К	Ѕ	2		
	0	0	1	1	0	3	КТ	СУЗ						с	l	t	ш	С	L	T	3		
	0	1	0	0	0	4	ВЫП	ВСТ	БК	ВКП				а	т	и	э	Д	М	U	4		
	0	1	0	1	0	5	ГТ	НС	ПС	ОУЧ				е	п	в	щ	Е	N	V	5		
	0	1	1	0	0	6	СБ	ВШ	КБ	ПБ				ю	ф	о	ш	Ч	F	O	W	6	
	0	1	1	1	0	7	ЗБ	ОЖД	АР2	КП				г	р	х	ь	Г	P	X	7		
	1	0	0	0	0	8		АН						б	h	q	у	Ю	H	Q	Y	8	
	1	0	0	1	0	9		КН						\	i	г	z	J	R	Z		9	
	1	0	1	0	0	A		УУК	УР		[]	!	:	д	к	Б						3	
	1	0	1	1	0	В	ВТ				.	ѣ	,	#	л	ц	и	у	Ш				
	1	1	0	0	0	С	ПФ	РФ		СТП	<	*	%	@	ф	м	т	Д	Й	П	Ж	Э	
	1	1	0	1	0	Д	ВК	РГ	КТМ	НЕТ	()	'	г	н				Я				Щ	
	1	1	1	0	0	Е	РУС	РЗ	ДА		+	:	>	=				Ж	Ф	Л		Ь	Ч
	1	1	1	1	0	Ф	ЛАТ	РЭ	ЗВ	ЗМ	!	?	"	и	п	в	Г					Ы	ЗБ

Цифровая тетрада

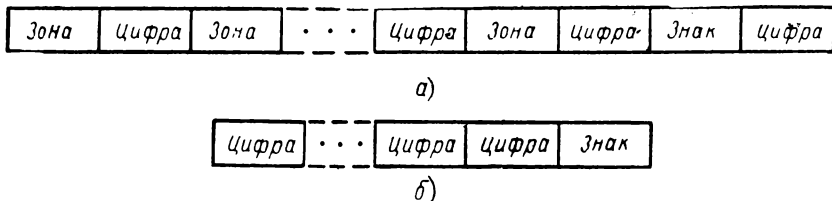


Рис. 2.2. Представление многоразрядных десятичных чисел:
а — в распакованном; *б* — в упакованном форматах

Следующие две цифры во 2-м и 3-м разрядах байта совместно с первыми двумя указывают столбец зоны, в которой находится символ, окончательно адресуемый последними четырьмя цифрами байта. Например, код 0001 в цифровой части байта в зонах 10 и 11 используется для кодирования букв *A, J, a, j*, диакритического знака $\bar{\quad}$, буквы *ы* и десятичной цифры 1. Конкретный же код буквы *A* есть 11000001, буквы *a* — 10000001 и т. п. Если десятичным цифрам от 0 до 9 соответствуют в цифровой части байта двоично-десятичные коды от 0000 до 1001, то при размещении их в столбце 11 зоны 11 им будут соответствовать восьмиразрядные двоичные коды: 11110000; 11110001; 11110010; ...; 11111001 (в шестнадцатиричной системе F0—F9).

Следует заметить, что на зональную и цифровую тетрады разбиваются и кодирующие байты КОИ-8. При кодировании в КОИ-8 в зональной тетраде кодов десятичных цифр содержится двоичный код 0101, в цифровой — двоично-десятичные коды от 0000 до 1001. Для обеспечения однозначного восприятия кодов символов в машине в слове состояния программы (ССП) выделен специальный разряд (12-й), содержимое которого указывает, в каком коде представлена информация. *Коду ДКОИ в 12-м разряде ССП соответствует 0, а коду КОИ-8—1.*

Многоразрядные десятичные числа, вводимые в машину в так называемом распакованном формате, показанном на рис. 2.2, *а*, для непосредственного выполнения действий над ними по специальной команде сворачиваются в упакованный формат, изображенный на рис. 2.2, *б*. Упакованное десятичное число в каждом байте содержит две двоично-десятичные цифры. В распакованном формате код знака числа размещается в зональной тетраде младшего (правого) байта. В коде ДКОИ знаки плюс и минус десятичных чисел кодируются соответственно как 1100 и 1101 (С и Д), в КОИ-8 те же знаки изображаются кодами 1010 и 1011 (А и В). Кроме 1100 и 1010, в качестве знака плюс могут восприниматься также комбинации 1110 и 1111, хотя последняя в распакованной форме при коде ДКОИ обычно представляет зону. При упаковке код знака перемещается в правую тетраду крайнего правого байта. Если необходимо, то десятичные числа программным способом

могут быть переведены в двоичную систему счисления. При выдаче из машины результатов в виде десятичных чисел последние по специальной команде преобразуются в распакованную форму (с зонами).

2.3. ФОРМАТЫ ДАННЫХ И РАСПОЛОЖЕНИЕ ИНФОРМАЦИИ В ОСНОВНОЙ ОПЕРАТИВНОЙ ПАМЯТИ

Цифровые вычислительные машины Единой системы относятся к *символьноориентированным* ЭВМ. При исполнении каждой команды, в соответствии с указаниями, содержащимися в ней, они обрабатывают по одному или несколько квантов информации. Каждый такой квант эквивалентен либо символу определенного входного языка, либо числу или его части в зависимости от его величины и системы счисления, в которой оно выражено.

Для создания системы машин, процесс обработки информации в которых был бы независим от размеров преобразуемых слов, в качестве основного кванта машинной информации был принят байт — 8-разрядный двоичный слог. Этот выбор обусловил внедрение единой иерархической структуры данных. В порядке увеличения количества разрядов эта структура содержит (см. рис. 2.3) фиксированные составные единицы — *полуслово*, *слово*, *двойное слово* и информационную единицу переменной длины — *поле*.

Полуслово состоит из двух байт, слово — из четырех и двойное слово — из восьми (соответственно — 16, 32 и 64 двоичных разрядов); поле может быть любой длины в пределах от одного до 256 байт. Для выполнения операций над операндами, соответствующими перечисленным единицам, в системе команд ЕС ЭВМ предусмотрены необходимые команды.

Если в указанных единицах размещаются двоичные числа, то располагаются они так, что веса их разрядов возрастают справа

байт

01110010

0 1 2 3 4 5 6 7

Полуслово

1110001111010110

0 7 8 15

Слово

1100010111000011110001101110000

0 7 8 15 16 23 24 31

Двойное слово

011100101100001011010100110001011100001111010111111010100000000

0 7 8 15 16 23 24 31 32 39 40 47 48 55 56 63

Поле максимальной длины

111100011111100111101111110011010101001010101000100110101001101011001000101

0 7 8 15 16 23 24 31 32 2040 2047

Рис. 2.3. Форматы данных

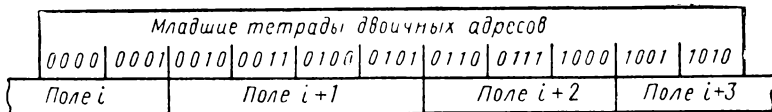


Рис. 2.4. Пример расположения полей данных в оперативной памяти

налево. Однако независимо от этого, разрядные позиции двоичных цифр всегда формально нумеруются слева направо, начиная с нуля. В байте разряды нумеруются от 0 до 7, в полуслове — от 0 до 15, в слове — от 0 до 31, в двойном слове — от 0 до 63 и в поле, например, максимальной длины (8×256) — от 0 до 2047.

Рассмотренная структура данных позволяет более плотно размещать информацию в оперативных запоминающих устройствах. Для оперативной памяти наименьшей адресуемой единицей является байт. При задании адреса операнда, соответствующего полуслову, слову, двойному слову или полю, указывается адрес старшего (левого) байта. Для поля еще необходимо в команде указать количество составляющих его байт.

Адрес поля, т. е. адрес его левого байта, может быть произвольным. Поэтому поля различной длины, как показано на рис. 2.4, могут располагаться вплотную друг за другом в порядке возрастания их адресов. На адреса же составных единиц фиксированной длины налагаются ограничения, при выполнении определенных команд упрощающие обращение к оперативным запоминающим устройствам. Например, адрес полуслова всегда должен быть кратен 2, слова — кратен 4 и двойного слова — кратен 8. Иными словами, адрес (граница) составной единицы фиксированной длины всегда должен быть кратен количеству содержащихся в ней байтов. Такие границы получили название целочисленных. Как показано на рис. 2.5, в двоичных кодах адресов, соответствующих целочисленным границам полуслова, слова и двойного слова, соответственно в одном, двух и трех младших разрядах должны содержаться нули.

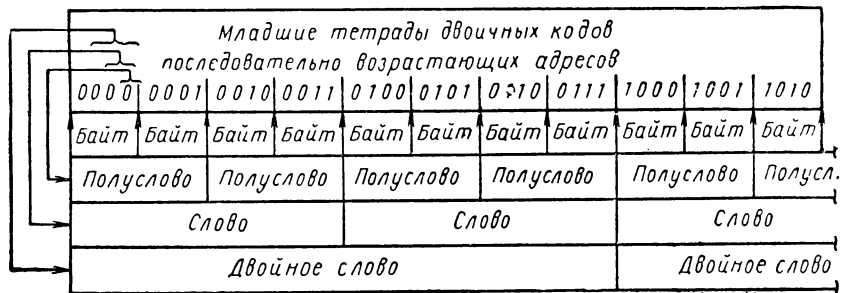


Рис. 2.5. Границы полуслов, слов и двойных слов при расположении их в оперативной памяти

Длина операнда фиксированной длины определяется кодом операции в команде. Правильность задания в командах адресов таких операндов автоматически контролируется в машинах. Если обнаруживается, что адрес полуслова, слова или двойного слова не соответствует должной целочисленной границе, то вырабатывается сигнал требования прерывания работы машины, называемый «Неправильная спецификация».

Передача и преобразование данных всех форматов в машинах Единой системы контролируется побайтно. Каждый байт информации в машинах автоматически сопровождается дополнительным (9-м) контрольным разрядом. Он вырабатывается из условия проверки 9-разрядных кодовых комбинаций на нечетность (количество единиц в разрядах байта, включая контрольный, должно быть нечетным). Количество контрольных разрядов, сопровождающих любой многобайтовый операнд или команду, равно количеству содержащихся в них байт. Обнаружение ошибки при проверке информации на нечетность в каком-либо месте машины приводит к выработке сигнала требования прерывания работы машины от схем контроля.

2.4. ФОРМЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ

Одной из отличительных особенностей вычислительных машин Единой системы является то, что они могут обрабатывать числовую информацию различной разрядности. Количества разрядов могут ограничиваться как рядом фиксированных форматов (полуслово, слово и двойное слово), так и быть переменными (выраженными определенным количеством байт) в пределах поля определенной длины. Расчеты могут выполняться как в двоичной системе счисления, так и в десятичной.

Для представления двоичных чисел различных порядков применяются две формы изображения: *полулогарифмическая (с плавающей запятой)* и *естественная (с фиксированным положением запятой)*. Десятичные числа представляются только в естественной форме (с фиксированной запятой). В обеих системах счисления числа в естественной форме представляются и обрабатываются только как целые.

Как и в машинах предшествовавших выпусков, в ЭВМ Единой системы полулогарифмическая форма представления основывается на изображении чисел в виде

$$A = \pm p, \pm m, \quad (2.4)$$

соответствующем записи

$$A = N^{\pm p} (\pm m), \quad (2.5)$$

где N — основание системы счисления;

p — целое число, выражающее порядок числа A ;

m — мантисса числа A , причем всегда $|m| < 1$.

Запись вида (2.5) называется полулогарифмической потому, что в логарифмической форме представляется не все число, а только его часть $N^{\pm p}$.

В записи (2.4) положение запятой в мантиссе m определяется величиной порядка p . С изменением порядка в большую или меньшую сторону запятая соответственно перемещается влево или вправо, т. е. «плавает» в изображении числа.

В машинах единой системы имеются две особенности в представлении чисел в полулогарифмической форме.

Первая особенность состоит в том, что основание системы счисления N принято равным $2^4 = 16$, а не 2, как это обычно имело место в ранее выпускавшихся ЭВМ. Применение такого основания имеет два преимущества: во-первых, существенно расширяется диапазон представления чисел при использовании для кода порядка того же количества разрядов, что и ранее при основании 2; во-вторых, представляется возможность достаточно простого ускорения выполнения операций, содержащих сдвиги, поскольку мантисса теперь может рассматриваться состоящей из шестнадцатиричных цифр, записанных двоичными тетрадами. Для таких мантисс легко организуются ускоренные сдвиги вправо и влево на 4 разряда или на кратное этой величине, например, на 8 разрядов. Каждый сдвиг мантиссы на 4 разряда вправо или влево равносильно соответственно ее делению или умножению на $2^4 = 16$. Поэтому такое перемещение на одну шестнадцатиричную позицию требует соответствующего изменения порядка шестнадцатиричного числа всего лишь на единицу. Например, $16^0 \cdot 2^{-4} = 16^1 \cdot 2^{-8}$, где мантиссы 2^{-4} и 2^{-8} в двоичном изображении имеют вид 0,0001 и 0,0000 0001.

Представленное в полулогарифмической форме число, у которого абсолютная величина мантиссы удовлетворяет условию

$$\frac{1}{N} \leq |m| < 1, \quad (2.6)$$

называется *нормализованным*. При $N = 16$ мантиссы нормализованных чисел оказываются заключенными в пределах $1/16$ и 1 или в двоичном изображении: 0,00010...0 и 0,11111...1.

Если для значащих разрядов мантиссы m и порядка p в машине отводится соответственно l и q двоичных разрядов, тогда диапазон представления нормализованных шестнадцатиричных A_{16} чисел, взятых по абсолютному значению, будет определяться пределами:

$$16^{-1} \cdot 16^{-(2^q-1)} \leq |A_{16}| \leq \left(1 - 16^{-\frac{l}{4}}\right) 16^{2^q-1}, \quad (2.7)$$

где 16^{-1} и $1 - 16^{-\frac{l}{4}}$ — наименьшее и наибольшее абсолютные значения нормализованных мантисс. В машинах Единой системы

$l > 20$ и $q = 6$. Поэтому, полагая с достаточной точностью $1 - 16^{-\frac{l}{4}} = 1 - 16^{\frac{20}{4}} \cong 1$, получим

$$16^{-64} \leq |A_{16}| \leq 16^{63},$$

что в десятичной системе эквивалентно

$$10^{-77} \leq |A_{10}| \leq 10^{76}.$$

Этот диапазон существенно больше, чем при тех же l и q для двоичных чисел. Действительно, для нормализованных двоичных чисел

$$2^{-1} 2^{-(2^q-1)} \leq |A_2| \leq (1 - 2^{-l}) \cdot 2^{2^q-1}. \quad (2.8)$$

Полагая $1 - 2^{-l} \cong 1$, получим

$$2^{-64} \leq |A_2| \leq 2^{63}$$

или в десятичной системе

$$10^{-19} \leq |A_{10}| \leq 10^{19}.$$

Нормализованное представление чисел позволяет сохранять в разрядной сетке большее количество значащих цифр и, следовательно, повышает точность вычислений, а также упрощает действия над порядками и мантиссами в процессе счета. Однако может возникнуть необходимость в выполнении действий и над ненормализованными числами. Поэтому в системе команд ЕС ЭВМ предусмотрена возможность выполнения действий как с автоматической нормализацией результатов, так и без нормализации.

Нормализация числа, у которого в мантиссе одна и более старших шестнадцатиричных цифр равны нулю, производится путем сдвига мантиссы влево. Сдвиг производится до тех пор, пока старшая шестнадцатиричная цифра (первая тетрада правее запятой) не станет отличной от нуля. При каждом сдвиге на 4 разряда влево порядок уменьшается на единицу.

Если в ходе вычислений получится мантисса, большая единицы (нарушение нормализации влево, в отличие от рассмотренного выше нарушения нормализации вправо), то в этом случае она приводится к виду, удовлетворяющему условию (2.6), путем сдвига ее вправо. Младшие разряды, выходящие за пределы разрядной сетки, отбрасываются.

Мантисса, равная нулю, не может быть нормализована. Случаи возникновения таких мантисс в результате выполнения операций сложения и вычитания, классифицируемые как потеря значимости, автоматически обнаруживаются в процессорах. При этом нормализация не производится, а вырабатывается требование прерывания.

Вторая особенность представления двоичных чисел в полугарифмической форме в ЕС ЭВМ состоит в том, что порядки чисел представляются увеличенными на 2^q . При шести разрядах, отво-

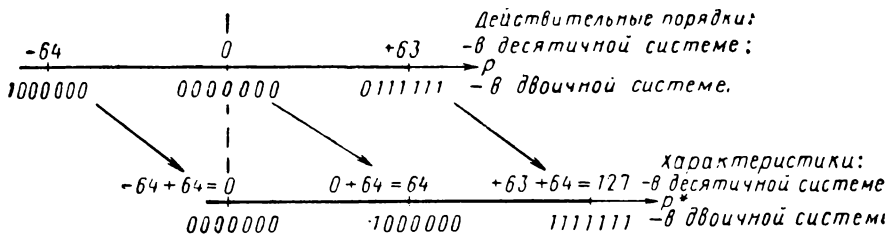


Рис. 2.6. Соотношения между величинами характеристик и действительными порядками

димых на представление порядка ($q = 6$ без знакового разряда), $2^q = 64$. В результате отрицательные и положительные порядки p , увеличившись на 64, т. е. превратившись в $p^* = p + 64$, оказываются смещенными на числовой оси в положительном направлении, как показано на рис. 2.6. Порядки диапазона

$$-2^q \leq p \leq 2^q - 1, \quad (2.9)$$

т. е.

$$-64 \leq p \leq 63,$$

теперь ограничиваются пределами

$$0 \leq p^* \leq 2^{q+1} - 1, \quad (2.10)$$

что соответствует $0 \leq p^* \leq 127$.

Чтобы не путать действительные и условные величины, смещенные порядки называют *характеристиками чисел*. Так как все характеристики являются положительными, то это упрощает выполнение некоторых действий над ними, например сравнения. Для этого вычитание из одной характеристики другой можно производить без предварительного анализа их знаков.

При сравнительно небольших количествах разрядов, отводимых на мантиссы и порядки, полулогарифмическая форма обеспечивает весьма широкий диапазон представления чисел. Это удобно для решения научных и инженерных задач, при решении которых часто приходится иметь дело с величинами, изменяющимися в больших пределах, во многих случаях заранее не известных.

Кроме действий над числами в полулогарифмической форме во многих случаях возникает необходимость в выполнении арифметических и логических операций над числами, представленными в естественной форме с фиксированным положением запятой. При этом числа, которыми приходится оперировать, либо просто являются целыми, либо их удобно представлять целыми в расчетах. Всегда целыми, например, являются некоторые числа, над которыми приходится выполнять действия как при трансляции программ с входных языков на машинные, так и при организации самого вычислительного процесса в ходе выполнения программ. К ним относятся адреса, константы модификации, счетчики и т. п.

Целыми удобно представлять числа, когда определение порядков результатов несложно: диапазон изменения исходных величин сравнительно невелик, и над ними выполняются в основном операции сложения, вычитания или логические. К задачам, при решении которых возможно оперирование такими данными, например, относятся учетно-плановые, статистические и другие расчеты экономического характера.

Для обеспечения возможности эффективной обработки информации указанного выше типа в ЕС ЭВМ предусмотрены аппаратные и программные средства для выполнения действий над числами, представленными в естественной форме. При этом запятая подразумевается фиксированной справа от младших цифр, т. е. все числа рассматриваются целыми. Такие действия могут выполняться как над двоичными числами, так и непосредственно над десятичными, цифры которых представлены двоичными тетрадами. В любом случае числа участвуют в операциях выравнивания по младшим разрядам (десятичные по младшим тетрадам).

По сравнению с широко применявшимся ранее выполнением действий над дробными числами, когда запятая фиксировалась слева от старших значащих разрядов чисел, использование целочисленной арифметики позволяет более экономно расходовать емкость оперативной памяти для размещения чисел. Для достижения требуемой точности нет необходимости учитывать большое количество младших разрядов, как это имело место при оперировании правильными дробями. Выравнивание по правой границе (по младшим разрядам) уменьшает вероятность возникновения переполнения, т. е. появления результата, по величине превосходящего максимально допустимый при количестве разрядов, предусмотренных командой для его размещения в оперативной памяти.

Диапазон представления целых двоичных чисел A_2^n в n -разрядных форматах данных ограничивается пределами

$$0 \leq |A_2^n| \leq 2^n - 1. \quad (2.11)$$

Например, при $n = 31$ $0 \leq |A_2^n| \leq 2^{31} - 1$, что соответствует диапазону чисел в десятичной системе $0 \leq |A_{10}^n| \leq 10^9$.

Для многих практических задач достаточный диапазон представления имеют и десятичные числа, цифры которых, выраженные в двоично-десятичной форме, в целом занимают столько же двоичных разрядов, сколько и двоичные числа. В формате, имеющем n двоичных разрядов, можно разместить любое целое десятичное число, представленное двоичными тетрадами,

$$0 \leq |A_{2-10}^n| \leq 10^{\frac{n}{4} - 1} - 1. \quad (2.12)$$

Например, при $n = 32$ верхний предел ограничивается числом $10^7 - 1$. Показатель степени $\frac{n}{4} - 1$ учитывает то обстоятельство, что одна тетрада используется для представления знака.

Непосредственное применение в ЭВМ десятичной арифметики особенно выгодно для сравнительно несложных вычислений над большими массивами исходных данных. В этих случаях из-за отсутствия необходимости перевода из десятичной системы счисления в двоичную и обратно получается существенный выигрыш во времени.

В ЕС ЭВМ двоичные числа с фиксированной запятой выражаются в виде операндов фиксированной длины: полусловом или словом. Десятичные данные в естественной форме размещаются в цифровых полях в распакованном или упакованном форматах. Непосредственно в арифметических операциях десятичные числа участвуют в упакованном формате. Поля, где размещаются десятичные операнды, могут начинаться с любого байта оперативной памяти, иметь длину от одного до 16 байт. Поэтому операции десятичной арифметики относятся к операциям над полями переменной длины.

Кроме десятичных чисел, в полях переменной длины могут располагаться различные двоичные коды смысловой информации, не имеющей количественного значения. В этих случаях максимальная длина поля может достигать до 256 байт.

2.5. МАШИННЫЕ КОДЫ ЧИСЕЛ

Двоичные числа. Коды знаков указываются в специальных разрядах форматов двоичных чисел, где знак плюс изображается цифрой 0, а знак минус — цифрой 1.

Как при естественной, так и при полулогарифмической форме представления чисел, код знака располагается слева от значащей части числа. При нумерации двоичных позиций разрядной сетки формата числа слева направо, начиная с нуля до $n-1$, знаковому разряду всегда соответствует нулевая позиция.

Естественно выраженные целые числа располагаются в отводимых для них форматах так, что младшие их разряды находятся в самых правых разрядных позициях форматов. Это означает, что в любом формате запятая всегда считается зафиксированной справа от самой крайней правой ($n-1$) позиции. В оперативной памяти двоичные числа с фиксированной запятой могут быть представлены в двух форматах: в виде 32-разрядного слова или 16-разрядного полуслова (рис. 2.7). Например, положительное целое число $+10110111$ в формате полуслова будет иметь вид $0.0000000\ 10110111$ *.

При выборке из оперативной памяти операндов длиной в полуслово они автоматически преобразуются в операнды длиной в слово. В таком виде они и участвуют в операциях с фиксирован-

* В дальнейшем тексте для упрощения чтения кодов чисел коды знаков отделяются от значащих частей точкой внизу, а место фиксирования запятой при необходимости указывается вверх.

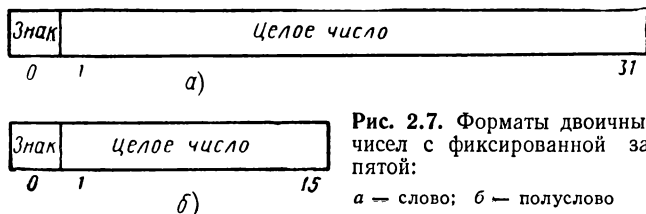


Рис. 2.7. Форматы двоичных чисел с фиксированной запятой:

а — слово; б — полуслово

ной запятой. В некоторых операциях, таких как умножение, деление, сдвиг и преобразование из одной системы счисления в другую, могут выполняться действия над 64-разрядными кодами целых чисел (двойными словами). Такие числа либо получаются в результате выполнения операций, например, умножения, либо формируются перед операцией специальным образом, например, перед делением.

Коды операндов в полулогарифмической форме также размещаются в форматах фиксированной длины. В операциях с плавающей запятой могут применяться два формата (рис. 2.8): короткий, длиной в слово (32 разряда), и длинный — в двойное слово (64 разряда). В любом из этих форматов нулевая разрядная позиция отводится для изображения знака числа, а следующие 7 — для характеристики. Поле мантиссы располагается правее характеристики числа.

Так как для представления чисел в полулогарифмической форме основание системы счисления принято равным 16, то в поле мантиссы слова или двойного слова соответственно располагается либо шесть, либо четырнадцать шестнадцатиричных цифр (двоичных тетрад). Числа могут представляться нормализованными или ненормализованными. Это зависит от характера производимых вычислений.

При 7 разрядах, отводимых для представления характеристик, все порядки от -1000000 до $+0111111$ ($-64 \div +63$) согласно соотношению (2.10) в смещенном виде кодируются целыми положительными числами от $0000000'$ до $1111111'$ (рис. 2.6). Для получения характеристики истинный порядок как число со знаком складывается с числом 64. В результате коды характеристик, соответ-

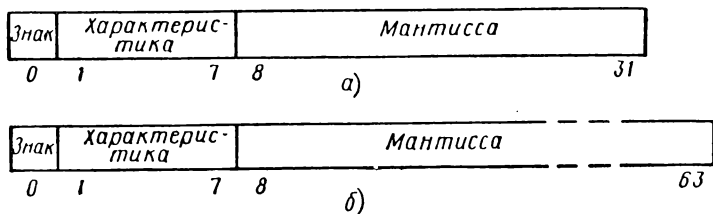


Рис. 2.8. Форматы двоичных чисел с плавающей запятой:

а — слово; б — двойное слово

ствующих отрицательным порядкам, во второй разрядной позиции формата слова или двойного слова имеют 0, а для положительных порядков — 1. Нулевому порядку соответствует характеристика 1000000'.

С учетом всего изложенного выше, число, например, —32756, переведенное в двоичную систему счисления и представленное в полулогарифмической форме, в формате 32-разрядного слова будет иметь вид

1.100 0101' 0111 1111 1111 0100 0000 0000.

Для шестнадцатиричного основания изображенная мантисса является нормализованной, характеристика $p^* = 5 + 2^6 = 69$. Весь код в шестнадцатиричной системе можно записать как C57FF400.

Число с нулевой характеристикой, нулевой мантиссой и положительным знаком называется истинным нулем. Он может появиться в результате арифметической операции при определенных величинах операндов. Как было отмечено в предыдущем параграфе, результат операции может быть также принудительно сделан равным истинному нулю, если имеет место исчезновение порядка (в результате операции получилась характеристика, меньшая нуля) или если мантисса результата равна нулю, и не предусмотрено прерывание исполняемой программы при потере значимости.

Знак суммы, разности, произведения или частного, имеющего нулевую мантиссу, всегда положителен. Знак мантиссы для других операций устанавливается в соответствии с правилами алгебры в зависимости от знаков операндов.

В представлении значащих частей положительных и отрицательных чисел с фиксированной и плавающей запятой в оперативной памяти и в процессе выполнения над ними операций имеется ряд особенностей.

Во всех моделях ЭВМ Единой системы *положительные двоичные числа с фиксированной запятой* хранятся в оперативной памяти и участвуют в операциях в *прямом коде, отрицательные — в дополнительном*.

Прямой код $[A^u]_{\text{пр}}$ целого положительного числа $A^u = a_{m-1}a_{m-2} \dots a_1a_0$ получается следующим образом:

$$[A^u]_{\text{пр}} = [n(0)] + A^u, \quad (2.13)$$

где n — количество позиций в разрядной сетке формата данных; $[n(0)]$ — n -разрядный код нуля и $A^u \leq 10^{n-1} - 1$ (10 — два для двоичной системы счисления), т. е. $m \leq n - 1$. Например, в 8-разрядном формате ($n = 8$) число $A^u = +101$ согласно (2.13) в прямом коде будет иметь вид

$$[A^u]_{\text{пр}} = 0000 0000 + 101 = 0.000 0101'.$$

Дополнительный код $[A^u]_{\text{доп}}$ целого отрицательного числа является его дополнением до 10^n , т. е.

$$[A^u]_{\text{доп}} = 10^n + A^u, \quad (2.14)$$

где n — также разрядность формата и 10 — два для двоичной системы счисления. Например, при $n = 8$ для $A^u = -1011$

$$[A^u]_{\text{доп}} = 10000\ 0000 + (-1011) = 1.111\ 0101'.$$

Можно сформулировать правило, согласно которому дополнительный код целого отрицательного числа, записанного в прямом коде в разрядной сетке того или иного формата данных получается следующим образом: в разряде знака сохраняется единица, а в остальных разрядах формата единицы заменяются нулями, а нули — единицами, и к младшему разряду прибавляется единица. Например, если для $A^u = -1011$ в 8-разрядном формате

$$[A^u]_{\text{пр}} = 1.000\ 1011',$$

то, действуя согласно правилу, получим

$$\begin{array}{r} 1.111\ 0100 \\ + 1 \\ \hline [A^u]_{\text{доп}} = 1.111\ 0101' \end{array}$$

Принятый в ЕС ЭВМ способ представления чисел с фиксированной запятой удобен для расширения и сжатия форматов, в которых они размещаются. Объясняется это тем, что при указанном способе любое число можно рассматривать как младшие разряды сколь угодно длинного числа. Когда оно положительно, в формате все разряды левее его старшего значащего разряда, в том числе и знаковый, равны нулю. В противном случае, когда число отрицательно, все разряды левее старшего значащего разряда, включая знаковый, равны единице. Поэтому увеличение длины операнда в сторону старших разрядов достигается расширением поля целой части числа влево и присвоением каждому новому разряду значения знакового разряда исходного числа.

По существу значение содержимого нулевой позиции любого формата выражает не просто знак числа, а показывает, в каком коде (прямом или дополнительном) оно представлено. Поэтому, если даже на примере 8-разрядных целых двоичных чисел рассмотреть их предельные значения, то нетрудно определить, что при использовании дополнительного кода общее количество отрицательных чисел оказывается на единицу больше, чем положительных.

Целое число	Десятичное значение	Знак ↓	Двоичное изображение
$2^7 - 1 =$	$127 =$		0.111 1111
$2^0 =$	$1 =$		0.000 0001
$0 =$	$0 =$		0.000 0000
$-2^0 =$	$-1 =$		1.111 1111
$-2^7 + 1 =$	$-127 =$		1.000 0001
$-2^7 =$	$-128 =$		1.000 0000

Первое число $2^7 - 1$ в рассматриваемом формате является наибольшим положительным, последнее -2^7 — наименьшим отрицательным. В рассматриваемом представлении нуль имеет единственное изображение 0.000...0. Действительно, преобразование отрицательного нуля $[0]_{\text{пр}} = 1.000...0$ в дополнительный код $[0]_{\text{доп}}$ дает 0.000...0.

Как известно, использование дополнений для представления отрицательных чисел позволяет все арифметические операции свести либо просто к сложению (алгебраическое сложение и вычитание), либо к сериям сложений и сдвигов (умножение и деление). Замена вычитания двоичных чисел $[A^u]_{\text{пр}} - [B^u]_{\text{пр}}$ сложением с дополнением $[A^u]_{\text{пр}} + [-B^u]_{\text{доп}}$ позволяет оперировать со всеми разрядами в формате, расположенными левее старшего значащего, включая знаковый, точно так же, как и со значащими. При использовании дополнительного кода для представления отрицательных чисел в процессе суммирования операндов единица переноса из знакового разряда не учитывается.

Пример 2.8. Сложить двоичные числа, соответствующие десятичным: $A_{10}^u = 69$ и $B_{10}^u = -54$. Двоичные числа представляются в формате полуслова.

$$\begin{array}{r}
 [A_2^u]_{\text{пр}} = 0.000\ 0000\ 0100\ 0101 \\
 [B_2^u]_{\text{пр}} = 1.000\ 0000\ 0011\ 0110 \\
 + \quad [69]_{\text{пр}} \quad [A_2^u]_{\text{пр}} = 0.000\ 0000\ 0100\ 0101 \\
 \underline{[-54]_{\text{доп}}} \quad [B_2^u]_{\text{доп}} = 1.111\ 1111\ 1100\ 1010 \\
 \hline
 \underbrace{1\ 111\ 1111\ 1000\ 000}_{\leftarrow} \oplus \text{Межразрядные переносы} \\
 [15]_{\text{пр}} [A_2^u + B_2^u]_{\text{пр}} = 0.000\ 0000\ 0000\ 1111 \quad \text{Сумма}
 \end{array}$$

Символ \oplus означает суммирование по mod 2. В данном примере поразрядно суммируются коды слагаемых и значения межразрядных переносов. Так как $A > B$, то результат получился в прямом коде.

Пример 2.9. Сложить двоичные числа, соответствующие десятичным: $A_{10}^u = 54$ и $B_{10}^u = -69$. Двоичные числа представляются в формате полуслова.

$$\begin{array}{r}
 [A_2^u]_{\text{пр}} = 0.000\ 0000\ 0011\ 0110 \\
 [B_2^u]_{\text{пр}} = 1.000\ 0000\ 0100\ 0101 \\
 + \quad [54]_{\text{пр}} \quad [A_2^u]_{\text{пр}} = 0.000\ 0000\ 0011\ 0110 \\
 \underline{[-69]_{\text{доп}}} \quad [B_2^u]_{\text{доп}} = 1.111\ 1111\ 1011\ 1011 \\
 \hline
 \underbrace{1\ 000\ 0000\ 0111\ 110}_{\leftarrow} \oplus \text{Межразрядные переносы} \\
 [-15]_{\text{доп}} [A_2^u + B_2^u]_{\text{доп}} = 1.111\ 1111\ 1111\ 0001 \quad \text{Сумма} \\
 [-15]_{\text{пр}} [A_2^u + B_2^u]_{\text{пр}} = 1.000\ 0000\ 0000\ 1111 \quad \text{Результат в прямом коде}
 \end{array}$$

Так как $A < B$, то сумма получилась в дополнительном коде.

В машинах отрицательные числа с фиксированной запятой в оперативной памяти хранятся в дополнительном коде, поэтому перед выполнением операции сложения их преобразование, как это имело место в рассмотренных выше примерах, не требуется. Двоичные коды просто складываются. Предварительный анализ знаков операндов, за исключением случаев, когда полуслово расширяется до слова, оказывается ненужным. Если же требуется выполнить операцию вычитания, то для второго операнда, независимо от его кода, всегда определяется дополнение до 2^n , где n — разрядность формата вычитаемого. Затем это вновь полученное дополнение прибавляется к уменьшаемому. Результат операции в том коде, в каком он получился, отсылается в оперативную память.

В результате выполнения арифметической операции может получиться число, по абсолютной величине превосходящее максимально допустимое для определяемого командой формата результата. Такой случай фиксируется в машинах как *переполнение с фиксированной запятой*. Полученный результат оказывается неправильным и без соответствующей коррекции не может быть использован в дальнейших вычислениях.

Переполнение с фиксированной запятой возникает при получении результата, либо превышающего максимально возможное положительное число 0111...1, либо меньшего отрицательного числа 1000...0. Оно автоматически определяется по значениям переносов в знаковый разряд и из знакового разряда. В примерах 2.8 и 2.9 эти межразрядные переносы обведены рамками.

Если при сложении переносы в знаковый разряд и из знакового разряда согласуются, т. е. либо оба отсутствуют, либо одновременно возникают, то переполнения нет. Сложение выполняется нормально, как было показано в примерах 2.8 и 2.9. *Наличие же переноса только в знаковый разряд или только из знакового разряда указывает на то, что имеет место переполнение.*

Пример 2.10. К двоичному числу, соответствующему десятичному $A_{10} = 69$ и представленному в 8-разрядном формате, прибавить равное ему.

$$\begin{array}{r}
 \begin{array}{l}
 [69]_{\text{пр}} \\
 + \\
 [69]_{\text{пр}} \\
 \hline
 \end{array}
 \quad
 \left.
 \begin{array}{l}
 [A_2^{\text{II}}]_{\text{пр}} = 0.100\ 0101 \\
 [A_2^{\text{II}}]_{\text{пр}} = 0.100\ 0101 \\
 \quad \quad \quad \boxed{0\ 1} \ 000\ 101 \leftarrow \\
 \hline
 \end{array}
 \right\} \oplus \text{Межразрядные переносы} \\
 \\
 [138]_{\text{пр}} \quad [A_2^{\text{II}} + B_2^{\text{II}}]_{\text{пр}} = 1.000\ 1010 \quad \text{Сумма}
 \end{array}$$

Возникло *переполнение*, так как переносы в знаковый разряд и из знакового (1 и 0) не совпадают.

Пример 2.11. Сложить равные числа, те же что и в предыдущем примере, но отрицательные.

$$\begin{array}{r}
 \begin{array}{r}
 (-69)_{\text{доп}} \\
 + \\
 (-69)_{\text{доп}} \\
 \hline
 \end{array}
 \quad
 \begin{array}{r}
 [A_2^{\text{н}}]_{\text{доп}} = 1.011\ 1011 \\
 [A_2^{\text{н}}]_{\text{доп}} = 1.001\ 1011 \\
 \hline
 \boxed{1}\boxed{0}111\ 011 \leftarrow
 \end{array}
 \left. \vphantom{\begin{array}{r} (-69)_{\text{доп}} \\ + \\ (-69)_{\text{доп}} \\ \hline \end{array}} \right\} \oplus \begin{array}{l} \text{Межразрядные} \\ \text{переносы} \end{array} \\
 \hline
 \begin{array}{r}
 (-138)_{\text{доп}} \\
 [A_2^{\text{н}} + A_2^{\text{н}}]_{\text{доп}} = 0.111\ 0110
 \end{array}
 \quad
 \text{Сумма}
 \end{array}$$

Возникло *переполнение*, так как переносы в знаковый разряд и из знакового (0 и 1) не совпадают.

При положительном переполнении содержимое знакового разряда суммы (1) ложно указывает на отрицательное число, а при отрицательном переполнении (0) — на положительное.

Анализ переполнения по переносам не позволяет получать дополнение от минимально возможного отрицательного числа 1.000...0. Если, например, в результате вычитания из нуля такого числа образуется его дополнение, то нетрудно убедиться, что число остается неизменным и формируется сигнал переполнения с фиксированной запятой.

Положительные и отрицательные числа с плавающей запятой (их характеристики и мантисы) во всех моделях ЭВМ Единой системы в оперативной памяти хранятся в прямом коде. Знак числа определяется по значению двоичной цифры в знаковом разряде.

Арифметические действия над характеристиками производятся как над целыми числами. Подразумевается, что запятая фиксируется справа от младшего разряда. При нахождении разности, например, для сравнения вычитаемая характеристика представляется в виде дополнения до 2^7 .

Все характеристики представляются положительными, поэтому в данном случае формулу определения дополнительного кода (2.14) можно записать просто как

$$[p^*]_{\text{доп}} = 2^7 - p^*. \quad (2.15)$$

Однако, определяя сами характеристики, при нахождении дополнительных кодов отрицательных порядков следует пользоваться соотношением вида (2.14). В целом 7-разрядные характеристики отрицательных порядков p вычисляются по формуле

$$p^* = (2^7 + p) + 2^6. \quad (2.16)$$

Например, отрицательному порядку $p_2 = -000\ 0010$ ($p_{10} = -2$) будет соответствовать характеристика

$$\begin{aligned}
 p_2^* &= [1000\ 0000 + (-000\ 0010)] + 100\ 0000 = \\
 &= 111\ 1110 + 100\ 0000 = 011\ 1110.
 \end{aligned}$$

Если эта характеристика будет вычитаться, то согласно (2.15) она преобразуется к виду

$$[p_2^*]_{\text{доп}} = 1000\ 0000 - 011\ 1110 = 100\ 0010.$$

При сравнении характеристик путем нахождения их разности знак последней в общем определяется по значению переноса из старшего разряда (из 1-го, привязываясь к формату числа с плавающей запятой). По значению этого переноса также производится анализ на переполнение ($p^* > 2^7 - 1$) или исчезновение ($p^* \leq 0$) порядка в результате арифметических действий. Правда, в различных моделях машин этот анализ реализуется по-разному, что обусловлено в первую очередь соображениями рациональности построения конкретных схем.

Алгебраические сложение и вычитание мантисс чисел с плавающей запятой производятся иначе, чем чисел с фиксированной запятой. Если знаки обеих мантисс одновременно положительные или отрицательные, то они складываются в прямом коде. Если же знаки разные, то вычитаемая мантисса вначале преобразуется в дополнительный код. Вычитаемой может быть либо отрицательная мантисса (в ЕС-1020), либо всегда мантисса второго операнда (в ЕС-1030), либо мантисса операнда, имеющего больший исходный порядок (в ЕС-1050).

Так как l -разрядные мантиссы чисел, представленных в полуположительной логарифмической форме, по абсолютной величине меньше единицы, то образование их дополнительного кода с учетом изображения знака производится по формуле

$$[A]_{\text{доп}} = 10 + A, \quad (2.17)$$

где 10 — два в двоичной системе счисления и $A = -0^{\cdot}a_{-1}a_{-2} \dots \dots a_{-l}$.

Например, для $A = -0,1011\ 0111$ согласно (2.17)

$$[A]_{\text{доп}} = 10 + (-0,1011\ 0111) = 1^{\cdot}0100\ 1001.$$

Результату сложения мантисс одинакового знака всегда присваивается знак слагаемых. Если же складываются мантиссы разных знаков, то знак результата определяется логическим анализом знаков мантисс перед непосредственным их сложением в сумматоре, видом кода (прямого или дополнительного) полученной суммы и, в случае необходимости, признаком, указывающим, мантисса какого операнда поступила на сложение в дополнительном коде. Последнее имеет место, например, в процессе ЕС-2050, где всегда инвертируется мантисса числа с большим порядком. Знак суммы, имеющей нулевую мантиссу, — всегда положительный.

Десятичные числа. Операции десятичной арифметики производятся над десятичными числами в упакованном формате (рис. 2.2, б). При таком представлении в каждом байте располагаются две десятичные цифры, выраженные в двоично-десятичной форме. Знак числа находится в правой половине крайнего правого

байта цифрового поля. Коды знаков плюса и минуса зависят от того, какой код (ДКОИ или КОИ-8) используется для кодирования алфавита входного языка (см. п. 2.3).

Упакованные десятичные числа, представленные в естественной форме, в оперативной памяти хранятся в прямом коде. Все двоичные тетрады соответствуют естественному двоичному кодированию десятичных цифр 0, 1, . . . , 9. При выполнении же арифметических операций они преобразуются в другой код, упрощающий сложение десятичных операндов в сумматорах. Например, в ЭВМ ЕС-1020, ЕС-1030 и ЕС-1050 нашел применение двоичный код с избытком шесть:

Десятичная цифра	Естественный двоично-десятичный код	Код с избытком 6
0	0000	0110
1	0001	0111
2	0010	1000
3	0011	1001
4	0100	1010
5	0101	1011
6	0110	1100
7	0111	1101
8	1000	1110
9	1001	1111

Если при сложении десятичных цифр получается результат, больший десяти, то необходимо сформировать единицу переноса и передать ее в следующий десятичный разряд. Перенос вырабатывается автоматически, если представить цифры одного из слагаемых в коде с избытком шесть. При этом следует иметь в виду, что когда при сложении таких кодов возникает перенос в следующую тетраду, результат в данном десятичном разряде получается в естественном двоично-десятичном коде. Если в каких-либо тетрадах межтетрадные переносы отсутствуют, то для получения истинного результата из кодов этих тетрад необходимо вычесть избытки 6. Вычитание числа 6 (0110) обычно заменяется прибавлением его дополнения до 2^4 , т. е. 1010. Возникающие при этом межтетрадные переносы не учитываются.

Пример 2.12. Сложить числа: 7 и 18.

Двоично-десятичные коды этих чисел, представленные одинаковым количеством двоичных разрядов, будут иметь вид 0000 0111 и 0001 1000. Представим первое число в коде с избытком 6:

$$\begin{array}{r} + 0000\ 0111 \\ + 0110\ 0110 \\ \hline 0110\ 1101 \end{array}$$

Сложив этот код со вторым числом, получим

$$\begin{array}{r} + 0110\ 1101 \\ + 0001\ 1000 \\ \hline 1000\ 0101 \\ \uparrow \quad \downarrow \\ \text{—} \text{—} \text{—} \end{array}$$

Так как возник перенос p только из первой тетрады, то необходимо скорректировать вторую тетраду, вычтя из нее 6 (прибавив по $\text{mod } 2^4$ дополнение 1010). Таким образом, истинный результат будет

$$\begin{array}{r} + 1000\ 0101 \\ \hline 1010\ 0000 \\ \hline 0010\ 0101, \\ p \leftarrow | \end{array}$$

т. е. получили десятичное число 25.

При вычитании вычитаемое представляется в виде дополнения до 10^n , где n — количество десятичных разрядов в наиболее длинном из участвующих в операции операнде. Таким образом, вычитание заменяется сложением прямого кода положительного числа с дополнением отрицательного. Дополнение числа определяется путем инвертирования значений двоичных разрядов в тетрадах с прибавлением единицы в младший разряд крайней справа тетрады. Например, дополнением числа $-0010\ 0111$ (-27) до 10^2 будет код 1101 1001. Такое дополнение получается с потетрадным избытком 6. Поэтому, если при сложении нет переносов из каких-либо тетрад, то результат в этих тетрадах корректируется вычитанием избытков. Как в рассмотренном выше примере, вычитание реализуется путем прибавления по $\text{mod } 2^4$ дополнения 1010.

Представление данных в форматах, разрядность которых кратна 8-разрядным байтам, и применение дополнительного кода при вычитании оказались удобными для построения ряда машин различной производительности. Высокопроизводительные модели ЭВМ Единой системы (ЕС-1050 и ЕС-1060) для выполнения арифметических и логических операций имеют устройства в основном параллельного действия. Например, в ЕС-1050 арифметическое устройство параллельного действия для операций над числами с фиксированной и плавающей запятой рассчитано на оперирование двойными словами. Модели меньшей производительности, такие как ЕС-1010 и ЕС-1020, могут осуществлять только последовательно-параллельную (побайтовую) переработку информации. С целью экономии аппаратуры в них, например, сложение n -разрядных слагаемых A и B выполняется в виде серии $\frac{n}{8} = k$ последовательных сложений байт A'_i и B'_i ($i = 1, 2, \dots, k$), представленных параллельными кодами. Сложение начинается с младших (крайних правых) байт. При сложении каждой пары байт вырабатывается и запоминается межбайтовый перенос, который учитывается при сложении следующей пары и т. д. В этих моделях поэтапно выполняются также и все остальные операции. На каждом этапе обрабатывается одновременно не более 2 байт.

2.6. СТРУКТУРА НАБОРА ОПЕРАЦИЙ ПРОЦЕССОРА. КОДЫ ОПЕРАЦИЙ. АДРЕСАЦИЯ ОПЕРАНДОВ В КОМАНДАХ

В машинах Единой системы, как и во всех цифровых машинах с программным управлением, процесс переработки информации в итоге складывается из выполнения последовательности арифметических, логических и других операций. Тип каждой исполняемой операции, количество и местоположение операндов, указание, куда поместить результат, и др. необходимая управляющая информация определяются командой, представленной в виде, удобном для восприятия схемами управления машины (в виде условного двоичного кода), или, как говорят, на машинном языке. На этом языке коды команд программы решаемой задачи формируются автоматически в процессе трансляции ее машиной с входного языка, на котором она была составлена программистом.

Система команд ЕС ЭВМ имеет обширный набор, обеспечивающий выполнение более 140 видов операций. Она согласована с получившей широкое распространение системой команд машин серии 360 американской фирмы IBM. В приложении (см. табл. 10) приведены 142 основные операции этого набора. Там же даны их английские названия и условные мнемонические обозначения, принятые для сокращенной записи машинных команд при программировании на языке АССЕМБЛЕРА. Команды в такой условной записи будем называть инструкциями, указывающими транслятору, как сформировать команду на машинном языке.

В машинах Единой системы предусмотрена возможность преобразования двоичных чисел, представленных в естественной и в полулогарифмической формах, десятичных чисел — в естественной форме. Кроме того, любая двоично кодированная информация может перерабатываться, будучи размещенной в полях переменной длины. Поэтому в наборе команд содержатся команды для выполнения арифметических и логических операций над числами с фиксированной запятой, над числами с плавающей запятой, над десятичными числами и логических операций над полями данных.

Для оперирования двоичными числами различных форматов с фиксированной и плавающей запятой предусмотрены соответствующие группы команд (см. приложение, табл. 10). Имеется также обширный набор команд, обеспечивающих выполнение операций:

- передачи управления;
- переключения состояний процессора, оперативной памяти, а также осуществления связи между машинами в многопроцессорных системах;
- управления вводом-выводом информации;
- контроля и диагностики.

Эти операции можно отнести к операциям управления, обеспечивающим требуемые очередности исполнения как отдельных

команд, так и программ в целом, а также организующим совместную работу различных устройств, например, процессора, оперативной памяти и устройств ввода-вывода и их автоматический контроль. Особенности выполнения ряда основных арифметических и логических операций, а также организация управления процессами обмена и переработки информации с помощью этих команд рассматриваются в последующих главах.

Каждая команда содержит двоичный код, определяющий, какая операция должна быть выполнена (код операции — КОп), а также двоичные коды, либо непосредственно являющиеся адресами операндов, участвующих в операции, либо указывающие, как сформировать эти адреса. В командах управления коды адресов определяют, куда должно быть передано управление, номера каналов и устройств ввода-вывода и т. п.

Во всех командах на код операции отводится один байт (см. приложение, табл. 10). Адресная же часть может иметь различную длину: от 1 до 5 байт. Это объясняется особенностью иерархической структуры оперативной памяти машин.

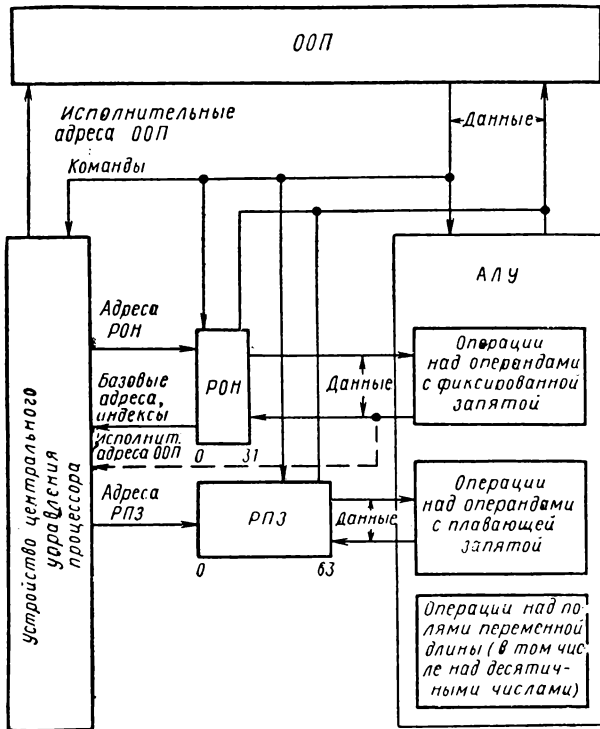


Рис. 2.9. Обобщенная структура информационных связей устройств центрального процессора с основной и регистровой оперативными памятьями

Кроме основной оперативной памяти (ООП) большой емкости, во всех моделях ЭВМ Единой системы имеется еще оперативная память малой емкости (рис. 2.9). Она состоит из 16 прямо адресуемых регистров общего назначения (РОН) и 4 регистров для операндов с плавающей запятой (РПЗ). В различных моделях регистровая память реализуется по-разному. В ЕС-1050 она выполнена на триггерах и конструктивно расположена в блоке центрального управления процессора. В ЕС-1030 РОН и РПЗ входят в состав блока местной памяти емкостью 64 слова, построенной на магнитных пленках. В ЕС-1020 это специальная область основной оперативной памяти.

Одной из причин применения специальной регистровой памяти явилось стремление повысить среднюю скорость выполнения операций. Как показали исследования, наличие указанных выше регистров примерно в 2 раза снижает частоту обращения к основной оперативной памяти. Поэтому там, где малая оперативная память имеет меньший цикл обращения, по сравнению с основной, получается выигрыш в быстродействии машины.

Регистры общего назначения — 32-разрядные. Они предназначены для временного запоминания исходных операндов и результатов арифметических операций с фиксированной запятой, логических операций над данными фиксированного формата, а также для хранения базовых адресов и индексов (констант модификации адресов). Как будет показано ниже, последние два типа величин используются при формировании исполнительных адресов основной оперативной памяти.

Регистрам общего назначения присвоены номера от 0 до 15 (рис. 2.10). Эти номера задаются в командах 4-разрядными двоичными адресами R . При выполнении некоторых операций два смежных регистра используются совместно, позволяя оперировать двойными словами. В этом случае указывается адрес регистра, хранящего старшие разряды операнда. Этот адрес должен быть обязательно четным. Младшие разряды автоматически записываются или выбираются из следующего по порядку регистра с нечетным адресом.

Регистры для операндов с плавающей запятой имеют номера 0, 2, 4 и 6. Каждый из этих регистров рассчитан на 64-разрядное слово. В командах эти регистры также указываются 4-разрядными

Номер регистра	Двоичный адрес R	РОН	РПЗ
0	0000	← 32 разряда →	← 64 разряда →
1	0001		
2	0010		
3	0011		
4	0100		
5	0101		
6	0110		
7	0111		
8	1000		
9	1001		
10	1010		
11	1011		
12	1100		
13	1101		
14	1110		
15	1111		

Рис. 2.10. Адресация регистров общего назначения и регистров для операндов с плавающей запятой

адресами R . Однако конкретный тип регистра РОН или РПЗ определяется исполняемой операцией.

Адресация основной оперативной памяти, максимальная емкость которой в более быстродействующих моделях может достигать 16777216 байт, осуществляется несколько сложнее. Для того чтобы иметь возможность адресоваться к любому из этого количества байт, код адреса должен иметь $\log_2 16777216 = 24$ двоичных разрядов. Помещать адреса такой разрядности в команды нерационально, так как это потребовало бы применения команд, имеющих чрезмерно большое количество разрядов. Программы, составленные из таких многоразрядных команд, вызвали бы нерациональное расходование емкости оперативной памяти.

Для сокращения в командах требуемой разрядности кодов адресов основной оперативной памяти, обращение к ней организуется путем применения, так называемой, *относительной* адресации. При такой адресации оперативная память условно принимается состоящей из секций по 4096 байт. Начало каждой секции определяется адресом ее первого байта. Этот адрес называется *базовым*. Секции можно размещать в произвольных местах оперативной памяти, начиная с любого байта. Поэтому для задания всех возможных базовых адресов отводится 24 двоичных разряда.

В пределах секции адрес каждого байта относительно базового определяется 12-разрядным двоичным числом, называемым *смещением* и обозначаемым буквой D . С учетом базы и смещения исполнительный адрес $A_{исп}$, по которому производится обращение к оперативной памяти, вычисляется как сумма по mod 2^{24} кода базового адреса и кода смещения. Складываются эти коды как положительные целые числа. Сложение может происходить либо в специальном сумматоре адреса центрального устройства управления (рис. 2.9), что имеет место, например, в модели ЕС-1050, либо в блоке арифметико-логического устройства (АЛУ) процессора, где выполняются операции над операндами с фиксированной запятой. Арифметико-логическое устройство для этой цели используется, например, в моделях ЕС-1020 и ЕС-1030. На рис. 2.9 цепь получения исполнительных адресов из АЛУ показана пунктиром.

По количеству команд и объему данных программа может размещаться либо в пределах одной секции или занимать несколько секций. Если она не выходит за пределы одной секции, то для формирования исполнительных адресов основной памяти во всех командах достаточно будет одного базового адреса. Этот адрес помещается в один из регистров общего назначения, который в этом случае получает название регистра базы. Базовый адрес занимает с 8-й по 31-ю (слева направо) разрядные позиции регистра базы. Например, код базового адреса 4756327_{10} , которому соответствует шестнадцатеричное число 489367_{16} будет размещаться в i -м регистре, как показано на рис. 2.11.

Для указания адреса основной оперативной памяти, по которому должно быть произведено обращение, в адресной части команды помещаются 12-разрядный двоичный код смещения D и 4-разрядный код адреса B . Последний определяет номер РОН, где находится базовый адрес. В процессе выполнения команды исполнительный адрес ООП $A_{исп}$ автоматически вычисляется по формуле

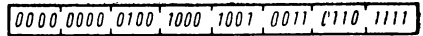


Рис. 2.11. Размещение базового адреса в регистре общего назначения

$$A_{исп} = (B) + D, \quad (2.18)$$

где B , заключенное в круглые скобки, обозначает содержимое регистра базы, имеющего адрес B .

Если смещение $D = 1021_{10} = 3FD_{16} = 11\ 1111\ 1101_2$ и в регистре с адресом B находится базовый адрес

$$4756327_{10} = 489367_{16} = 100\ 1000\ 1001\ 0011\ 0110\ 0111_2,$$

то двоичный исполнительный адрес $A_{исп\ 2}$ будет равен

$$\begin{array}{r} + \quad 100\ 1000\ 1001\ 0011\ 0110\ 0111 \\ \quad \quad \quad \quad \quad \quad \quad \quad 11\ 1111\ 1101 \\ \hline 100\ 1000\ 1001\ 0111\ 0110\ 0100, \end{array}$$

что соответствует 489764_{16} или 4757348_{10} .

Если программа и данные располагаются в нескольких секциях памяти, то в регистрах общего назначения приходится хранить несколько базовых адресов. При этом в командах, в зависимости от того, какой требуется базовый адрес, указывается адрес соответствующего регистра базы. В качестве таковых могут использоваться любые РОН, кроме нулевого. Если в команде указывается $B = 0$, то предполагается, что базовый адрес равен нулю независимо от содержимого регистра с номером 0.

Кроме сокращения разрядности адресных частей команд, применение относительной адресации упрощает при необходимости перемещение программ или их сегментов в оперативной памяти, задание входных, выходных и рабочих областей памяти. Для этого достаточно в соответствующие регистры базы поместить нужные базовые адреса. В командах же программы в пределах каждой предполагаемой секции указываются лишь смещения, которые могут принимать значения от 0 до 4095. Распределение оперативной памяти и назначение базовых адресов при вводе рабочей программы осуществляется автоматически управляющей программой операционной системы.

При выполнении операций над операндами фиксированной длины количество байт, записываемых в память или выбираемых из нее вместе с адресуемым, как было указано в п. 2.3, определяется неявно кодом операции. Так как при оперировании полусловами,

словами или двойными словами адресом в команде указывается самый левый байт, имеющий наименьший адрес, то вместе с этим байтом, например, при считывании, выбирается определенное количество байт, имеющих следующие в порядке возрастания адреса. В отличие от этого в командах, управляющих выполнением операций над полями переменной длины, кроме адреса начала поля, необходимо указывать его длину. Она задается специальным кодом в адресной части команды.

Использование рассмотренных способов адресации, объясняющихся наличием двух видов оперативной памяти, обусловило применение команд различных форматов. Набор команд ЕС ЭВМ в основном состоит из двухадресных команд пяти форматов, показанных на рис. 2.12. Адреса указывают, откуда должны быть взяты исходные операнды. Кроме того, в большинстве операций подразумевается, что результат помещается на место первого операнда. В зависимости от местоположения и способов адресации исходных операндов, форматы команд условно обозначаются форматными кодами: RR, RX, RS, SI и SS.

Форматный код RR обозначает операции типа регистр—регистр; коды RX, RS и SI — операции, в которых один из операндов находится в основной оперативной памяти, а код SS — операции типа память—память.

Меньше всего разрядов (одно полуслово) требуется для размещения кодов команд типа RR. На рис. 2.12 в формате команды этого типа R_1 и R_2 обозначают соответственно 4-разрядные

Условные обозначения формата	Формат и структура команды	Значение разрядов 0-20 и 1-20 Коп																																																												
RR	Коп R_1 R_2	00																																																												
RX	Коп R_1 X_2 B_2 D_2	01																																																												
RS	Коп R_1 R_3 B_2 D_2	10																																																												
SI	Коп I_2 B_1 D_1																																																													
SS	Коп L_1 L_2 B_1 D_1 B_2 D_2	11																																																												
	Коп L B_1 D_1 B_2 D_2																																																													
<table border="1" style="width: 100%; text-align: center;"> <tr> <td colspan="4">Полуслово 1</td> <td colspan="4">Полуслово 2</td> <td colspan="4">Полуслово 3</td> </tr> <tr> <td>байт 1</td><td>байт 2</td><td>байт 3</td><td>байт 4</td> <td>байт 5</td><td>байт 6</td><td>байт 7</td><td>байт 8</td> <td>байт 9</td><td>байт 10</td><td>байт 11</td><td>байт 12</td> <td>байт 13</td><td>байт 14</td><td>байт 15</td><td>байт 16</td> <td>байт 17</td><td>байт 18</td><td>байт 19</td><td>байт 20</td> <td>байт 21</td><td>байт 22</td><td>байт 23</td><td>байт 24</td> <td>байт 25</td><td>байт 26</td><td>байт 27</td><td>байт 28</td> <td>байт 29</td><td>байт 30</td><td>байт 31</td><td>байт 32</td> <td>байт 33</td><td>байт 34</td><td>байт 35</td><td>байт 36</td> <td>байт 37</td><td>байт 38</td><td>байт 39</td><td>байт 40</td> <td>байт 41</td><td>байт 42</td><td>байт 43</td><td>байт 44</td> <td>байт 45</td><td>байт 46</td><td>байт 47</td><td>байт 48</td> </tr> </table>			Полуслово 1				Полуслово 2				Полуслово 3				байт 1	байт 2	байт 3	байт 4	байт 5	байт 6	байт 7	байт 8	байт 9	байт 10	байт 11	байт 12	байт 13	байт 14	байт 15	байт 16	байт 17	байт 18	байт 19	байт 20	байт 21	байт 22	байт 23	байт 24	байт 25	байт 26	байт 27	байт 28	байт 29	байт 30	байт 31	байт 32	байт 33	байт 34	байт 35	байт 36	байт 37	байт 38	байт 39	байт 40	байт 41	байт 42	байт 43	байт 44	байт 45	байт 46	байт 47	байт 48
Полуслово 1				Полуслово 2				Полуслово 3																																																						
байт 1	байт 2	байт 3	байт 4	байт 5	байт 6	байт 7	байт 8	байт 9	байт 10	байт 11	байт 12	байт 13	байт 14	байт 15	байт 16	байт 17	байт 18	байт 19	байт 20	байт 21	байт 22	байт 23	байт 24	байт 25	байт 26	байт 27	байт 28	байт 29	байт 30	байт 31	байт 32	байт 33	байт 34	байт 35	байт 36	байт 37	байт 38	байт 39	байт 40	байт 41	байт 42	байт 43	байт 44	байт 45	байт 46	байт 47	байт 48															
Разряды	0		7 8 11 12				15 16 19 20				31 32				33 34				47																																											

Рис. 2.12. Форматы, их условные обозначения и структуры команд

адреса регистров 1-го и 2-го операндов. Как было указано выше, в данном случае применяется прямая адресация.

Коды команд типа RX, RS и SI имеют длину, соответствующую двум полусловам. Разница между этими командами состоит в следующем.

В формате RX первый операнд находится в регистре R_1 , а второй — в основной оперативной памяти по адресу, определяемому суммой кода смещения D_2 с содержимыми регистров базы и индекса, имеющими 4-разрядные адреса соответственно B_2 и X_2 , т. е.

$$A_{2 \text{ исп}} = (X_2) + (B_2) + D_2. \quad (2.19)$$

Все три кода складываются как положительные целые числа по mod 2^{24} .

Модификация адресов путем прибавления индексов используется при организации циклических вычислительных процессов. В машинах Единой системы индекс, как и базовый адрес, к началу исполнения команды должен находиться в регистре общего назначения. Регистр, в котором хранится индекс, получает название регистра индекса.

Команды формата RS также относятся к типу регистр-память. Однако модификация адреса ООП здесь не предусмотрена. Поэтому разряды с 12-го по 15-й отведены для представления третьего адреса. Не во всех командах данного типа используется поле адреса R_3 , однако в общем случае их можно рассматривать как трехадресные. Примером трехадресной команды является команда ЗАГРУЗИТЬ ГРУППУ (LM). По этой команде данные, находящиеся в основной памяти в области, начинающейся с адреса $A_{2 \text{ исп}} = (B_2) + D_2$, загружаются в РОН, начиная с R_1 по R_3 .

В командах формата SI, как и в предыдущих двух типах, имеется один относительный адрес основной оперативной памяти, определяемый смещением D_1 и адресом регистра базы B_1 . Второй же операнд в этом формате является непосредственно адресуемым. Он имеет фиксированную длину (1 байт) и содержится в разрядах с 8-го по 15-й код команды. Примером команды формата SI является команда ПЕРЕСЛАТЬ (MVI). Она помещает непосредственно адресуемый операнд в основную память, в ячейку с адресом $A_{1 \text{ исп}} = (B_1) + D_1$.

Операции над операндами, располагающимися в основной памяти в полях переменной длины, выполняются по командам формата SS. Этот формат, имеющий длину 3 полуслова, содержит два относительных адреса основной памяти: B_1, D_1 и B_2, D_2 . Каждый относительный адрес занимает полуслово. В зависимости от способа указания длины полей имеются две разновидности команд формата SS (рис. 2.12).

Команды операций над десятичными числами содержат в разрядах 8—15 два поля, в которых помещаются 4-разрядные указатели L_1 и L_2 длины операндов. Поэтому операнды — десятичные числа могут быть разной длины и занимать поля, не превышающие

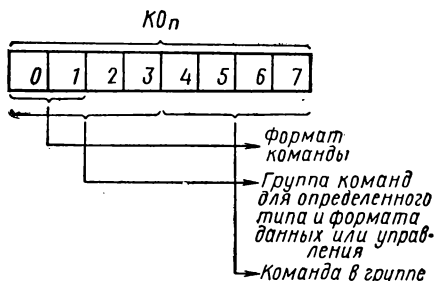


Рис. 2.13. Назначение составляющих частей операции

симально возможная длина каждого поля составляет 256 байт.

В рассмотренных выше типах форматов коды X_i , B_i и D_i могут быть равны нулю. Это указывает на равенство нулю индекса, базы или смещения. Адрес всегда формируется до выполнения операции. Поэтому один и тот же общий регистр может быть использован в команде и для формирования адреса, и как операнд.

Полную информацию о типе операции, которая должна быть выполнена, о формате самой команды и данных, о типе данных и их местонахождении (виде памяти) несет код операции. Во всех форматах команд код операции находится в разрядах с 0-го по 7-й (рис. 2.12). Из них разряды 0-й и 1-й (см. рис. 2.13) служат для указания формата команды и, следовательно, общей характеристики местонахождения данных. Соответствие значений содержимых этих разрядов форматам команд, показанное справа на рис. 2.12, нетрудно установить, анализируя таблицу кодов операций (см. приложение, табл. 10). Значения следующих двух разрядов (2-го и 3-го) совместно с первыми двумя (0-м и 1-м) определяют группу команд, объединенных типом и форматом данных (слова или полуслова с фиксированной запятой, короткие или длинные операнды с плавающей запятой и т. п.), а также специфичностью выполняемых операций. Последние четыре разряда 4—7 определяют конкретную команду в группе.

Одной из особенностей рассмотренных форматов и структуры команд является то, что благодаря их применению оказалось возможным решение проблемы программной совместимости моделей ЭВМ Единой системы, различных по быстродействию и имеющих различные емкости основной оперативной памяти. Инвариантность форматов команд по отношению к емкости основной памяти обеспечивается применением относительной адресации. При этом способе, независимо от емкости памяти, если производится обращение к ней, в командах указываются фиксированные по длине коды смещения и адреса регистров базы. Предельный же объем памяти находит свое отражение в максимально возможном для модели базовом адресе.

16 байт каждый. В командах логических операций над полями переменной длины оба поля берутся равными. Поэтому в таких командах используется один указатель длины L . При 8 разрядах, отводимых на код длины, $L_{\max} \leq 255$. Так как L указывает количество байт в дополнение к каждому из адресуемых $A_{i \text{ исп}} = (B_1) + D_i$ и $A_{2 \text{ исп}} = (B_2) + D_2$, то мак-

Кратность форматов команд полуслову и указание их длины в кодах операций дает возможность любой модели воспринимать команды независимо от того, сколько байт за одно обращение позволяет выбирать ее оперативная память. Указатель формата, например, в ЕС-1020 позволяет выбирать команды из ООП и обрабатывать их по частям (по 2 байта).

Применение команд различных форматов и в особенности с короткими адресами, относящимися к регистровой памяти, кроме отмеченного выше сокращения частоты обращения к основной оперативной памяти, способствует уменьшению длин программ. Это позволяет более компактно размещать программы в ООП, экономичнее расходуя емкость памяти.

ПРИНЦИПЫ ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ ОПЕРАЦИЙ В ПРОЦЕССОРАХ МОДЕЛЕЙ

3.1. ОБЩИЕ СВЕДЕНИЯ О СТРУКТУРАХ И ОСОБЕННОСТЯХ РЕАЛИЗАЦИИ МАШИННЫХ АЛГОРИТМОВ ОПЕРАЦИЙ

В электронных вычислительных машинах Единой системы как цифровых машинах с программным управлением циклы выполнения арифметических и логических операций слагаются из следующих действий. Вначале из основной оперативной памяти в процессор выбирается очередная команда. Она расшифровывается, и в зависимости от ее формата формируются исполнительные адреса операндов. По этим адресам из регистровой (местной, локальной) или основной оперативной памяти в арифметико-логическое устройство (АЛУ) процессора вызываются операнды, где над ними производятся необходимые действия. Каждая арифметическая и логическая операция заканчивается отсылкой результата в регистровую память (РОН или РПЗ) или в ячейку ООП по первому адресу команды. Процессы выборки команд, операндов и выполнения действий в АЛУ сопровождаются обязательной проверкой правильности кодов команд, адресов и данных. Организация такого контроля позволяет отличать ошибки программ от машинных сбоев.

В конце большинства арифметических и логических операций производится анализ результата и выработка его признака. Последний сохраняется до конца исполнения следующей команды, которая, в зависимости от особенностей производимых ею действий, может заменить его либо новым значением, либо оставить без изменения.

Во всех моделях ЭВМ Единой системы управление выполнением операций организуется по асинхронному принципу. Время выполнения операции зависит от ее типа, формата команды, длины операндов и их значения, а также от быстродействия схем и организации структуры модели, разрядности процессора и принятого в модели количества байт, одновременно выбираемых из оперативной памяти (от ширины выборки) и обрабатываемых в операционной части АЛУ.

Среди перечисленного особое место занимают разрядность процессора и ширина выборки из оперативной памяти, которые

являются одними из основных параметров, составляющих технические данные моделей. То, что в ЕС-1020 за одно обращение к оперативной памяти выбирается или записывается 2 байта, в ЕС-1030 — 4 байта, а в ЕС-1050 — 8 байт, само по себе является причиной ряда особенностей выполнения операций в каждой из перечисленных моделей. Например, в ЕС-1020 4- и 6-байтовые коды команд приходится выбирать соответственно за 2 или 3 обращения к памяти. Аналогично и операнды, как фиксированной, так и переменной длины, представляется возможным выбирать в процессор всего лишь по 2 байта за одно обращение к ОП. В противоположность этому в ЕС-1050 за одно обращение к памяти выбирается участок программы, в котором может содержаться до четырех команд формата RR или до двух команд формата RX и т. п. В самом процессоре из этой последовательности производится окончательный выбор для исполнения очередной команды.

В модели ЕС-1020 АЛУ рассчитано на побайтовую обработку данных во всех операциях. В АЛУ ЕС-1030 при выполнении арифметических и логических операций над двоичными данными фиксированной длины и над десятичными числами действия одновременно производятся над 4 байтами. В логических же операциях над полями переменной длины данные обрабатываются побайтно. В АЛУ модели ЕС-1050 блоки для выполнения арифметических и логических операций над данными фиксированной длины рассчитаны на параллельную обработку 64-разрядных операндов. Кроме того, имеется специальный блок для арифметических действий над десятичными числами и для выполнения логических операций над полями переменной длины. Он может осуществлять только побайтовую переработку информации.

Особенностям построения и функционирования процессоров, обусловленным разрядностью используемых в них устройств и блоков для преобразования команд и данных, посвящены гл. 6 и 7. В данной же главе рассматриваются лишь способы выполнения арифметических и логических действий, положенные в основу машинных алгоритмов операций различных моделей. При этом основное внимание уделяется тому, что способствует либо упрощению побайтового преобразования информации, либо достижению высокой скорости ее переработки, либо просто отражает принципиальные особенности реализуемых в моделях машинных алгоритмов.

3.2. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ НАД ДВОИЧНЫМИ ЧИСЛАМИ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

В ЕС ЭВМ такие операции выполняются над *целыми* двоичными числами, представленными в естественной форме. Это могут быть адреса, константы модификации (индексы), величины, отражающие состояния программных счетчиков, либо просто данные, выраженные в форме с фиксированной запятой. Отрицательные

числа хранятся в оперативной памяти и участвуют в операциях в дополнительном коде. Исходные операнды могут либо оба располагаться в регистрах общего назначения, либо один — в регистре, а другой — в основной оперативной памяти. Поэтому в наборе команд для одинаковых операций имеются как команды формата RR, так и RX.

Команды формата RX имеются в двух вариантах. Один из них предназначен для операций, при выполнении которых выбираемый из основной памяти операнд является полным словом, а в другом — полусловом. Первый операнд, всегда расположенный в регистре, имеет длину слова. Исключение составляют операции деления и логического сдвига двойного слова. При выполнении их в первом случае 64-разрядное делимое, а во втором — сдвигаемое берутся из двух смежных регистров. Выбираемые из основной памяти полуслова автоматически удлиняются до 32 разрядов и участвуют в операциях в виде полных слов.

Для действий над операндами с фиксированной запятой имеется обширный набор операций. Кроме арифметических и логических операций, предусмотрены команды операций: загрузки регистров общего назначения информацией из основной оперативной памяти, записи информации в ООП, изменения знаков чисел, их сдвига, преобразования из десятичной системы счисления в двоичную и обратно.

Результаты действий, как правило, помещаются в регистровую память на место первых операндов. После сложения, вычитания, сдвига и логических операций вырабатывается признак результата.

Сложение и вычитание. По командам СЛОЖИТЬ и ВЫЧЕСТЬ операнды, в зависимости от модели, либо целиком, либо по частям, вызываются из оперативных запоминающих устройств во входные регистры арифметико-логического устройства процессора. Операции выполняются над числами, выравненными по младшим разрядам, правее которых считается фиксированной запятой.

Операнды — положительные целые числа, занимающие только часть разрядов слова, имеют во всех разрядах левее старшей значащей цифры нули, а отрицательные — единицы. При необходимости увеличение длин операндов в сторону старших разрядов осуществляется простым добавлением разрядов, содержащих согласно знаку нули при положительном числе и единицы — при отрицательном.

Процесс выполнения команды ВЫЧЕСТЬ сводится к алгебраическому сложению, для чего знак вычитаемого, поступившего в арифметико-логическое устройство, искусственно меняется на обратный. Изменение знака осуществляется взятием дополнения для кода вычитаемого. Для этого поразрядно, включая знаковый разряд, инвертируется слово второго операнда и к младшему разряду прибавляется единица. Таким образом, как при сложении, так и при вычитании нет необходимости производить предварительный анализ знаков исходных операндов.

В моделях ЕС-1030 и ЕС-1050 сложение двоичных чисел производится в сумматорах арифметико-логических устройств. В ЕС-1020 оно выполняется в арифметико-логическом блоке (АЛБ) процессора, не имеющем явно выраженного сумматора. На входы суммирующих схем положительные числа поступают в прямом коде, а отрицательные — в дополнительном. Знаковые разряды участвуют в сложении наравне с остальными разрядами слагаемых.

По окончании полного процесса суммирования производится анализ результата и выработка его признака. В анализ входит определение: равен, меньше или больше нуля результат и не возникло ли переполнение. Признак результата (ПР) представляется двумя двоичными разрядами. При сложении и вычитании значения их вырабатываются в соответствии с приведенными ниже данными.

Результат операции	Признак результата
0	00
< 0	01
> 0	10
Переполнение	11

Переполнение возникает, если в результате сложения 32-разрядных операндов получаются числа, большие, чем

$$2^{31} - 1 = 2\ 147\ 483\ 647_{10} = 01111111\ 11111111\ 11111111\ 11111111_2,$$

или меньшие, чем

$$-2^{31} = -2\ 147\ 483\ 648_{10} = 10000000\ 00000000\ 00000000\ 00000000_2$$

Это обнаруживается путем анализа переносов в знаковый разряд и из него. При отсутствии переполнения либо не должно быть переносов, либо они должны быть оба вместе. В случае переполнения переносы из знакового разряда и в знаковый не совпадают. Если не наложен запрет на прерывание по переполнению, то появление признака ПР-11 вызывает прерывание исполняемой программы.

В п. 2.5 примерами 2.8—2.11 были пояснены собственно процессы сложения целых двоичных чисел с использованием дополнительного кода для представления отрицательных чисел. Здесь рассмотрим примеры, поясняющие действия над числами по командам СЛОЖИТЬ и ВЫЧЕСТЬ и получение результатов, соответствующих различным признакам ПР.

Пример 3.1. Пусть требуется выполнить операции алгебраического сложения над следующими целыми числами.

$$\begin{array}{l}
 1) \left. \begin{array}{l} [+5]_{\text{пр}} \\ [-5]_{\text{доп}} \end{array} \right\} \oplus \left\{ \begin{array}{l} 0.000\ 0101 \\ 1.111\ 1011 \end{array} \right\} \\
 \oplus \left\{ \begin{array}{l} 1.111\ 1110 \\ \underline{1\ 1\ 1\ 1\ 1\ 1\ 1\ 1} \end{array} \right\} \begin{array}{l} \text{— поразрядная сумма слагаемых по mod 2} \\ \text{— переносы} \end{array} \\
 \overline{[0]}_{\text{пр}} \quad 0.000\ 0000 \quad \text{— результат, получившийся равным нулю; ПР-00}
 \end{array}$$

Примечание: $+125_{10} = 0111\ 1101_2$

$$\begin{array}{l}
 2) \left. \begin{array}{l} [+5]_{\text{пр}} \\ [-125]_{\text{доп}} \end{array} \right\} \oplus \left\{ \begin{array}{l} 0.000\ 0101 \\ 1.000\ 0011 \end{array} \right\} \\
 \oplus \left\{ \begin{array}{l} 1.000\ 0110 \\ \underline{1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1} \end{array} \right\} \begin{array}{l} \text{— поразрядная сумма слагаемых по mod 2} \\ \text{— переносы} \end{array} \\
 \overline{[-120]}_{\text{доп}} \quad 1.000\ 1000 \quad \text{— отрицательный результат в дополнительном коде; ПР-01}
 \end{array}$$

3)
$$\left. \begin{array}{r} [+125]_{\text{пр}} \\ [-5]_{\text{доп}} \end{array} \right\} \oplus \begin{array}{r} 0.111\ 1101 \\ 1.111\ 1011 \\ \hline 1.000\ 0110 \\ \underline{\overline{111}}\ 111\ 111 \\ \hline 0.111\ 1000 \end{array} \oplus \left\{ \begin{array}{l} 1.000\ 0110 \text{ — поразрядная сумма слагаемых по mod 2} \\ \underline{\overline{111}}\ 111\ 111 \text{ — переносы} \end{array} \right.$$

$$\overline{[+120]_{\text{пр}}} \quad 0.111\ 1000 \text{ — положительный результат в прямом коде; ПР-10}$$

4)
$$\left. \begin{array}{r} [+5]_{\text{пр}} \\ [+125]_{\text{пр}} \end{array} \right\} \oplus \begin{array}{r} 0.000\ 0101 \\ 0.111\ 1101 \\ \hline 0.111\ 1000 \\ \underline{\overline{01}}\ 111\ 101 \\ \hline 1.000\ 0010 \end{array} \oplus \left\{ \begin{array}{l} 0.111\ 1000 \text{ — поразрядная сумма слагаемых по mod 2} \\ \underline{\overline{01}}\ 111\ 101 \text{ — переносы} \end{array} \right.$$

$$\overline{[+130]_{\text{пр}}} \quad 1.000\ 0010 \text{ — результат; ПР-11. Возникло переполнение, так как перенос в знаковый разряд есть, а из знакового — нет!}$$

5)
$$\left. \begin{array}{r} [-5]_{\text{доп}} \\ [-125]_{\text{доп}} \end{array} \right\} \oplus \begin{array}{r} 1.111\ 1011 \\ 1.000\ 0011 \\ \hline 0.111\ 1000 \\ \underline{\overline{110}}\ 000\ 011 \\ \hline 0.111\ 1110 \end{array} \oplus \left\{ \begin{array}{l} 0.111\ 1000 \text{ — поразрядная сумма слагаемых по mod 2} \\ \underline{\overline{110}}\ 000\ 011 \text{ — переносы} \end{array} \right.$$

$$\overline{[-130]_{\text{доп}}} \quad 0.111\ 1110 \text{ — результат; ПР-11. Возникло переполнение, так как есть перенос из знакового разряда, а в знаковый — нет!}$$

Пример 3.2. Пусть требуется выполнить операции алгебраического вычитания по команде ВYЧЕСТЬ.

1)
$$\overbrace{[+125]_{\text{пр}}} - \overbrace{[+5]_{\text{пр}}} = \overbrace{[+125]_{\text{пр}}} + \overbrace{[-5]_{\text{доп}}} = \overbrace{[+120]_{\text{пр}}}$$

$$0.111\ 1101 - 0.000\ 0101 = \oplus \left\{ \begin{array}{l} 0.111\ 1101 + 1.111\ 1011 \\ 1.111\ 1011 \leftarrow \text{переносы} \\ \hline 1.000\ 0110 \text{ — поразрядная сумма по mod 2} \\ \underline{\overline{111}}\ 111\ 111 \text{ — переносы} \\ \hline 0.111\ 1000 \text{ — результат; ПР-10} \end{array} \right.$$

2)
$$\overbrace{[+125]_{\text{пр}}} - \overbrace{[-5]_{\text{доп}}} = \overbrace{[+125]_{\text{пр}}} + \overbrace{[+5]_{\text{пр}}} = \overbrace{[+130]_{\text{пр}}}$$

$$0.111\ 1101 - 1.111\ 1011 = \oplus \left\{ \begin{array}{l} 0.111\ 1101 + 0.000\ 0101 \\ 0.000\ 0101 \leftarrow \text{переносы} \\ \hline 0.111\ 1000 \text{ — поразрядная сумма по mod 2} \\ \underline{\overline{01}}\ 111\ 101 \text{ — переносы} \\ \hline 1.000\ 0010 \text{ — результат; ПР-11} \\ \text{Переполнение!} \end{array} \right.$$

Результат в том коде, в каком он получился, отсылается по адресу R_1 .

В различных моделях ЭВМ единой системы коды операндов по-разному подаются на сложение. В ЕС-1020 схема суммирования 8-разрядная, поэтому 32-разрядные операнды складываются по-байтно, последовательно-параллельно, начиная с младших (правых) байт. Межбайтовый перенос, возникший при сложении 2 байт, запоминается и учитывается при сложении следующей

(старшей) пары байт. В ЕС-1030 и ЕС-1050, имеющих соответственно 32- и 64-разрядный сумматоры, осуществляются полностью параллельные процессы сложения.

Умножение. В наборе команд ЕС ЭВМ имеются команды, дающие возможность умножать 32-разрядные операнды — множимые на множители как 32-разрядные, так и 16-разрядные. Для первого варианта предусмотрены команды форматов типа RR и RX для второго — только RX, т. е. 16-разрядный операнд — множитель может быть взят только из основной оперативной памяти.

В результате перемножения 32-разрядных операндов получается 64-разрядное произведение. Оно отсылается на место первого операнда — множимого в два смежных регистра общего назначения. Старшие разряды помещаются в регистр с четным номером, а младшие в соседний нечетный в порядке возрастания адресов. Переполнения при таком умножении не может быть. При умножении на 16-разрядный множитель длина его автоматически увеличивается до 32 разрядов путем распространения значения знакового разряда на старшие разряды формируемого таким образом полного слова. Однако в качестве результата используются только 32 младших разряда произведения, которые отсылаются на место первого операнда. В команде умножения на полуслово адрес первого операнда может быть как четным, так и нечетным. Старшие разряды результата умножения на полуслово не используются и на значимость не проверяются. Переполнение в данном случае также не имеет места. Поэтому результаты выполнения операций умножения чисел с фиксированной запятой на пополнение не анализируются. Выработка остальных признаков результата также не производится. Признак, полученный в ходе выполнения предыдущей команды, остается неизменным.

Если хотя бы один из сомножителей равен нулю, то процесс умножения не выполняется, а сразу формируется нулевой результат, причем всегда с положительным знаком.

Так как отрицательные числа с фиксированной запятой хранятся в памяти в дополнительном коде, то в моделях ЕС-1030 и ЕС-1050 реализуется способ умножения без преобразования отрицательных сомножителей в прямой код. В модели ЕС-1020 множимое всегда участвует в умножении как положительное число в прямом коде. При этом во всех моделях применяются различные искусственные приемы, способствующие ускорению процессов умножения. В моделях ЕС-1030 и ЕС-1050 знак произведения вырабатывается автоматически в ходе перемножения сомножителей, и произведение получается в коде, соответствующем его знаку. В модели ЕС-1020 знак произведения определяется по алгебраическому правилу путем анализа знаков сомножителей. Код произведения, соответствующий знаку, получается специальным действием после перемножения сомножителей.

Общим для всех моделей является то, что умножение производится, начиная с младших разрядов множителя и со сдвигом

сумм частных произведений вправо. Основные же различия обуславливаются количествами одновременно анализируемых разрядов множителя, способами накопления сумм частных произведений и в связи с этим особенностями проведения их коррекции при отрицательных сомножителях.

Рассмотрим сначала умножение с поразрядным анализом множителя, в случае отрицательного знака, представляемого в дополнительном коде. При этом возможны четыре случая.

1. Если перемножаются положительные целые числа A и B , представленные n -разрядными прямыми кодами $[A]_{пр}$ и $[B]_{пр}$, то положительное произведение C получается в прямом коде просто как

$$[C]_{пр} = [A]_{пр} \times [B]_{пр}. \quad (3.1)$$

Прямой код множимого $[A]_{пр}$ умножается на каждую цифру прямого кода множителя $[B]_{пр}$. Так как в каждом разряде множителя может быть либо 1, либо 0, то процесс умножения представляет собой последовательность операций суммирования и сдвига. Если очередной разряд множителя равен 1, то частное произведение, равное множимому, прибавляется к накопленной промежуточной сумме частных произведений. Полученная новая сумма сдвигается на разряд вправо и осуществляется переход к анализу следующего разряда множителя. Если очередной разряд множителя равен 0, то и очередное частное произведение равно 0. При этом этап суммирования пропускается, а происходит только сдвиг на разряд вправо суммы частных произведений, накопленной к данному шагу умножения.

В старшем разряде кода множителя (в нулевой разрядной позиции его формата) находится знак, в данном случае равный нулю. Умножение на этот нуль приведет лишь к сдвигу предпоследней суммы частных произведений, что обеспечит обязательное появление нуля, соответствующего знаку плюс, в левом крайнем разряде окончательного $2n$ -разрядного произведения.

Пример 3.3. Множимое $A = +7$ множитель $B = +5$ заданы 4-разрядными прямыми двоичными кодами соответственно $[A]_{пр} = 0.111$ и $[B]_{пр} = 0.101$. Требуется найти их произведение.

	Множимое: 0.111	Множитель: 0.101	

Исходная сумма частных произведений:	0000 0000	↓	
	+		
1-е частное произведение	. 0111		1
1-я сумма частных произведений	0111 0000		
Сдвинутая 1-я сумма	0011 1000		
Сдвинутая 2-я сумма	+ 0001 1100		0
3-е частное произведение	+ 0111		1
3-я сумма частных произведений	1000 1100		
Сдвинутая 3-я сумма	0100 0110		
Сдвинутая 4-я сумма	0010 0011		0

Сдвинутая 4-я сумма частных произведений и есть окончательное произведение $[AB]_{пр} = 0.010\ 0011_2 = +35_{10}$.

2. Если множимое $A > 0$, а множитель $B < 0$, то в арифметическое устройство множимое поступает в прямом коде $[A]_{\text{пр}}$, а отрицательный множитель — в дополнительном коде $[B]_{\text{доп}} = 2^n - |B|$, где n — разрядность формата множителя. Для большей наглядности в последнем выражении для дополнительного кода использовано абсолютное значение B .

В результате перемножения положительного и отрицательного чисел должно получиться отрицательное произведение в дополнительном коде $[C]_{\text{доп}} = 2^{2n} - [A]_{\text{пр}} |B|$, однако непосредственное умножение $[A]_{\text{пр}}$ на $[B]_{\text{доп}}$ без специальной коррекции приводит к псевдопроизведению

$$[A]_{\text{пр}}(2^n - |B|) \neq 2^{2n} - [A]_{\text{пр}}|B|. \quad (3.2)$$

Для того чтобы получить точное произведение, необходимо в левую часть (3.2) ввести поправочный член $(2^n - |A|) \cdot 2^n$. Тогда

$$[A]_{\text{пр}}(2^n - |B|) + (2^n - |A|)2^n = 2^{2n} - [A]_{\text{пр}}|B|.$$

Таким образом, для получения точного произведения C в дополнительном коде при $A > 0$ и $B < 0$ необходимо алгоритм умножения на цифры дополнительного кода множителя строить в соответствии с формулой

$$[C]_{\text{доп}} = [A]_{\text{пр}}[B]_{\text{доп}} + [-A]_{\text{доп}}2^n. \quad (3.3)$$

Пример 3.4. Множимое $A = +7$ и множитель $B = -5$ заданы 4-разрядными двоичными кодами: $[A]_{\text{пр}} = 0.111$ и $[B]_{\text{доп}} = 1.011$. Требуется найти их произведение.

	Множимое: 0.111	Множитель: 1.011
Исходная сумма частных произведений:	0000 0000	0000 0000
1-е частное произведение	+ 0111	1
1-я сумма частных произведений	0111 0000	
Сдвинутая 1-я сумма	+ 0011 1000	
2-е частное произведение	0111	1
2-я сумма частных произведений	1010 1000	
Сдвинутая 2-я сумма	0101 0100	
Сдвинутая 3-я сумма	+ 0010 1010	0
4-е частное произведение	0111	1
4-я сумма частных произведений	1001 1010	
Сдвинутая 4-я сумма	+ 0100 1101	
Коррекция	+ 1001	
Результат	1101 1101	

При коррекции прибавляем дополнительный код $[-A]_{\text{доп}}2^4 = (2^4 - A) \cdot 2^4 = 1001\ 0000_2$. Полученный результат есть дополнительный код $[AB]_{\text{доп}} = 1.101\ 1101_2$, равный десятичному числу -35 .

3. При непосредственном умножении отрицательного множимого $A < 0$ на положительный множитель $B > 0$, представленных соответственно дополнительным и прямым кодами $[A]_{\text{доп}}$ и $[B]_{\text{пр}}$, если не корректировать результат, также получается псевдопроизведение

$$(2^n - |A|) [B]_{\text{пр}} \neq 2^{2n} - |A| [B]_{\text{пр}}. \quad (3.4)$$

В данном случае для получения точного произведения в левую часть (3.4) необходимо ввести поправочный член $(2^n - |B|) 2^n$; тогда формула для построения алгоритма умножения будет иметь вид

$$[C]_{\text{доп}} = [A]_{\text{доп}} [B]_{\text{пр}} + [-B]_{\text{доп}} 2^n. \quad (3.5)$$

По сравнению с поправкой в (3.3) одновременный учет поправки в (3.5) оказывается выполнить гораздо сложнее. Для этого необходимо до конца умножения сохранять целиком множитель и иметь возможность в нужный момент, изменив его знак и образовав дополнительный код $[-B]_{\text{доп}}$, прибавить последний к старшим разрядам псевдопроизведения. Однако процессы умножения обычно реализуют таким образом, что разряды множителей, на которые произведено умножение, теряются. Поэтому, чтобы не усложнять схемы, поправку, обусловленную выражением (3.5), вводят в процессе умножения последовательно. Это делается путем искусственного введения на этапах умножения необходимого количества единиц в знаковые разряды сдвигаемых сумм частных произведений.

В процессе умножения отрицательного множимого очередная сумма частных произведений получается в дополнительном коде с единицей в знаковом разряде, если при суммировании не возникает переполнение. При сдвиге этой суммы вправо в знаковом разряде появляется нуль. Чтобы сохранить знак минус, в знаковый разряд при сдвиге вводится специальным образом единица, т. е. выполняется так называемый *модифицированный сдвиг*. Необходимое количество вводимых таким образом в процессе умножения единиц оказывается соответствующим количеству единиц в дополнительном коде множителя $[-B]_{\text{доп}}$. При подсуммировании очередного частного произведения может возникнуть переполнение, сопровождающееся появлением переноса из знакового разряда и в знаковом разряде цифры нуль. В процессе сдвига полученной промежуточной суммы эта единица должна быть помещена в знаковый разряд, хотя она и не относится к числу корректирующих.

Для исключения потерь единиц, возникающих при переполнениях, либо искусственно вводят дополнительный знаковый разряд, тем самым реализуя накопление отрицательных сумм частных произведений в модифицированном дополнительном коде, либо осуществляют предварительный сдвиг множимого вправо, либо применяют специальные схемы анализа и коррекции.

Пример 3.5. Множимое $A = -7$ и множитель $B = +5$ заданы 4-разрядными двоичными кодами $[A]_{\text{доп}} = 1.001$ и $[B]_{\text{пр}} = 0.101$. Требуется найти их произведение.

	Множимое:	1.001	Множитель: 0.101
Исходная сумма частных произведений		+ 0000 0000	↓
1-е частное произведение		<u>1001</u>	1
1-я сумма частных произведений		1001 0000	
Сдвинутая 1-я сумма (Модифицированный сдвиг) *		1100 1000	
Сдвинутая 2-я сумма (Модифицированный сдвиг) *		1110 0100	0
3-е частное произведение		+ <u>1001</u>	1
3-я сумма частных произведений (Переполнение!)	1 ←	0111 0100	
Сдвинутая 3-я сумма	→	1011 1010	
Сдвинутая 4-я сумма (Модифицированный сдвиг) *		1101 1101	0

Последняя сдвинутая сумма частных произведений и есть окончательное произведение $[AB]_{\text{доп}} = 1.101\ 1101$, равное десятичному числу -35 . В примере звездочками отмечены случаи коррекции путем модифицированных сдвигов, соответствующие дополнительному коду поправочного члена в (3.5): $[-B]_{\text{доп}} \times 2^4 = (2^4 - B) \cdot 2^4 = 1011\ 0000_2$.

4. Когда $A < 0$ и $B < 0$, непосредственное перемножение дополнительных кодов даст псевдопроизведение

$$(2^n - |A|)(2^n - |B|) = 2^{2n} + |A||B| - |A|2^n - |B|2^n. \quad (3.6)$$

В данном случае для получения действительного произведения $[C]_{\text{пр}} = [AB]_{\text{пр}}$ необходимо введение в (3.6) двух поправочных членов:

$$(2^n - [A]_{\text{доп}}) 2^n = |A| 2^n \text{ и}$$

$$(2^n - [B]_{\text{доп}}) 2^n = |B| 2^n.$$

Подставляя их в правую часть (3.6), получим

$$[C]_{\text{пр}} = 2^{2n} + |A||B|. \quad (3.7)$$

Так как при суммировании частных произведений по модулю 2^n слагаемое 2^{2n} выйдет за пределы разрядной сетки, то фактически будем иметь истинное произведение $[C]_{\text{пр}} = [|A||B|]_{\text{пр}}$. Поэтому при отрицательных сомножителях алгоритм умножения может строиться в соответствии с формулой

$$[C]_{\text{пр}} = [A]_{\text{доп}} [B]_{\text{доп}} + |A| 2^n + |B| 2^n \quad (3.8)$$

По аналогии с рассмотренным выше первый корректирующий член может быть учтен в конце, а второй — последовательно в ходе умножения.

Пример 3.6. Множимое $A = -7$ и множитель $B = -5$ заданы 4-разрядными двоичными дополнительными кодами $[A]_{\text{доп}} = 1.001$ и $[B]_{\text{доп}} = 1.011$. Требуется найти их произведение.

	Множимое: 1.001	Множитель: 1.011
Исходная сумма частных произведений	+ 0000 0000	↓
1-е частное произведение	<u>1001</u>	1
1-я сумма частных произведений	1001 0000	
Сдвинутая 1-я сумма (Модифицированный сдвиг) *	1100 1000	
2-е частное произведение	+ <u>1001</u>	1
2-я сумма частное произведение (Переполнение!)	1 ← 0101 1000	
Сдвинутая 2-я сумма	→ 1010 1100	
Сдвинутая 3-я сумма (Модифицированный сдвиг) *	+ 1101 0110	0
4-е частное произведение	+ <u>1001</u>	1
4-я сумма частное произведение (Переполнение!)	1 ← 0110 0110	
Сдвинутая 4-я сумма	→ 1011 0011	
Коррекция	+ 0111	
Результат	<u>0010 0011</u>	

Полученный результат равен десятичному числу $+35$. Корректирующий член $|B|_{2^4} = 0101\ 0000$ учтен в ходе умножения при модифицированных сдвигах, отмеченных звездочками, а член $|A|_{2^4} = 0111\ 0000$ — на заключительном этапе.

В машинах Единой системы используются схемы ускоренного умножения. Например, в модели ЕС-1030 применен способ одновременного умножения на два разряда множителя, а в ЕС-1050 — на четыре. В ЕС-1020 также одновременно анализируются четыре разряда множителя.

В АЛУ процессора модели ЕС-1030 в каждом цикле умножения анализируется пара разрядов множителя, начиная с младшей. В результате анализа определяется частное произведение и способ его подсуммирования к уже накопленной сумме, после чего последняя сдвигается на два разряда вправо.

Разряды очередной пары могут содержать 00, 01, 10 или 11. При расшифровывании комбинации 00 производится только сдвиг вправо на два разряда накопленной ранее суммы частных произведений, поскольку частные произведения при умножении на каждый из нулей равны нулю.

Паре разрядов 01 соответствует одно подсуммирование множителя, как при умножении на очередную 1 множителя в рассмотренных выше примерах, но с последующим двойным сдвигом вправо. Умножение на 10 требует удвоения множителя перед суммированием. Чтобы не делать цепей передачи множителя в сумматор со сдвигом на один разряд влево, что равносильно его удвоению, а также из-за особенностей принятого способа коррекции, рассматриваемого ниже, порядок действий при расшифровывании комбинации 10 изменяется. Вначале на один разряд вправо сдвигается накопленная ранее сумма частных произведений, затем в сумматор

подается код очередного частного произведения, после чего результат сложения сдвигается вправо еще на один разряд.

Если же очередной расшифровываемой парой оказывается комбинация 11, то она рассматривается как разность 100—01. Умножение в данном цикле производится на отрицательную величину —01, что выполняется путем вычитания множимого из накопленной суммы частных произведений с последующим сдвигом уменьшенной таким образом суммы на два разряда право. Величина же 100 (четыре) учитывается как дополнительная единица к младшему разряду следующей анализируемой пары. С учетом ее следующая пара разрядов 00, 01, 10 или 11 будет рассматриваться соответственно как 01, 10, 11 или 100. Для обеспечения этого, при обнаружении комбинации 11, в единичное состояние устанавливается специальный триггер, определяющий вариант способа расшифровки следующей пары. Единичное состояние триггера указывает на необходимость учета недостатка, возникшего при умножении на предыдущую пару разрядов. Если при этом следующие два разряда преобразуются к виду 100, то происходит умножение на нулевую пару, а единица продолжает храниться в триггере варианта (ТВАР) и учитывается при анализе последующей пары разрядов.

Рассмотренным способом производится умножение на все пары разрядов 32-разрядного операнда-множителя в том числе и на последнюю пару, включающую знаковый разряд. Одновременное умножение на два старших разряда приводит только к изменениям в выполнении коррекции на величину $[-A] 2^n$ согласно выражению (3.3) или на $|A| 2^n$ в (3.8), когда в знаковом разряде оказывается единица. Если в этих разрядах оказывается комбинация 11 и триггер ТВАР находится в нулевом состоянии, или 10 либо 11 и триггер ТВАР — в единичном состоянии, то приходится умножать на —01 или 00 ($11 + 1 = 100$). В обоих случаях возникающая недостача равносильна введению одного из указанных выше корректирующих членов. Когда же старшая пара разрядов представляет собой комбинацию 10 и содержимое триггера ТВАР равно нулю, то соответствующую поправку надо вводить. Однако для уменьшения количества суммирований в процессе умножения эта коррекция производится отлично от рассмотренного в примерах 3.4 и 3.6.

В примере 3.4 сумму 4-го частного произведения, сдвинутого на разряд вправо, и корректирующего члена можно преобразовать к виду

$$A2^{n-1} + (2^n - |A|) 2^n = (2^n - |A| 2^{-1}) 2^n, \quad (3.9)$$

а в примере 3.6

$$(2^n - |A|) 2^{n-1} + |A| 2^n = (2^n + |A| 2^{-1}) 2^n. \quad (3.10)$$

Согласно выражениям (3.9) и (3.10) умножение множимого на единицу в знаковом разряде отрицательного множителя и последующее вычитание или подсуммирование множимого на этапе коррекции псевдопроизведения можно заменить соответственно вычи-

танием или подсуммированием сдвинутого на один разряд множимого к предварительно сдвинутой на один разряд вправо накопленной сумме частных произведений. Поэтому при умножении на комбинацию 10 в старших разрядах множителя, когда отсутствует единица переноса из предыдущей младшей пары, производится алгебраическое вычитание множимого из суммы частных произведений, сдвинутой на один разряд. Затем выполняется второй сдвиг на разряд вправо. Вычитание между двумя сдвигами суммы частных произведений исключает необходимость в специальном сдвиге множимого вправо на завершающем этапе умножения при необходимости коррекции.

Учет корректирующих членов $[-B]_{\text{доп}} 2^n$ и $|B| 2^n$ согласно выражениям (3.5) и (3.8) производится в ходе умножения путем модифицированных сдвигов сумм частных произведений. Для упрощения организации таких сдвигов, а также для исключения потерь старших значащих разрядов при переполнениях в промежуточных суммах в алгоритм умножения модели ЕС-1030 включен еще один прием.

Перед началом умножения анализируются два старших разряда множимого с целью определения значимости 1-го разряда. Если обнаруживается комбинация 0.1 или 1.0, т. е. правее знакового разряда находится значащая цифра, то множимое сдвигается на один разряд вправо. Для отрицательного операнда производится модифицированный сдвиг. Таким образом, получается модифицированный код множимого, имеющий два знаковых разряда, что, кроме исключения потери значащих разрядов, позволяет сохранять знак промежуточной суммы. Последнее упрощает определение вида сдвига.

Сдвинутое на разряд вправо множимое умножается на множитель. Содержимое же 31-го разряда, выйдя за пределы разрядной сетки сумматора, не может участвовать в процессе накопления сумм частных произведений. Поэтому для определения необходимости коррекции результата значение 31-го разряда запоминается до конца умножения в специальном триггере.

Если в младшем разряде сдвинутого множимого была единица, то в ходе умножения в младших n разрядах ($n = 32$) произведения окажется недостаток, равный полному коду множителя, включая его знаковый разряд, а при отрицательном множителе в старших n разрядах еще и избыток на величину 2^n . Действительно, при A , например, положительном, а B отрицательном согласно (3.3) будем иметь

$$\begin{aligned} & [(A - 1) 2^{-1}]_{\text{пр}} [2^n - |B|] 2 + [2^n - (A - 1)] 2^n = \\ & = 2^{2n} - [A]_{\text{пр}} |B| - (2^n - |B|) + 2^n. \end{aligned} \quad (3.11)$$

Поэтому в случаях, когда производится предварительный сдвиг множимого, в конце умножения анализируется содержимое триггера, хранящего значение вытолкнутого младшего разряда. При единичном состоянии триггера после умножения на последнюю

пару разрядов к младшим 32-м разрядам псевдопроизведения прибавляется код множителя. Перед этой коррекцией предварительно компенсируется начальный сдвиг множимого путем сдвига на один разряд влево накопленной суммы частных произведений.

При положительном множителе в результате его прибавления в конце умножения к младшим разрядам накопленной суммы частных произведений получается окончательный результат. В случаях отрицательного множителя после его подсуммирования необходимо еще устранить избыток 2^n . Вычитание величины 2^n осуществляется путем прибавления дополнения

$$2^{2n} - 2^n = \underbrace{11 \dots 1}_{n \text{ разрядов}} \underbrace{00 \dots 0}_{n \text{ разрядов}}$$

Признаками необходимости этой коррекции служат: наличие предварительного сдвига кода множимого и единицы в его младшем разряде, а также отрицательный знак множителя.

Пример 3.7. Рассмотрим особенности действий при различных комбинациях цифр в анализируемых парах разрядов и возможные случаи коррекции.

а) $A < 0$; $B > 0$. $[A]_{\text{доп}} = 1.001 (-7)$ и $[B]_{\text{пр}} = 0.111 (+7)$.

В множимом за знаком следует значащий разряд, поэтому оно сдвигается на один разряд вправо. Единица из младшего разряда запоминается как указатель необходимости коррекции (в примере выделена рамкой).

Сдвинутое множимое: 1100 1	Множитель: 0111
Исходная сумма частных произведений	+ 0000 0000
Вычитание	+ 0100
Сдвиг на 2 разряда вправо	0001 0000
Сдвиг на 1 разряд вправо	0000 1000
Прибавление	+ 1100
	1100 1000
Сдвиг на 1 разряд вправо	1110 0100
Сдвиг на 1 разряд влево	1100 1000
Коррекция	+ 0111
Результат в дополнительном коде	1100 1111 (-49)

↓ ↓

11 = 100 - 1

↓ ↓

01 + 01 = 10

б) $A > 0$; $B < 0$. $[A]_{\text{пр}} = 0.111 (+7)$ и $[B]_{\text{доп}} = 1.001 (-7)$. Первый разряд множимого — значащий.

Сдвинутое множимое: 0011 1	Множитель: 1001
Исходная сумма частных произведений	+ 0000 0000
Прибавление	+ 0011
	0011 0000
Сдвиг на 2 разряда вправо	0000 1100
Сдвиг на 1 разряд вправо	+ 0000 0110
Вычитание	+ 1101
	1101 0110
Сдвиг на 1 разряд вправо	1110 1011
Сдвиг на 1 разряд влево	1101 0110
Первая коррекция	+ 1001
	1101 1111
Вторая коррекция	+ 1111
Результат	1100 1111 (-49)

↓ ↓

10

↓

01

↓

10

Умножение на старшую пару разрядов 10

в) $A < 0$; $B < 0$. $[A]_{\text{доп}} = 1.001 (-7)$ и $[B]_{\text{доп}} = 1.011 (-5)$. Первый разряд множимого — значащий.

Сдвинутое множимое: 1100 1	Множитель: 1011
Исходная сумма частных произведений	+ 0000 0000
Вычитание	+ 0100
Сдвиг на 2 разряда вправо	+ 0100 0000
Вычитание	+ 0001 0000
Сдвиг на 2 разряда вправо	+ 0100
Сдвиг на 1 разряд влево	0101 0000
Первая коррекция	0001 0100
Вторая коррекция	+ 0010 1000
Результат в прямом коде	+ 1111 0111
	0010 0011 (+35)

$11 = 100 - 1$
 $10 + 01 = 11 = 100 - 1$
 Вычитание 2^4

В модели ЕС-1050 производится умножение одновременно на 4 разряда множителя, начиная с младших разрядов. На каждом шаге умножения анализируется очередная тетрада множителя. Для каждой пары разрядов тетрады определяется одно частное произведение. Полученные таким образом два частных произведе-

Таблица 3.1

Частные произведения при умножении одновременно на 4 разряда

Комбинация цифр в $i + 1$ -й тетраде множи-теля	Пара частных произведений, прибавляемая к i -й сумме частных произведений	Комбинация цифр в $i + 1$ -й тетраде множи-теля	Пара частных произведений, прибавляемая к i -й сумме частных произведений
0000	$0A + 0A$	1000	$2^3A + 0A$
0001	$0A + 1A$	1001	$2^3A + 1A$
0010	$0A + 2^1A$	1010	$2^3A + 2^1A$
0011	$2^2A - 1A$	1011	$-2^2A - 1A$
0100	$2^2A + 0A$	1100	$-2^2A + 0A$
0101	$2^2A + 1A$	1101	$-2^2A + 1A$
0110	$2^2A + 2^1A$	1110	$-2^2A + 2^1A$
0111	$2^3A - 1A$	1111	$0A - 1A$

ния одновременно суммируются с предварительно сдвинутой на четыре разряда вправо текущей суммой частных произведений.

Как и при одновременном умножении просто на два разряда множителя, каждой младшей паре разрядов тетрады ставятся в соответствие частные произведения $0A$, $1A$, 2^1A или $-1A$, получаемые как кратные множимого A соответственно при расщирфровке комбинаций 00, 01, 10 или 11. Последняя комбинация рассматривается как $11 = 100 - 1$ ($2^2 - 1$). При этом, если необходимо, то

учитывается единица переноса, обусловленная, например, комбинацией 11 в старшей паре разрядов предыдущей тетрады.

Для старших двух разрядов каждой тетрады частное произведение определяется по тем же правилам, что и для младших, но имеет дополнительный сдвиг на два разряда влево (умножается на 2^2). Анализ старшей пары разрядов производится с учетом комбинации цифр в младшей паре.

Разновидности пар частных произведений, подсуммируемых к i -й сдвинутой сумме частных произведений, соответствующие различным комбинациям цифр в тетраде, даны в табл. 3.1.

Для исключения возможности потери старших значащих цифр в процессе формирования и суммирования частных произведений, в начале умножения множимое, располагающееся в старших 32 разрядах 64-разрядного регистра, сдвигается вправо на четыре разряда. Этот предварительный сдвиг компенсируется отсутствием сдвига после умножения на последнюю, старшую тетраду.

Пример 3.8. Множимое $[A]_{\text{пр}} = 0111\ 0000 (+112)$ и множитель $[B]_{\text{пр}} = 0111\ 1001 (+121)$. Умножение производится с одновременной расшифровкой 4 разрядов множителя.

Множимое, сдвинутое на 4 разряда

вправо:	0000	0111	Множитель:	0111				1001	
Исходная сумма частных произведений	0000	0000	0000	0000					
Первая пара частных произведений	+	{	0000	0111					
			0011	1000					10
Сдвинутая 1-я сумма частных произведений	+	{	0011	1111	0000	0000			
Вторая пара частных произведений			0000	0011	1111	0000			
			1111	1001					
			0011	1000					
Результат в прямом коде			0011	0100	1111	0000			(+13552)

После предварительного сдвига 32-разрядного множимого все его разряды продолжают участвовать в накоплении суммы частных произведений с помощью 64-разрядного сумматора. Поэтому в данном случае нет необходимости в коррекции произведения из-за предварительного сдвига множимого, присущей модели ЕС-1030. Однако коррекции, обусловленные умножением операндов, представленных в дополнительных кодах, приходится выполнять. При отрицательном множимом здесь также имеют место модифицированные сдвиги. Коррекция же на величину множимого согласно (3.3) и (3.8) при отрицательном множителе вводится только при наличии в старшей тетраде последней комбинации 1000, 1010 или 1001.

В этих случаях на последнем шаге умножения к накопленной и сдвинутой на четыре разряда вправо сумме прибавляется только частное произведение, соответствующее младшей паре разрядов старшей тетрады. В результате, например, при $A > 0$ формируется псевдопроизведение $A2^{32} - A|B| - A2^{31}$. Слагаемое $-A2^{31}$ ука-

зывает на то, что не произведено умножение на знаковый разряд множителя. Это умножение не выполняется, а к псевдопроизведению добавляется величина $2^{64} - A2^{31}$. В итоге получается точный результат

$$A2^{32} - A|B| - A2^{31} + 2^{64} - A2^{31} = 2^{64} - A|B|.$$

Корректирующее слагаемое вводится путем сдвига на 2 разряда влево кода множимого, сдвинутого в начале умножения. В результате предварительного сдвига в умножении участвует множимое $(A2^{32}) 2^{-4} = A2^{28}$. Сдвигая его при коррекции на два разряда влево, получим $(A2^{28}) 2^2 = A2^{30}$. Последующее двойное вычитание этой величины равносильно вычитанию $A2^{31}$.

Пример 3.9. Множимое $[A]_{пр} = 0111\ 0000 (+112)$ и множитель $[B]_{доп} = 1000\ 0111 (-121)$. Перемножим эти величины, пользуясь рассмотренным выше способом.

<p>Множимое, сдвинутое на 4 разряда вправо:</p> <p>Исходная сумма частных произведений</p> <p>Первая пара частных произведений</p> <p>Сдвинутая 1-я сумма частных произведений</p> <p>Первое корректирующее вычитаемое</p> <p>Второе корректирующее вычитаемое</p> <p>Результат в дополнительном коде</p>	<p>+</p> <p>+</p> <p>+</p> <p>+</p> <p>+</p>	<p>0000 0111. Множитель: 1000 0111</p> <pre> 0000 0000 0000 0000 1111 1001 0011 1000 ----- 0011 0001 0000 0000 0000 0011 0001 0000 0000 0000 1110 0100 ----- 1110 0111 0001 0000 1110 0100 ----- 1100 1011 0001 0000 (-13552) </pre> <p style="text-align: right; margin-right: 20px;"> $11 = 100 - 1$ $01 + 01 = 10$ 00 </p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

При комбинации в старшей тетраде множителя 1100, 1101, 1110, 1111 или 1011 никаких специальных корректирующих действий не производится. Результат корректируется автоматически путем неучета единицы переноса из старших разрядов тетрады, представляемых в виде $11 = 100 - 1$.

Пример 3.10. Множимое $[A]_{доп} = 1001\ 0000 (-112)$ и множитель $[B]_{доп} = 1011\ 1001 (-71)$. Ускоренное умножение этих величин будет происходить следующим образом.

<p>Множимое, сдвинутое на 4 разряда вправо:</p> <p>Сдвинутая 1-я сумма частных произведений</p> <p>Результат в дополнительном коде</p>	<p>+</p> <p>+</p> <p>+</p>	<p>1111 1001. Множитель: 1011 1001</p> <pre> 0000 0000 0000 0000 1111 1001 1100 1000 ----- 1100 0001 0000 0000 1111 1100 0001 0000 0000 0111 0001 1100 ----- 0001 1111 0001 0000 (+7952) </pre> <p style="text-align: right; margin-right: 20px;"> $11 = 100 - 1$ $10 + 01 = 11 = 100 - 1$ </p>
----------------------------------------------------------------------------------------------------------------------------------------	----------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

В модели ЕС-1020 каждой тетраде множителя ставится в соответствие подсуммирование к текущей сумме частных произведений не более двух частных произведений, как показано в табл. 3.2.

Т а б л и ц а 3.2

Комбинации подсуммируемых частных произведений
в зависимости от значения очередной анализируемой тетрады множителя

$i+1$ -я тетрада множителя	Подсуммируемые частные произведения (кратные множимому A), в зависимости от значения i -й тетрады (T_i) множителя		$i+1$ -я тетрада множителя	Подсуммируемые частные произведения (кратные множимому A), в зависимости от значения i -й тетрады (T_i) множителя	
	$T_i < 1000$	$T_i > 1000$		$T_i < 1000$	$T_i > 1000$
0000	0	1A	1000	$-(6A + 2A)$	$-(1A + 6A)$
0001	1A	2A	1001	$-(1A + 6A)$	$-6A$
0010	2A	3A	1010	$-6A$	$-(3A + 2A)$
0011	3A	$2A + 2A$	1011	$-(3A + 2A)$	$-(2A + 2A)$
0100	$2A + 2A$	$3A + 2A$	1100	$-(2A + 2A)$	$-3A$
0101	$3A + 2A$	6A	1101	$-3A$	$-2A$
0110	6A	$1A + 6A$	1110	$-2A$	$-1A$
0111	$1A + 6A$	$6A + 2A$	1111	$-1A$	0

В начале операции умножения для множимого A вычисляются его кратные: $2A$, $3A$ и $6A$. Они, как и само A , запоминаются в специально отведенных для них рабочих ячейках местной (локальной) оперативной памяти МОП. Если множимое отрицательно, то предварительно для его кода берется дополнение и фактически по положительному числу определяются указанные кратные величины. Множитель участвует в операции в коде, соответствующем его знаку. Знак произведений определяется согласно алгебраическому правилу о знаках сомножителей и произведения.

Тетрады множителя анализируются последовательно, начиная с младшей. Для определения характера действий на очередном этапе умножения учитывается результат анализа тетрады на предыдущем этапе. Для самой младшей тетрады предыдущая считается равной нулю.

По результату анализа определяется целесообразный вариант суммирования кратных множимого. Перечисленные в табл. 3.2 варианты суммирования составлены из условия обеспечения минимума сложений. Для тетрад от 1000 до 1111 вместо сложения производится вычитание. Накопление сумм частных произведений осуществляется таким образом, что при умножении на очередную тетраду предыдущая сумма участвует в сложении как сдвинутая на 4 разряда вправо.

Знаковый разряд множителя участвует в умножении наравне со значащими. Поэтому знак суммы частных произведений после умножения на самую старшую тетраду соответствует знаку множи-

теля. Отрицательная сумма получается в дополнительном коде. Для определения кода действительного произведения анализируется знак исходного множимого. Если он отрицательный, то от полученной суммы берется дополнение.

Пример 3.11. Перемножим по рассмотренному алгоритму следующие числа: множимое $[A]_{\text{доп}} = 1111\ 1001$ (-7) и множитель $[B]_{\text{доп}} = 1111\ 0101$ (-11).

Дополнение кода множимого: 0000 0111 — оно же есть $1A$

Кратные множимого $\left\{ \begin{array}{l} 0000\ 1110 - 2A \\ 0001\ 0101 - 3A \\ 0010\ 1010 - 6A \end{array} \right.$

		Множитель: 1111	0101
			↓
			3A
Умножение на младшую тетраду	{	+ 0000 0000	
		+ 0001 0101	
		+ 0001 0101	
		+ 0000 1110	
Умножение на старшую тетраду	{	+ 1111 1001	-1A
		0010 0011	2A
		1111 1011 0011	
Сдвинутая вправо сумма частных произведений		1111 1111 1011 0011	
Двоичное дополнение суммы (результат в прямом коде)		0000 0000 0100 1101 (+77)	

При умножении на старшую тетраду частное произведение $-1A$ участвует в сложении со сдвигом на 4 разряда влево («с перекосом»). Необходимый сдвиг байт произведения вправо осуществляется при записи их в МОП. Так как множимое отрицательно, то для конечной суммы частных произведений определяется дополнение.

Если множимое — минимально возможное отрицательное число, дополнение от которого есть такое же число, то умножение производится по другому алгоритму. Этот более простой алгоритм дает возможность искусственным приемом скомпоновать из множителя результат, не формируя кратные множимого.

Деление. По команде РАЗДЕЛИТЬ делимое (первый операнд) делится на делитель (второй операнд), и частное с остатком помещается на место делимого. В наборе команд для действий с фиксированной запятой имеются две команды, одна из которых — формата RR, другая — RX. Для выполнения операции деления, как по команде первого, так и второго формата, делимое — 64-разрядное число со знаком — берется из двух смежных регистров общего назначения. Старшие разряды делимого всегда находятся в регистре с четным адресом, младшие — в следующем в порядке возрастания адресов нечетном регистре. Остаток деления помещается в РОН на место старших разрядов делимого, частное — на место младших. Делитель, остаток и частное со знаками имеют 32-разрядные форматы. В моделях ЕС-1030 и ЕС-1050 отрицательные исходные операнды участвуют в делении в дополнительном коде. Поэтому знак частного в них определяется автоматически в про-

цессе деления кода делимого на код делителя. Отрицательные остаток и частное также получаются в дополнительном коде. В модели ЕС-1020 делятся абсолютные значения делимого и делителя. Знак частного получается по алгебраическому правилу о знаках делимого, делителя и частного. Во всех моделях машин остаток имеет тот же знак, что и делимое. Нулевые частное и остаток всегда положительны.

Перед началом деления делимое и делитель проверяются на равенство нулю. Если обнаруживается нулевой операнд, то деление не производится. Если равно нулю делимое, то на его месте в регистрах РОН остается нулевой результат. В случае нулевого делителя вырабатывается требование прерывания исполняемой программы.

Налагается также ограничение на относительные величины делимого и делителя. Так как лишь для предельных отрицательных значений делителя и частного делимое может быть равно 2^{62} (01000...0), то во всех остальных случаях оно не должно превосходить $|(2^{31} - 1) \cdot 2^{31}| = 2^{62} - 2^{31}$. Если соотношение между делимым и делителем таково, что частное как целое число со знаком не сможет уместиться в 32-разрядном формате, то деление не производится и формируется требование прерывания. Такой случай классифицируется как некорректность деления. Проверка на корректность осуществляется перед началом операции.

Для того чтобы частное по абсолютной величине не превысило число, которое вместе со знаком можно разместить в пределах 32 разрядов, должно выполняться условие

$$\frac{|D|}{|d|} < 2^{31} \quad (3.12)$$

в случае одинаковых знаков делимого D и делителя d или

$$\frac{|D|}{|d|} < 2^{31} + 1, \quad (3.13)$$

если знаки разные. В последнем неравенстве единицей в правой части учитывается то, что при использовании дополнительного кода в одной и той же разрядной сетке может быть представлено отрицательных чисел на единицу больше, нежели положительных. В начале деления с фиксированной запятой в зависимости от знаков исходных операндов производится проверка корректности деления либо по условию (3.12), либо (3.13).

При одинаковых знаках определяется и анализируется разность $D - 2^{31}d$, а при разных знаках — $D - (2^{31}d + d)$ *. Если уменьшаемое по абсолютной величине оказывается меньше вычитаемого, то деление возможно. После этого начинается многошаговый процесс определения цифр частного.

* В модели ЕС-1050 удобнее оказалось реализовать анализ по разностям $2^{31}d - D$ и $2^{31}d + d - D$. В модели ЕС-1020, так как делятся абсолютные значения делимого и делителя, проверка корректности всегда осуществляется с помощью разности $D - 2^{31}d$.

Процесс собственно деления складывается из последовательности циклически повторяющихся операций сдвига делимого и последующих остатков и нахождения разности между ними и делителем. В моделях ЭВМ Единой системы принят метод деления без восстановления остатка. Поэтому независимо от получившегося на предыдущем шаге деления знака остатка, последний перед очередным нахождением разности для определения следующей цифры частного сдвигается на 1 разряд влево. Из положительного остатка вычитается абсолютное значение делителя, к отрицательному — прибавляется. Последнее правомерно, так как сдвиг двоичного остатка R_k на разряд влево есть не что иное, как умножение его на два, и прибавление к сдвинутому отрицательному остатку абсолютного значения $|d|$ равносильно вычитанию его из предварительно восстановленного и сдвинутого остатка, т. е.

$$2R_k + |d| = 2(R_k + |d|) - |d|. \quad (3.14)$$

В модели ЕС-1020 делятся абсолютные значения делимого и делителя. Для вычитания из абсолютного значения делимого и положительных текущих остатков используется дополнение кода делителя. При таком способе деления цифры частного определяются по следующему правилу: *если очередная разность положительна, то в соответствующий разряд частного заносится 1, если отрицательна, то — 0*. При первом вычитании делителя из старших разрядов делимого устанавливается возможность дальнейшего деления. Если первая цифра частного получается равной единице, то это указывает на некорректность деления.

Знак последнего остатка должен соответствовать знаку делимого. При делении абсолютных величин он должен быть положительным. Если последний остаток получается отрицательным, то он восстанавливается добавлением к нему делителя. После этого остатку присваивается знак действительного делимого.

В процессе деления получающиеся цифры частного заносятся в освобождающиеся разряды регистров, где располагались младшие разряды делимого.

Пример 3.12. Пусть $D > 0$ и $d > 0$, причем делимое $[D]_{\text{пр}} = 0.001\ 0001'$ (+17) и делитель $[d]_{\text{пр}} = 0,100'$ (+4). Требуется выполнить деление рассмотренным выше методом.

Вначале проверяем корректность деления. Для этого следует найти разность $[D]_{\text{пр}} - [d2^3]_{\text{пр}} = [D]_{\text{пр}} + [-d \cdot 2^3]_{\text{доп}}$, т. е.

$$\begin{array}{r} + 0.001\ 0001 \\ + 1.110\ 0 \\ \hline 1.111\ 0001 \end{array}$$

Деление возможно, так как результат получился отрицательным.

В действительности проверка на корректность совмещается с первым циклом деления. При этом предварительно анализируется на значимость 1-й разряд делимого. В данном случае правее знакового разряда находится нуль —значащая цифра, поэтому сдвигаем делимое на разряд влево и начинаем деление*.

* Если в первом разряде значащая цифра, то следует удостовериться, не является ли делитель минимально возможным отрицательным числом и можно ли на него разделить делимое.

Сдвинутое на разряд влево делимое: Вычитание делителя	$ \begin{array}{r} + 0010\ 0010 \text{ Делитель: } 0.100. \\ \underline{1100} \\ 1110\ 001\overline{0} \\ + 1100\ 01\overline{0} \\ \underline{0100} \\ 0000\ 01\overline{0}\overline{1} \\ + 0000\ 1\overline{0}\overline{1}\overline{0} \\ \underline{1100} \\ 1100\ 1\overline{0}\overline{1}\overline{0} \\ + 1001\ \overline{0}\overline{1}\overline{0}\overline{0} \\ \underline{0100} \\ \left\{ \begin{array}{l} + 1101\ \overline{0}\overline{1}\overline{0}\overline{0} \\ \underline{0100} \end{array} \right. \text{ — остаток должен быть} \\ \text{положительный!} \\ 0001\ \overline{0}\overline{1}\overline{0}\overline{0} \\ \hline \underbrace{\hspace{1.5cm}}_{\text{Остаток}} \quad \underbrace{\hspace{1.5cm}}_{\text{Частное}} \end{array} $
Отрицательный остаток. Деление возможно!	
Сдвиг 1-го остатка Добавление делителя	
Сдвиг 2-го остатка Вычитание делителя	
Сдвиг 3-го остатка Добавление делителя	
Коррекция 4-го остатка путем восстановления	
Результат	

Цифры, обведенные рамкой, иллюстрируют процесс формирования значенных разрядов частного и последовательного ввода его на место младших разрядов делимого. Частное p и остаток r получились в прямом коде, т. е. $[p]_{\text{пр}} = 0.100 (+4)$ и $[r]_{\text{пр}} = 0.001 (+1)$.

При положительных операциях в коррекции частного нет необходимости.

Так как в моделях ЕС-1030 и ЕС-1050 операнды участвуют в делении вместе со знаками, то все отрицательные числа оказываются представленными в АЛУ в дополнительном коде. В связи с этим, поскольку исходный делитель выражен не в виде абсолютного значения, при вычислении очередной разности между остатком и делителем, в каждом конкретном случае производится следующее действие:

Знак остатка	Знак делителя	Производимое действие
+	{ + —	Вычитание Прибавление
—	{ + —	Прибавление Вычитание

Это означает, что если знак полученного остатка не совпадает со знаком делителя, то после сдвига остатка на разряд влево делитель подсуммируется к нему в том коде, в каком он поступил на деление в АЛУ. В случае совпадения знаков, при посылке делителя в сумматор от его исходного кода берется дополнение, которое и суммируется с кодом остатка.

При делении с учетом алгебраических знаков операндов разряды частного определяются по следующему более общему пра-

вилу: если знак очередного остатка совпадает со знаком делителя, то соответствующий шаг деления разряд частного равен 1, в противном случае — 0. Как и в рассмотренном выше примере, очередная цифра частного обычно заносится в младший разряд регистра, где вначале находилось делимое, а затем остатки. Этот разряд освобождается при сдвиге влево остатка и цифр частного, определенных на предыдущих шагах деления. Если, как в примере 3.12, проверка на корректность деления совмещается с определением первой цифры частного, то на первом шаге делимое сдвигается на один разряд влево и производится алгебраическое вычитание из него делителя. Путем сравнения знаков 1-го остатка и делителя вырабатывается значение нулевого разряда частного. Затем, как и при делении абсолютных значений, 1-й остаток сдвигается влево на один разряд и далее повторяются действия согласно рассмотренному выше алгоритму. Для определения 32 разрядов частного они повторяются 32 раза.

Как и при умножении с использованием дополнительных кодов отрицательных операндов, для получения истинных значений частного и остатка в конце процесса деления может возникать необходимость в проведении коррекции полученных чисел. Однако, в отличие от умножения, в данной операции не представляется возможной коррекция путем закономерного учета каких-то определенных поправочных членов. Она может быть выполнена только на основе сравнительного анализа исходных операндов и результата.

Для получения остатка, имеющего тот же знак, что и делимое, в конце деления производится сравнение их знаков. Если знаки не совпадают, то делитель прибавляется к окончательному остатку либо вычитается из него таким образом, чтобы знак и величина последнего изменились в нужном направлении. Нулевой остаток всегда имеет положительный знак.

Необходимость коррекции частного обуславливается неточностью его определения в пределах заданного формата в случае не делящихся нацело исходных чисел, а также из-за погрешности в определении последней цифры вследствие применения дополнительного кода для представления отрицательных операндов. Округление для учета отбрасываемых разрядов не производится. Погрешность же второго вида корректируется.

При делении по рассмотренному выше способу с использованием дополнительного кода частное может получаться с недостатком на единицу младшего разряда. Случаи появления недостатка могут обнаруживаться путем анализа знаков делимого, делителя и остатка. При некоторых комбинациях знаков требуется также определять равенство остатка нулю. В результате, если необходимо, то к младшему разряду частного прибавляется единица.

Деление положительных чисел согласно последнему алгоритму не отличается от приведенного в примере 3.12. Поэтому рассмотрим ряд примеров деления целых отрицательных чисел,

Пример 3.13. Пусть $D > 0$ и $d < 0$. Возьмем, как и в предыдущем примере, $[D]_{\text{пр}} = 0.001\ 0001 (+17)$, а делитель — отрицательным, т. е. $[d]_{\text{доп}} = 1.100 (-4)$.

Сдвинутое на разряд влево делимое:

$$\begin{array}{r} +0010\ 0010 \text{ Делитель: } 1.100 \\ \hline +1100 \end{array}$$

Сдвиг 1-го остатка

$$\begin{array}{r} 1110\ 001\overline{11} \\ \hline +1100\ 01\overline{11}0 \\ \hline 0100 \end{array}$$

Сдвиг 2-го остатка

$$\begin{array}{r} 0000\ 01\overline{11}0 \\ \hline +0000\ 1\overline{11}0\overline{0} \\ \hline 1100 \end{array}$$

Сдвиг 3-го остатка

$$\begin{array}{r} 1100\ 1\overline{11}0\overline{11} \\ \hline +1001\ \overline{11}0\overline{11}0 \\ \hline 0100 \end{array}$$

Коррекция остатка и частного

$$\left\{ \begin{array}{r} +1101\ \overline{11}0\overline{11}1 \\ \hline +0100\ \quad +1 \\ \hline 0001\ \overline{11}10\overline{01} \end{array} \right.$$

Результат

$$\begin{array}{r} 0001\ \overline{11}10\overline{01} \\ \hline \hline \end{array}$$

Остаток Частное

В данном случае частное $[p]_{\text{доп}} = 1.100 (-4)$ и остаток $[r]_{\text{пр}} = 0.001 (+1)$. При положительном делимом и отрицательном делителе частное всегда требует коррекции. Остаток корректируется только в тех случаях, когда его знак не совпадает со знаком делимого.

Пример 3.14. $D < 0$; $d > 0$. Этот случай рассмотрим на примере деления $[D]_{\text{доп}} = 1.110\ 1111 (-17)$ на $[d]_{\text{пр}} = 0.100 (+4)$.

Сдвинутое на разряд влево делимое:

$$\begin{array}{r} +1101\ 1110 \text{ Делитель: } 0.100 \\ \hline +0100 \end{array}$$

Сдвиг 1-го остатка

$$\begin{array}{r} 0001\ 111\overline{11} \\ \hline +0011\ 11\overline{11}0 \\ \hline 1100 \end{array}$$

Сдвиг 2-го остатка

$$\begin{array}{r} 1111\ 11\overline{11}0 \\ \hline +1111\ 1\overline{11}0\overline{0} \\ \hline 0100 \end{array}$$

Сдвиг 3-го остатка

$$\begin{array}{r} 0011\ \overline{11}0\overline{11} \\ \hline +0111\ \overline{11}0\overline{11}0 \\ \hline 1100 \end{array}$$

Коррекция остатка и частного

$$\left\{ \begin{array}{r} +0011\ \overline{11}0\overline{11}1 \\ \hline +1100\ \quad +1 \\ \hline 1111\ \overline{11}10\overline{01} \end{array} \right.$$

Результат

$$\begin{array}{r} 1111\ \overline{11}10\overline{01} \\ \hline \hline \end{array}$$

Остаток Частное

Частное и остаток получались в дополнительном коде: $[p]_{\text{доп}} = 1.100 (-4)$ и $[r]_{\text{доп}} = 1.111 (-1)$. При делении отрицательного делимого на положительный делитель для определения необходимости коррекции частного приходится анализировать не только знаки операндов и остатка, но и величину последнего. Например, коррекция частного не требуется, если в остатке получился нуль.

Пример 3.15. $D < 0$; $d < 0$. Разделим $[D]_{\text{доп}} = 1.110\ 1111 (-17)$ на $[d]_{\text{доп}} = 1.100 (-4)$.

Сдвинутое на разряд влево делимое:	$\begin{array}{r} + 1101\ 1110 \text{ Делитель: } 1.100 \\ \hline 0100 \\ \hline 0001\ 111\ \overline{0} \\ \hline 0011\ 11\ \overline{0} \\ \hline 1100 \\ \hline 1111\ 11\ \overline{0}\ \overline{1} \\ \hline 1111\ \overline{0}\ \overline{1}\ \overline{0} \\ \hline 0100 \\ \hline 0011\ \overline{0}\ \overline{1}\ \overline{0} \\ \hline 0111\ \overline{0}\ \overline{1}\ \overline{0}\ \overline{0} \\ \hline 1100 \\ \hline \left\{ \begin{array}{l} + 0011\ \overline{0}\ \overline{1}\ \overline{0}\ \overline{0} \\ + 1100 \end{array} \right. \\ \hline 1111\ \overline{0}\ \overline{1}\ \overline{0}\ \overline{0} \end{array}$
Сдвиг 1-го остатка	
Сдвиг 2-го остатка	
Сдвиг 3-го остатка	
Коррекция окончательного остатка	
Результат	$\underbrace{1111}_{\text{Остаток}} \underbrace{\overline{0}\ \overline{1}\ \overline{0}\ \overline{0}}_{\text{Частное}} \quad -1; +4$

При отрицательном делимом и отрицательном делителе частное требует коррекции только при делении нацело, т. е. когда получается нулевой остаток.

Рассмотренный выше способ деления с фиксированной запятой применен в модели ЕС-1030.

В модели ЕС-1050 также принят метод деления без восстановления остатка. Однако реализуется он иначе.

Особенность деления в модели ЕС-1050 состоит в том, что для сокращения времени его выполнения во всех определяемых остатках, кроме конечного, не производится полное приведение переносов. Каждый очередной остаток получается в двухрядном коде, состоящем из кодов поразрядной суммы и поразрядных переносов. Для определения знака остатка полное приведение переносов осуществляется только в старших его разрядах. В зависимости от знаков приведенной части остатка и делителя при определении очередной цифры частного делитель прибавляется к остатку или вычитается из него (см. стр. 97). При этом на входы сумматора подаются три слагаемых: сдвинутые коды поразрядной суммы и поразрядных переносов и соответствующий действию код делителя.

В ходе деления формируется не истинный код частного, а два условных кода, которые обозначим A и B , где A — положительная

составляющая частного, а B — отрицательная. Для этих кодов справедливо равенство

$$D = (A - B)d + r, \quad (3.15)$$

где, как и ранее, D — делимое, d — делитель и r — остаток. Частное p определяется в конце деления как

$$p = A - B. \quad (3.16)$$

Однорядный код остатка r получается лишь в конце деления путем полного приведения поразрядных переносов. Если необходимо, то он предварительно корректируется (восстанавливается), что сопровождается коррекцией частного.

Для понимания причины применения такого способа деления необходимо кратко рассмотреть особенность функционирования схемы суммирования АЛУ процессора ЕС-2050. Эта схема состоит из двух ступеней. В первой комбинационной ступени происходит образование поразрядных сумм и поразрядных переносов трех 64-разрядных слагаемых. Для временного запоминания кодов сумм $S[0-64]$ и переносов $Q[0-64]$ на выходе этой ступени имеются два регистра. Окончательное приведение переносов и, таким образом, получение однорядного кода результата происходит во второй ступени — комбинационном сумматоре с каскадно-параллельными цепями распространения переносов.

В первой ступени при сложении трех 64-разрядных кодов $X = x_0x_1 \dots x_i \dots x_{63}$, $Y = y_0y_1 \dots y_i \dots y_{63}$ и $Z = z_0z_1 \dots z_i \dots z_{63}$ в каждом i -ом разряде вначале вырабатываются промежуточные сумма s'_i и перенос q'_i , а затем — окончательные значения суммы s_i и переноса q_i , т. е.

$$\begin{array}{rcl}
 \dots & x_i & x_{i+1} \dots & \text{— первое слагаемое} \\
 \dots & y_i & y_{i+1} \dots & \text{— второе слагаемое} \\
 \dots & z_i & z_{i+1} \dots & \text{— третье слагаемое} \\
 \hline
 \dots & s'_i & s'_{i+1} \dots & - S' \\
 \dots & q'_i & q'_{i+1} & q'_{i+2} \dots & - Q' \\
 \hline
 \dots & s_i & s_{i+1} \dots & - S \\
 & q_i & q_{i+1} \dots & - Q
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \\ \text{— промежуточный двухрядный код} \\ \\ \text{— окончательный двухрядный код} \end{array}$$

Промежуточные и окончательные значения получаются в соответствии с формулами:

$$s'_i = x_i \oplus y_i \oplus z_i; \quad (3.17)$$

$$q'_i = x_i y_i \vee x_i z_i \vee y_i z_i; \quad (3.18)$$

$$s_i = s'_i \oplus q'_{i+1}; \quad (3.19)$$

$$q_i = s'_i q'_{i+1}. \quad (3.20)$$

Предположим, что к кодам 0010 1101 и 0100 1000 прибавляется дополнение кода 0100 1111, т. е. 1011 0001.

Суммирование в первой ступени будет происходить следующим образом:

$$\begin{array}{r}
 + \left\{ \begin{array}{l} 0010 \ 110 \ 1 \\ 0100 \ 100 \ 0 \\ 1011 \ 000 \ 0 \end{array} \right. \\
 \hline
 S' \quad 1101 \ 010 \ 1 \\
 Q' \quad 0101 \ 000 \ \overline{11} \\
 \hline
 S \quad 1000 \ 010 \ 0 \\
 Q \quad 0101 \ 000 \ 1 \quad \left. \vphantom{\begin{array}{l} S \\ Q \end{array}} \right\} \text{— окончательный двухрядный код}
 \end{array}$$

Вначале на входы третьего слагаемого подается обратный код 1011 0000, представляющий собой поразрядную инверсию кода 0100 1111. Единица, которую необходимо подсуммировать к младшему разряду обратного кода, чтобы получить дополнение, заносится в младший разряд промежуточного кода Q' поразрядных переносов. Эта единица выделена рамкой. Цена единицы переноса q_i соответствует соседнему старшему двоичному разряду $i - 1$. Это означает, что при подаче, например, кодов окончательных разрядных сумм и переносов снова на входы схемы суммирования первой ступени код переносов надо сдвинуть на разряд влево. В процессе деления двухрядные коды остатков сдвигаются влево. При этом, если код поразрядных сумм сдвигается на один разряд, то код поразрядных переносов — в ту же сторону на два разряда.

Способ деления с полным приведением переносов лишь в нескольких старших разрядах остатков применим только при наложении определенных ограничений на величину делителя. Для операции деления с фиксированной запятой в модели ЕС-1050 принято, что делитель d должен удовлетворять условию

$$2^{30} \leq |d| \leq 2^{31}, \quad (3.21)$$

т. е. должен быть нормализован.

Тогда, если соотношение между величинами делимого и делителя удовлетворяет также условиям корректности (3.12) и (3.13), то любой очередной остаток по абсолютной величине будет меньше делителя. Следовательно, ни в одном из остатков не может быть обнаружено переполнение. Это, в свою очередь, дает возможность, введя дополнительный знаковый разряд, определять знак остатка путем полного приведения переносов, например, лишь в четырех старших разрядах его модифицированного кода. При исключении

случаев переполнения в четырех приведенных разрядах, включающих два знаковых, возможно появление следующих девяти комбинаций:

Значение четырех старших приведенных разрядов остатка	Знак остатка		
00.00 } 00.01 } 00.10 } 00.11 }	Положительный		
10.11 } 11.00 } 11.01 } 11.10 }		Отрицательный	
11.11			Не может быть определен без знания переноса из остальной неприведенной части остатка

Первые четыре комбинации соответствуют положительному остатку, вторые — отрицательному. Комбинации с 01. . . и 10. . . в знаковых разрядах, кроме приведенной выше, встретиться не могут, так как соответствуют числам, по абсолютной величине большим единицы. Комбинация 10.11 является допустимой, так как возможен перенос из неприведенной части, при учете которого получится код 11.00. . .

Девятая комбинация в зависимости от наличия или отсутствия переноса из неприведенной части остатка может превратиться соответственно либо в 00.00. . ., либо остаться такой, как есть: 11 11. . . Поэтому в данном случае без знания переноса из неприведенной части невозможно определить знак остатка. Однако, если учесть, что как при комбинации 00.00, так и при 11.11 по абсолютной величине остаток не превосходит 2^{61} , то можно провести следующие рассуждения.

Пусть полученный остаток r_i удовлетворяет условию

$$-2^{61} \leq r_i < 2^{61}, \quad (3.22)$$

тогда удвоение его даст

$$-2^{62} \leq 2r_i < 2^{62}. \quad (3.23)$$

В зависимости от знаков остатка и делителя делитель либо будет прибавляться к старшей половине удвоенного остатка, либо вычитаться из нее, т. е. следующим действием будет

$$r_{i+1} = 2r_i \pm 2^{32}d. \quad (3.24)$$

Поскольку величина $2r_i$ удовлетворяет неравенству (3.23), то при нормализованном делителе (3.21) знак r_{i+1} окажется противоположным знаку r_i . Следовательно, для определения r_{i+2} придется выполнять действие, обратное предыдущему, т. е.

$$r_{i+2} = 2(2r_i \pm 2^{32}d) \mp 2^{32}d, \quad (3.25)$$

что равносильно

$$r_{i+2} = 2 \cdot 2r_i \pm 2^{32}d \quad (3.26)$$

или

$$r_{i+2} = 2r_{i+1} \pm 2^{32}d. \quad (3.27)$$

Таким образом, при появлении комбинации 11.11... можно не определять очередную разность, ограничившись лишь сдвигом остатка на разряд влево. После сдвига величину $2r_i$ можно рассматривать как новый несдвинутый двухрядный остаток r_{i+1} .

Для того чтобы определить характер следующего действия, надо привести и проанализировать его старшие четыре * разряда.

При каждом вычислении разности между полученным остатком и делителем формируются цифры положительной A и отрицательной B составляющих частного. Правило формирования этих цифр сводится к следующему.

Если при определении очередной разности на вход сумматора подается код делителя, обратный исходному, то в разряд положительной составляющей частного заносится единица, а соответствующий разряд отрицательной составляющей остается в нулевом состоянии. Если же делитель подается на вход сумматора, не изменяясь, в том коде, в каком он поступил на деление, то единица, наоборот, заносится в разряд отрицательной составляющей частного. В случае обнаружения комбинации, не определяющей знак остатка, соответствующие шагу деления разряды положительной и отрицательной составляющих частного имеют нулевое значение.

Коды A и B по сути дела отражают последовательность действий при определении частного. Если единицы кода A имеют положительный знак, а единицы кода B — отрицательный, то, выполняя над делителем процесс, обратный делению, получим

$$D = Ad + B(-d) + r, \quad (3.28)$$

что есть то же самое, что и (3.15).

Согласно (3.16) код частного p определяется путем вычитания из кода положительной составляющей A кода отрицательной составляющей B . Для этого прямой код $[A]_{\text{пр}}$ суммируется в сумматоре АЛУ с дополнительным кодом $[B]_{\text{доп}}$.

Процесс деления начинается с проверки операндов на равенство их нулю и на корректность соотношения между ними (см. стр. 95). Если никаких особенностей не обнаружено, то в случае необходимости производится лишь нормализация делителя. Значащая часть кода делителя сдвигается влево до тех пор, пока он не станет удовлетворять условию (3.21). Удовлетворение этому условию обнаруживается по наличию в нулевом и первом разрядах разных цифр, т. е. 0.1... или 1.0....

* В ЕС-1050 в специальном сумматоре АУ одновременно приводятся переносы в пяти старших разрядах. Однако при рассмотрении деления в данной главе это принципиального значения не имеет.

Количество разрядов, на которое сдвинулся делитель при нормализации, подсчитывается специальным счетчиком. Затем на столько же разрядов влево сдвигается делимое. На первом шаге деления делимое сдвигается еще на разряд влево и, в зависимости от знаков операндов, делитель вычитается из него или прибавляется к нему.

Циклически повторяющийся процесс деления продолжается до тех пор, пока либо не будут определены значения всех 32 разрядов частного, либо не будет обнаружен остаток, равный нулю. Последнее имеет место в случае деления нацело.

По окончании процесса деления вычисляется частное. Если последний остаток имеет тот же знак, что и делимое, или равен нулю, то частное определяется просто как разность (3.16).

Пример 3.16. Пусть требуется разделить число $[D]_{пр} = 0.001\ 0000 (+16)$ на число $(d)_{пр} = 0.100 (+4)$.

Делитель d представляет собой нормализованное число, поэтому его предварительного сдвига не требуется. Опуская этап проверки на корректность, процесс деления рассмотрим, начиная с вычитания делителя. Оно будет осуществляться путем прибавления к модифицированному прямому коду сдвинутого на разряд влево делимого (00.010 0000) модифицированного обратного кода делителя 11.011 1111.

	Полное приведение переносов в стар- ших четырех раз- рядах	Услов- ный код частного
	+ 00.010 0000 + 11.011 1111	+ 00.01 + 11.01
	----- S' 11 001 1111 Q' 00 100 0001 -----	11.10 -----
1-й остаток	{ S 11 101 1110 Q 00 000 0001 -----	
Сдвинутый влево 1-й оста- ток	{ 11 011 1100 + 00 000 0100 00 100 0000 -----	11.01 + 00.00 00.10 -----
	S' 11 111 1000 Q' 00 000 1000 -----	11.11 -----
2-й остаток	{ S 11 111 0000 Q 00 000 1000 -----	
3-й остаток	{ 11 110 0000 00 001 0000 -----	A : 100 B : 010

Так как в результате приведения старших четырех разрядов 1-го остатка совместно с прибавляемым делителем получилась комбинация 11.11, то для получения 3-го остатка ограничиваемся лишь сдвигом на разряд влево 2-го остатка.

Третий остаток получился равным нулю. Специальной схемой это обнаружится перед приведением его старших разрядов и процесс деления закончится. В незаполненных разрядах условного кода частного останутся нули, т. е. в данном случае будем иметь

A : 1000
B : 0100

На завершающем этапе, вычисляя частное в соответствии с (3.16), получим:

$$\begin{array}{r} + 1000 \\ \hline 1100 \\ \hline 0100 \text{ (частное } +4), \end{array}$$

и на месте делимого в АЛУ будет сформирован код 0.0000 0100. В старших четырех разрядах этого кода находится нулевой остаток.

Так как в любом i -м разряде двухрядного кода остатка одновременно s_i и q_i не могут быть равны единице, то в общем случае двухрядный код нуля имеет вид

$$\begin{array}{l} S \ 11 \dots 100 \dots 00 \\ Q \ 00 \dots 010 \dots 00 \end{array}$$

В АЛУ ЕС-1050 имеется специальная схема, которая обнаруживает нулевой остаток при появлении любого из следующих двухрядных кодов:

$$\begin{array}{l} \left\{ \begin{array}{l} S \ 111 \ 1100 \dots 0 \\ Q \ 000 \ 0010 \dots 0, \end{array} \right. \quad \left\{ \begin{array}{l} S \ 110 \ 0000 \dots 0 \\ Q \ 001 \ 0000 \dots 0, \end{array} \right. \\ \left\{ \begin{array}{l} S \ 111 \ 1000 \dots 0 \\ Q \ 000 \ 0100 \dots 0, \end{array} \right. \quad \left\{ \begin{array}{l} S \ 100 \ 0000 \dots 0 \\ Q \ 010 \ 0000 \dots 0, \end{array} \right. \\ \left\{ \begin{array}{l} S \ 111 \ 0000 \dots 0 \\ Q \ 000 \ 1000 \dots 0, \end{array} \right. \quad \left\{ \begin{array}{l} S \ 000 \ 0000 \dots 0 \\ Q \ 000 \ 0000 \dots 0. \end{array} \right. \end{array}$$

При обнаружении одного из перечисленных нулевых кодов, сдвиги остатков, а следовательно, и процесс деления заканчиваются. Если нулевой код не соответствует одному из приведенных выше, но в результате приведения старших разрядов все время получается код 11,11, то сдвиги двухрядного кода будут продолжаться. Сдвиги будут до тех пор, пока не будет получен либо обнаруживаемый нулевой код остатка, либо очередное приведение даст код, отличный от 11,11.

Если последний остаток не равен нулю и знак его противоположен знаку делимого, то по окончании процесса деления производится восстановление остатка. При этом делитель складывается с ним или вычитается из него согласно следующей схеме:

Знак		Действие	
делимого	остатка	делителя	восстановления
+	—	+	Сложение
+	—	—	Вычитание
—	+	+	Вычитание
—	+	—	Сложение

Восстановление последнего остатка влечет за собой необходимость коррекции окончательно вычисляемого частного. Если при восстановлении код делителя прибавляется к остатку, то частное уменьшается на единицу. Это уменьшение осуществляется путем подсуммирования к положительной составляющей A частного всего лишь обратного кода отрицательной составляющей B , а не дополнения, как в примере 3.16. В случае же вычитания кода делителя частное, наоборот, увеличивается на единицу. При под-

суммировании дополнения от B к составляющей A , к младшему разряду суммы добавляется еще одна единица.

Пример 3.17. Предположим, требуется разделить $[D]_{\text{пр}} = 0.001\ 0001 (+17)$ на $[d]_{\text{пр}} = 0.100 (+4)$. Как и в предыдущем примере, делитель представляет собой нормализованное число.

Собственно деление начинается с вычитания делителя из сдвинутого на ряд влево делимого.

		Полное приведение Услов- переносов в стар- ших четырех раз- рядах	
	+ 00.010 0010	+ 0001	$A : 1$
	<u>11.011 1111</u>	<u>1101</u>	$B : 0$
S'	11 001 1101	1110	
Q'	<u>00 100 0101</u>		
S	11 101 1000		
Q	<u>00 000 0101</u>		
Сдвинутый влево 1-й оста- ток	{ 11 011 0000	1101	$A : 10$
	+ 00 001 0100	+ 0000	$B : 01$
	<u>00 100 0000</u>	<u>0010</u>	
S'	11 110 0100	1111	
Q'	<u>00 010 0000</u>		
S	11 100 0100		
Q	<u>00 010 0000</u>		
Сдвинутый влево 2-й оста- ток	{ 11 000 1000	+ 1100	$A : 100$
	<u>00 100 0000</u>	<u>0100</u>	$B : 010$
		0000	
Сдвинутый влево 3-й оста- ток	{ 10 001 0000	+ 1000	$A : 1001$
	+ 10 000 0000	+ 1000	$B : 0100$
	<u>11 011 1111</u>	<u>1101</u>	
S'	11 010 1111	1101	
Q'	<u>00 010 0001</u>		
S	11 000 1110		
Q	<u>00 010 0001</u>		

Так как 4-й остаток отрицателен, то необходимо его восстановить. В данном случае восстановление будет осуществляться путем простого подсуммирования делителя к двухрядному коду 4-го остатка:

11 000 1110	
+ 00 100 0010	
<u>00 100 0000</u>	— делитель
S'	11 000 1100
Q'	<u>01 000 0100</u>
S	10 000 1000 — двухрядный код
Q	01 000 0100 — окончательного остатка

После полного приведения переносов в двухрядном коде остатка получим

+ 10 000 1000
<u>10 000 1000</u>
00 001 0000
остаток, равный +1

С учетом поправки (—1) на восстановление остатка частное определится как

$$\begin{array}{r} + 1001 \\ \underline{1011} \\ 0100 (+4) \end{array}$$

На место делимого будет помещен код 0001 0100, в первых четырех разрядах которого находится восстановленный остаток (+1), а в следующих — частное (+4).

Пример 3.18. Требуется разделить $[D]_{\text{доп}} = 1.110\ 1111$ (—17) на $[d]_{\text{пр}} = 0.100$ (+4).

Процесс собственно деления будет выполняться следующим образом.

	Полное приведение переносов в старших четырёх разрядах	Услов- ный код частного
	+ 11.101 1110	+ 11.10
	<u>00.100 0000</u>	<u>00.10</u>
S'	11 001 1110	00.00
Q'	<u>01 000 0000</u>	
S	10 001 1110	
Q	<u>01 000 0000</u>	
Сдвинутый влево 1-й оста- ток {	00 011 1100	00.01
	00 000 0000	00.00
	+ 11 011 1111	+ 11.01
S'	<u>11 000 0011</u>	11.10
Q'	<u>00 111 1001</u>	
S	11 111 1010	
Q	<u>00 000 0001</u>	
Сдвинутый влево 2-й оста- ток {	11 111 0100	11.11
	+ 00 000 0100	+ 00.00
	00 100 0000	00.10
S'	<u>11 011 0000</u>	00.01
Q'	<u>01 000 1000</u>	
S	10 011 1000	
Q	<u>01 000 0000</u>	
Сдвинутый влево 3-й оста- ток {	00 111 0000	00.11
	+ 00 000 0000	+ 00.00
	11 011 1111	11.01
S'	<u>11 100 1111</u>	00.00
Q'	<u>00 110 0001</u>	
4-й остаток { S	11 010 1110	
Q	<u>00 100 0001</u>	

Последнее приведение переносов в старших разрядах показывает, что 4-й остаток имеет положительный знак. Для его восстановления надо вычесть дели-
тель, т. е.

$$\begin{array}{r} 11\ 010\ 1110 \\ + 01\ 000\ 0010 \\ \underline{11\ 011\ 1111} \\ S'\ 01\ 001\ 0011 \\ Q'\ 10\ 101\ 1101 \\ \underline{S\ 11\ 100\ 1110} \\ Q\ 00\ 001\ 0001 \end{array}$$

После окончательного приведения переносов получим

$$\begin{array}{r} 11\ 100\ 1110 \\ 00\ 010\ 0010 \\ \hline 11\ 111\ 0000 \end{array}$$

остаток, равный -1

С учетом поправки (+1) на восстановление остатка частное будет равно

$$\begin{array}{r} 0101 \\ +0110 \\ \hline 1 \end{array} \quad (-4).$$

В итоге на место делимого будет помещен результат

$$1111\ 1100 \quad (-1; -4).$$

Если перед началом деления имела место нормализация делителя, то результирующий остаток получается увеличенным. Поэтому при наличии предварительной нормализации, после восстановления остатка и определения частного, производится сдвиг окончательного остатка вправо. Остаток сдвигается на количество разрядов, на которое сдвигался делитель при нормализации.

Преобразование десятичных чисел в двоичные. По команде ПРЕОБРАЗОВАТЬ В ДВОИЧНУЮ, имеющей формал RХ, десятичное число, находящееся по адресу второго операнда, переводится в двоичную систему счисления. Полученное двоичное число в коде, соответствующем знаку, помещается по адресу первого операнда R. Команда может оперировать только десятичными числами, представленными в упакованном формате. Переводимые числа всегда должны располагаться в 64-разрядных ячейках, адреса которых начинаются с целочисленных границ и должны делиться на 8. В четырех младших разрядах десятичного числа располагается знак. Коды знака и двоично-десятичных цифр в процессе перевода проверяются на правильность. Цифры не должны превышать девяти, а код знака, наоборот, должен быть больше девяти. При обнаружении ошибки, она фиксируется как некорректность данных, и формируется требование прерывания исполняемой программы.

Переведенное число должно иметь не более 32 разрядов. В таком формате могут быть размещены двоичные числа, соответствующие десятичным, не превышающим 2 147 483 647 и не меньшим $-2\ 147\ 483\ 648$. Если переводимое число выходит за данные пределы, то в регистр с адресом помещаются младшие 32 разряда переведенного числа и возникает требование прерывания. При этом вырабатывается признак: «Некорректность деления с фиксированной запятой». Невозможность размещения в 32 разрядах получающегося в ходе перевода двоичного числа обнаруживается по возникающему переносу из старшего его разряда.

Машинный перевод целых десятичных чисел в двоичную систему счисления выполняется по схеме Горнера на основе соотношения

(2.2). В данном случае a_i — цифры десятичного числа в двоично-десятичном изображении, а $N = 10$. Процесс перевода строится аналогично рассмотренному в примере 2.3, с тем лишь отличием, что в операции участвуют двоично-десятичные цифры и десятичное основание в виде 1010. В моделях ЕС-1050 и ЕС-1020 умножение на 1010 выполняется путем сложения двух слагаемых: 1-е — результат умножения на 2^1 (0010), получающийся сдвигом множимого на разряд влево, 2-е — результат умножения на 2^3 . Последнее образуется путем сдвига множимого влево на три двоичных разряда.

Пример 3.19. Перевести десятичное число 95 (1001 0101) в двоичное.

$$\begin{array}{r}
 \times \quad 1001 \quad 0101 \\
 \quad 1010 \\
 \hline
 \text{1-е слагаемое} \quad 0010010 \\
 \text{2-е слагаемое} \quad 1001000 \\
 \hline
 \text{Следующая цифра} \quad + \quad 1011010 \\
 \quad \quad \quad \quad \quad \quad 0101 \leftarrow \\
 \hline
 \text{Результат} \quad \quad \quad 1011111
 \end{array}$$

Если десятичное число отрицательное, то результат — двоичное число — в конце операции переводится в дополнительный код.

В модели ЕС-1030 умножение каждой двоично-десятичной цифры на 10_{10} (1010_2) выполняется по схеме $((a_i + a_i) 2 + a_i) 2$. В остальном перевод осуществляется так же, как в рассмотренном примере.

Преобразование двоичных чисел в десятичные. Команда ПРЕОБРАЗОВАТЬ В ДЕСЯТИЧНУЮ также имеет формат RX. По этой команде первый операнд — 32-разрядное двоичное число переводится в десятичную систему счисления. Полученное целое десятичное число со знаком, выраженное в упакованном формате в виде слова двойной длины, отсылается по адресу второго операнда. Кодирование знака (по КОИ-8 или ДКОИ) зависит от значения 12-го разряда ССП (см. п. 2.2). Знак помещается в младшие четыре разряда. Под значащую часть числа отводится 60 двоичных разрядов, где могут быть размещены 15 десятичных цифр, выраженных в двоично-десятичной форме. Поэтому при переводе 32-разрядного двоичного числа переполнение невозможно.

Операция перевода целого двоичного числа в десятичное может выполняться путем последовательного деления его на 10, как показано, например, в примере 2.2. Однако из-за длительности и сложности выполнения самой операции деления этот способ оказывается нерациональным.

В модели ЕС-1050 операция перевода в десятичную систему счисления строится на основе многократного повторения операций умножения. Для этого целое двоичное число вначале преобразуется в правильную дробь. Так как $10^9 < 2^{32} < 10^{10}$, то для преобразо-

вания в правильную дробь достаточно исходное двоичное число умножить на двоичную константу

$$K_2 = 0001\ 1011\ 0111\ 1100\ 1101\ 1111\ 1101\ 1010,$$

удовлетворяющую соотношению

$$K_{10} \cdot 2^{-63} = 10^{-10}.$$

Переводимое число и константа перемножаются как целые числа. Затем для получения двоично-десятичных цифр произведение последовательно умножается на 10_{10} (1010_2), как в примере 2.2 при переводе правильной дроби.

В модели ЕС-1030 перевод в десятичную систему счисления осуществляется на основе алгоритма, рассмотренного в примере 2.3. Так как в результате перевода формируется двоично-десятичное число, то в реализации этого алгоритма имеется ряд особенностей.

Умножение на 2 и подсуммирование очередной двоичной цифры производится путем сдвига переводимого числа на один разряд влево. При этом выталкиваемые из одного регистра двоичные цифры последовательно вводятся в другой регистр, где формируются и накапливаются двоично-десятичные тетрады переведенного числа. Содержимое младших четырех разрядов принимающего регистра все время анализируется. Как только содержимое этой тетрады становится больше 9, к нему прибавляется 6. Возникающая при этом единица переноса подсуммируется к цифре соседней старшей тетрады. Если же при очередном сдвиге из младшей тетрады выталкивается единица, то также производится коррекция. К младшей тетраде прибавляется 6, и единица переноса передается в соседнюю старшую тетраду.

Пример 3.20. Перевести двоичное число 0101 1111 в десятичную систему счисления.

Исходное число		0101 1111
Сдвиг 1-й	0000 0000	1011 1110
2-й	0000 0001	0111 1100
3-й	0000 0010	1111 1000
4-й	0000 0101	1111 0000
Сдвиг 5-й	0000 1011	1110 0000
	+	
Коррекция	0000 0110	
	0001 0001	1110 0000
Сдвиг 6-й	0010 0011	1100 0000
7-й	0100 0111	1000 0000
8-й	1000 1111	0000 0000
	+	
Коррекция	0110	
Результат	1001 0111	

В итоге получили десятичное число 95.

Отрицательные двоичные числа перед переводом преобразуются в прямой код.

В модели ЕС-1020 перевод в десятичную систему выполняется также по схеме примера 2.3 в основном аналогично рассмотренному в примере 3.20.

3.3. АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ НАД ДВОИЧНЫМИ ЧИСЛАМИ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

В системе команд ЕС ЭВМ имеется большой набор команд для выполнения операций над числами, представленными в полулогарифмической форме. Особенности такого представления чисел в машинах единой системы рассмотрены в пп. 2.4 и 2.5. Чтобы можно было оперировать как длинными, так и короткими операндами, в наборе команд все команды для операций с плавающей запятой представлены в двух вариантах (см. приложение табл. 10). Кроме того, в каждом из этих вариантов для большинства операций предусмотрены как команды формата RR, так и RX. В данном случае адресами R указываются номера специальных 64-разрядных регистров для операндов с плавающей запятой. Этих регистров четыре (рис. 2.9 и 2.10). Кроме регистров РПЗ, для выполнения операций с плавающей запятой во всех моделях ЭВМ Единой системы используется в основном то же оборудование, что и для операций с фиксированной запятой.

Как и для действий с фиксированной запятой, набор команд для операций с плавающей запятой содержит команды арифметических операций, а также команды загрузки регистров РПЗ, отсылки в память, сравнения и присвоения знака.

При выполнении всех команд, кроме отсылки в память, результаты замещают операнды, адресуемые первым адресом. После операций сложения, вычитания, сравнения и присвоения знака вырабатывается и устанавливается в слове состояния программы (ССП) код условия.

Сложение и вычитание. Для сложения и вычитания как длинных, так и коротких операндов, имеются две разновидности команд: с нормализацией результата и без нормализации. По команде СЛОЖИТЬ И НОРМАЛИЗОВАТЬ второй операнд алгебраически складывается с первым операндом и нормализованная сумма помещается на место первого операнда. Если нормализация не требуется, то на место первого операнда записывается ненормализованный результат. Выполнение операций вычитания как с нормализацией, так и без нормализации отличается от сложения тем, что в начале операции знак второго операнда искусственно меняется на обратный. В результате получается, что второй операнд вычитается из первого, хотя формально во всем остальном эта операция выполняется как алгебраическое сложение.

В основной памяти и в РПЗ характеристики и мантиссы, из которых состоят числа в форме с плавающей запятой, хранятся

в прямом коде. При вызове операндов в арифметическое устройство по команде СЛОЖИТЬ или ВЫЧЕСТЬ характеристики и мантиссы разделяются и поступают в схемы, где над ними производятся необходимые действия. Вначале производится сравнение порядков исходных чисел. В модели ЕС-1030 оно выполняется просто путем логического сопоставления кодов характеристик с целью определения наличия неравенства, а также меньшей из них. В моделях ЕС-1020 и ЕС-1050 из характеристики первого операнда всегда вычитается характеристика второго. Если порядки слагаемых или уменьшаемого и вычитаемого не одинаковы, то производится выравнивание порядков путем сдвига вправо мантиссы числа с меньшим порядком. Так как в ЕС ЭВМ при представлении чисел в полулогарифмической форме основание системы счисления принято равным $2^4 = 16$, то при выравнивании порядков сдвиги производятся на количества разрядов, кратные четырем. Иными словами, производится сдвиг вправо на количество шестнадцатиричных цифр, равное величине разности характеристик. При разности, большей 8, в ЕС-1050 осуществляются ускоренные сдвиги сразу на 8 разрядов. Одновременно со сдвигом мантиссы вправо на 4 или на 8 разрядов ее характеристика увеличивается соответственно на единицу или на два.

Если основание системы счисления $N = 16$, то сумма двух нормализованных чисел: $A = m_A 16^{p_A^*}$ и $B = m_B 16^{p_B^*}$, представленных в полулогарифмической форме, может быть записана в виде

$$S = m_S 16^{p_S^*} = m_A 16^{p_A^*} + m_B 16^{p_B^*}, \quad (3.29)$$

где p_A^* , p_B^* и p_S^* — характеристики слагаемых и суммы.

Мантиссы представляются меньшими единицы. Поэтому, чтобы не получить мантиссы, большей единицы, порядки выравниваются в сторону большего порядка. Пусть $p_A^* > p_B^*$, тогда для выравнивания порядков характеристику p_B^* следует увеличить на величину разности $p_A^* - p_B^*$, т. е.

$$B \frac{16^{p_A^* - p_B^*}}{16^{p_A^* - p_B^*}} = \frac{m_B}{16^{p_A^* - p_B^*}} 16^{p_B^*} \cdot 16^{p_A^* - p_B^*} = m'_B 16^{p_A^*}, \quad (3.30)$$

где m'_B — мантисса m_B , разделенная на $16^{p_A^* - p_B^*}$, т. е. сдвинутая на $p_A^* - p_B^*$ шестнадцатиричных разрядов вправо. Так как $16^{p_A^* - p_B^*} = 2^4 (p_A^* - p_B^*)$, то этот шестнадцатиричный сдвиг соответствует сдвигу на $4(p_A^* - p_B^*)$ двоичных разрядов.

Сумма чисел с выравненными порядками будет равна:

$$S = m_S 16^{p_S^*} = m_A 16^{p_A^*} + m'_B 16^{p_A^*} = (m_A + m'_B) 16^{p_A^*}. \quad (3.31)$$

Следовательно, мантисса суммы $m_S = m_A + m_B$ и характеристика суммы $p_S^* = p_A^*$. Эти значения могут быть не окончательными, если после сложения мантисс будет производиться нормализация результата.

Таким образом, собственно сложение чисел с разными порядками при наличии нормализации выполняется в три этапа:

- 1) выравнивание порядков слагаемых;
- 2) алгебраическое сложение мантисс;
- 3) нормализация результата.

Так как в формате числа в полулогарифмической форме мантисса всегда находится в прямом коде, то в арифметических устройствах мантиссы с одинаковыми знаками всегда складываются как положительные числа. В этом случае результату всегда присваивается знак первого операнда. Если же знаки не равны, то, например, в моделях ЕС-1020 и ЕС-1030 при алгебраическом сложении отрицательная мантисса преобразуется в дополнительный код.

Как было указано в п. 2.5, в процессоре модели ЕС-1050 при разных знаках всегда переводится в дополнительный код мантисса числа с большим порядком, находящаяся на месте второго операнда. Объясняется это тем, что на место мантиссы первого операнда всегда помещается мантисса числа с меньшим порядком. Только на этом месте может быть произведен ее сдвиг при выравнивании порядков. Поэтому, если сравнение порядков показывает, что характеристика первого операнда больше второго, то перед их выравниванием мантиссы меняют местами (производится их взаимоперекрестный обмен).

При сложении чисел с разными знаками, как это делается в ЕС-1030, для определения знака результата достаточно просуммировать цифры, отображающие знаки исходных операндов, и значение переноса из старшего разряда суммы мантисс. В модели ЕС-1050 возможность перекрестной перемены мест мантисс усложняет определение знака результата.

При сложении мантисс в процессоре этой модели используются модифицированный прямой и дополнительный коды. Это означает, что в сумматоре два разряда левее старшего значащего разряда выполняют роль знаковых. Знак, получающийся в результате суммирования кодов мантисс, определяется по значению левого из этих двух разрядов. Для определения же знака результата выполнения операции сложения (вычитания) в целом этот знак анализируется совместно с кодом операции, результатом сравнения (суммы по mod 2) знаков исходных операндов и признаком, указывающим на то, была ли взаимоперекрестная перемена мест мантисс слагаемых (или уменьшаемого и вычитаемого) или нет. Учет кода операции необходим, так как при вычитании в начале операции знак вычитаемого искусственно меняется на обратный.

Если в итоге сложения кодов мантисс результат получается в дополнительном коде, то он в конце выполнения операции переводится в прямой код.

При сложении мантисс, по абсолютной величине меньших единицы, возможны случаи переполнения разрядной сетки мантиссы результата. Часто это называют нарушением нормализации влево. В модели ЕС-1050, например, признаком переполнения служит появление разных цифр в двух знаковых разрядах перед запятой $01' \dots$ или $10' \dots$. Оно устраняется сдвигом мантиссы вправо на четыре разряда и увеличением характеристики результата на единицу. При этом может возникнуть переполнение порядка. Тогда выработается требование прерывания.

Получение результата, по абсолютной величине меньшего $0'.00010 \dots 0$, классифицируется как нарушение нормализации вправо. Если выполняется операция с нормализацией, то результирующая мантисса сдвигается влево до тех пор, пока старшая тетрада мантиссы результата не станет отличной от нуля. При каждом сдвиге мантиссы на четыре разряда влево характеристика результата уменьшается на единицу. В модели ЕС-1050, как и при выравнивании порядков, сдвиги могут происходить сразу на восемь разрядов. При нормализации влево возможно появление характеристики, меньшей нуля (порядка, меньшего — 64). В этом случае вырабатывается требование прерывания по причине исчезновения порядка, а характеристике и мантиссе результата присваиваются нулевые значения.

В результате алгебраического сложения может получиться мантисса, равная нулю. В этом случае реакция на такой результат зависит от того, разрешено или нет прерывание при потере значимости [5]. Знак суммы при нулевой мантиссе результата всегда положителен.

Пример 3.21. Пусть требуется выполнить операцию СЛОЖИТЬ и НОРМАЛИЗОВАТЬ над числами: $[A]_{\text{пр}} = 0.1000101' \quad '00010010$ и

$\underbrace{\hspace{10em}}_{\text{зн. числа}} \quad \downarrow \quad \underbrace{\hspace{10em}}_{\text{мантисса}}$
 $\hspace{10em} \downarrow$
 $\hspace{10em} \text{характеристика}$

$[B]_{\text{пр}} = 1.1000100' \quad '11110000$, что соответствует

$\underbrace{\hspace{10em}}_{\text{зн. числа}} \quad \downarrow \quad \underbrace{\hspace{10em}}_{\text{мантисса}}$
 $\hspace{10em} \downarrow$
 $\hspace{10em} \text{характеристика}$

$$A = m_A 16_{10}^{p_A^*} = +0,00010010 \cdot 10000_{10}^{1000101} \text{ и}$$

$$B = m_B 16_{10}^{p_B^*} = -0,11110000 \cdot 10000_{10}^{1000100}$$

Для упрощения рассматриваем мантиссы 8-разрядными. Характеристики p_A^* и p_B^* представляют собой

$$p_A^* = [p_A]_{\text{пр}} + 2_{10}^6 = 0.000101' + 1000000_2 = 1000101' \text{ и}$$

$$p_B^* = [p_B]_{\text{пр}} + 2_{10}^6 = 0.000100' + 1000000_2 = 1000100',$$

где p_A и p_B — истинные порядки ($p_A = +000101$ и $p_B = +000100$).

1. Выравниваем порядки слагаемых, для чего из характеристики p_A^* вычитаем характеристику p_B^* . Вычитание заменяем прибавлением дополнения p_B^* до 2^2 , т. е. 0111100.

Тогда

$$\begin{array}{r} + 1000101 \\ + 0111100 \\ \hline 0000001^* \end{array}$$

Так как p_A^* на шестнадцатиричную единицу больше p_B^* , то сдвигаем вправо на четыре разряда мантиссу $[m_B]_{\text{пр}} = 1,1111\ 0000$. В результате получаем $[m'_B]_{\text{пр}} = 1.0000\ 1111$.

2. Складываем мантиссы с использованием дополнительного кода:

$$\begin{array}{r} [m_A]_{\text{пр}} = 0.0001\ 0010 \\ + \\ [m'_B]_{\text{доп}} = 1.1111\ 0001 \\ \hline [m_A]_{\text{пр}} + [m'_B]_{\text{доп}} = \overset{\leftarrow}{0}.\overset{\leftarrow}{0}0000\ 0011 = [m_A + m'_B]_{\text{пр}} \end{array}$$

При сложении переносы в знаковый разряд и из знакового согласуются (отмечено стрелками). Поэтому переполнения разрядной сетки мантиссы нет. Однако, поскольку в мантиссе суммы старшая тетрада равна нулю, то имеет место нарушение нормализации вправо.

3. Нормализуем результат, для чего сдвигаем цифровую часть мантиссы на четыре разряда влево и уменьшаем на единицу характеристику суммы. В итоге получим

$$m_S = 0.0011\ 0000.$$

Так как характеристикой суммы является большая характеристика p_A^* , от

$$p_S^* = 100\ 0100.$$

Окончательный результат

$$[S]_{\text{пр}} = 0.100\ 0100''0011\ 0000.$$

В отличие от рассмотренного, в модели ЕС-1050 сложение мантисс будет происходить иначе. Так как характеристика первого операнда больше характеристики второго, то перед сдвигом для выравнивания порядков мантиссы поменяются местами. Знаки мантисс не одинаковые. Поэтому после сдвига мантиссы m_B будет происходить вычитание путем сложения $[m'_B]_{\text{пр}}^M$ с модифицированным дополнительным кодом $[m_A]_{\text{доп}}^M$, т. е.

$$\begin{array}{r} [m'_B]_{\text{пр}}^M = 00.0000\ 1111 \\ + \\ [m_A]_{\text{доп}}^M = 11.1110\ 1110 \\ \hline 11.1111\ 1101 \end{array}$$

Так как при выполнении операции сложения операндов с разными знаками имела место перестановка мантисс, то знак результата, а следовательно, и код его мантиссы должны быть противоположны полученным, т. е. ненормализованная мантисса в прямом коде будет равна

$$0.0000\ 0011.$$

Этот результат соответствует вычисленному выше.

Умножение. По команде УМНОЖИТЬ множимое (первый операнд) умножается на множитель (второй операнд) и нормализо-

ванный результат помещается на место множимого. Имеется четыре варианта команд УМНОЖИТЬ (см. приложение, табл. 10).

Произведение двух чисел $A = m_A 16^{p_A^*}$ и $B = m_B 16^{p_B^*}$ в общем виде равно

$$C = m_C 16^{p_C^*} = m_A m_B 16^{p_A^* + p_B^* - 64}, \quad (3.32)$$

где 64 — величина, на которую все характеристики отличаются от истинных порядков ($64 = 2^6$). Таким образом, мантисса произведения $m_C = m_A m_B$, а характеристика произведения $p_C^* = p_A^* + p_B^* - 64$. В модели ЕС-1030 предварительный порядок произведения вычисляется до перемножения мантисс, в ЕС-1050 — одновременно с их умножением, а в ЕС-1020 — после умножения. Окончательный порядок определяется в конце операции после нормализации результата.

Во всех моделях ЭВМ Единой системы перемножение мантисс чисел, представленных в форме с плавающей запятой, выполняется в основном по тем же алгоритмам, по которым производится умножение чисел с фиксированной запятой. Мантисса первого операнда является множимым, а второго — множителем. Всегда представленные в прямом коде мантиссы перемножаются как положительные числа. Это исключает необходимость в различных коррекциях результатов, имеющих место при использовании дополнительных кодов. Знак произведения определяется путем анализа знаков сомножителей, например, простым суммированием их по mod 2. Полученный знак в конце выполнения операции заносится в знаковый разряд результата.

Умножение производится, начиная с младших разрядов множителя и со сдвигом вправо сумм частных произведений. Число циклов умножения зависит от того, на сколько разрядов множителя одновременно производится умножение, и от разрядности мантисс. В ЕС-1050 может быть не более 14 циклов умножения, что соответствует перемножению 56-разрядных мантисс. В модели ЕС-1030 мантиссы длинных операндов, а в модели ЕС-1020 любых операндов перемножаются по частям. В модели ЕС-1030 младшая половина множителя умножается сначала на старшую половину множимого, а затем на младшую. После этого старшая половина множителя умножается последовательно на старшую и на младшую половины множимого. Все промежуточные произведения, соответствующим образом сдвинутые относительно друг друга, складываются последовательно таким образом, что остаются 56 значащих разрядов.

В отличие от умножения чисел с фиксированной запятой, в модели ЕС-1020 для перемножения мантисс оказалось удобнее использовать следующие кратные мантиссы множимого A : 1А, 2А и 5А. Эти кратные величины формируются в начале умножения и запоминаются в рабочей области местной оперативной памяти.

В зависимости от значения анализируемой шестнадцатиричной цифры множителя согласно табл. 3.3, осуществляется прибавление кратных множимого к текущей сумме частных произведений или вычитание из нее. После умножения на одну шестнадцатиричную цифру множителя накапливаемая сумма частных произведений сдвигается на четыре разряда вправо. Если при умножении на старшую шестнадцатиричную цифру мантиссы множителя производилось вычитание, то после сдвига промежуточного произведения к нему дополнительно прибавляется множимое (1А).

Т а б л и ц а 3.3

Кратные множимого А — частные произведения, соответствующие очередной анализируемой шестнадцатиричной цифре множителя

Цифра множителя	Частные произведения в зависимости от того, как на предыдущую цифру выполнялось умножение		Цифра множителя	Частные произведения в зависимости от того, как на предыдущую цифру выполнялось умножение	
	путем сложения	путем вычитания		путем сложения	путем вычитания
0	0	+А	8	+5А + 2А + А	-5А - 2А
1	+1А	+2А	9	-5А - 2А	-5А - А
2	+2А	+2А + А	А	-5А - А	-5А
3	+2А + А	+2А + 2А	В	-5А	-2А - 2А
4	+2А + 2А	+5А	С	-2А - 2А	-2А - А
5	+5А	+5А + А	Д	-2А - А	-2А
6	+5А + А	+5А + 2А	Е	-2А	-А
7	+5А + 2А	+5А + 2А + А	F	-А	0

Мантиссы результатов как длинных, так и коротких форматов, нормализуются. Мантиссы исходных операндов также нормализуются. В ЕС-1030 нормализуются мантиссы только длинных операндов, а в ЕС-1020 и ЕС-1050 — как длинных, так и коротких.

Перед началом умножения мантиссы множимого и множителя проверяются на равенство нулю. Если обнаруживается мантисса, равная нулю, то умножение не производится, а сразу формируется нулевой результат.

После нормализации мантиссы результата, если в этом была необходимость, и окончательного формирования его характеристики произведение анализируется на переполнение характеристики и на исчезновение порядка. При переполнении разрядной сетки характеристики (если она превысит 127) исполнение программы прерывается. Если характеристика результата окажется меньшей нуля, то ей и мантиссе присваиваются нулевые значения и вырабатывается требование прерывания по причине исчезновения порядка. Прерывание по этой причине происходит, если есть разрешение в соответствующем разряде маски ССП. При умножении признак результата не вырабатывается.

Деление. По любой из модификаций команд РАЗДЕЛИТЬ (см. приложение, табл. 10) делимое (1-й операнд) делится на делитель (2-й операнд) и замещается частным. Остаток от деления чисел, представленных в полулогарифмической форме, не сохраняется.

После вызова операндов в АУ выполнение операции деления с плавающей запятой состоит в вычислении характеристики частного и делении мантисс. Для определения характеристики частного из характеристики делимого вычитается характеристика делителя и к полученной разности прибавляется число 64_{10} ($1\ 000\ 000_2$), т. е. находится $p_D^* - p_d^* + 64$. Этот результат затем может корректироваться после вычисления мантиссы результата.

Процесс деления мантисс в основном строится по тому же алгоритму, по которому в данной модели ЭВМ делятся целые числа в операциях с фиксированной запятой. Отличие обусловлено тем, что мантиссы по абсолютной величине меньше единицы (правильные дроби) и представлены в прямом коде. Поэтому они всегда делятся как положительные числа по методу без восстановления остатка. Знак частного определяется путем анализа знаков делимого и делителя. Результирующий знак заносится в нулевой разряд кода частного в конце операции.

Перед делением мантиссы делимого и делителя нормализуются. Поэтому мантисса частного получается нормализованной. В модели ЕС-1030 нормализация производится таким образом, чтобы в регистрах, где расположены мантиссы делимого и делителя, содержимые разряды с 8-го по 11-й не были равны нулю, т. е. были отличны от нуля старшие тетрады. Для исключения переполнения при делении мантисс делимое должно быть меньше делителя. В ЕС-1030 после нормализации это условие проверяется путем пробного вычитания делителя из делимого. Если разность окажется положительной, то перед началом деления делитель сдвигается влево еще на четыре разряда и к предварительной характеристике частного прибавляется единица. Если при этом не произошло переполнения, то выполняется деление. При возникновении переполнения происходит прерывание исполняемой программы. Деление мантисс при этом не производится.

В арифметическом устройстве процессора модели ЕС-1050 мантиссы делимого и делителя сдвигаются на семь разрядов влево, а затем нормализуются. Сдвиг на семь разрядов позволяет разместить мантиссы так же, как целые числа, и тем самым реализовать метод деления, как и в операции с фиксированной запятой. После этого сдвига делимое нормализуется с точностью до шестнадцатичной цифры, а делитель — до двоичной.

Одинаковый сдвиг мантисс делимого и делителя влево на семь разрядов, как и сдвиг их при нормализации на количество разрядов, разнящееся не более чем на 3, не влечет за собой изменения характеристики частного. Если мантисса делителя сдвигается влево

лишь на один, два или три двоичных разряда больше по сравнению с мантиссой делимого, то для получения правильного результата частное должно увеличиваться соответственно в 2, 4 или 8 раз. Это достигается соответствующим сдвигом мантиссы частного влево на один, два или три двоичных разряда. При сдвиге делителя в процессе нормализации более чем на 4 разряда по сравнению со сдвигом делимого характеристика частного корректируется. Она увеличивается на число единиц, равное количеству четверок разрядов, составляющих разность между сдвигами мантисс делимого и делителя. Кроме того, если в результате нормализации все же оказывается, что мантисса делимого равна или больше мантиссы делителя, то для получения мантиссы частного, меньшей 1, требуется ее дополнительная нормализация на одну шестнадцатиричную позицию вправо (сдвиг вправо на четыре двоичных разряда). В этом случае также производится увеличение на единицу характеристики частного. Как и в ЕС-1030, при переполнении характеристики в ЕС-1050 происходит прерывание исполняемой программы.

В модели ЕС-1020, для упрощения действий по выработке цифр частного, перед делением мантисс образуются кратные величины нормализованной мантиссы делителя: d , $2d$, $4d$, $8d$, где d — мантисса делителя. Использование этих кратных величин исключает необходимость сдвигов остатков после каждого очередного прибавления или вычитания делителя. Деление также выполняется по методу без восстановления остатка. Однако в данном случае цикл деления содержит четыре действия типа сложения или вычитания последовательно кратных $8d$, $4d$, $2d$ и d . После каждого сложения или вычитания определяется двоичная цифра частного. Определение производится по правилу для деления абсолютных значений делимого и делителя. После выработки четырех двоичных цифр частного остаток сдвигается на четыре двоичных разряда (на шестнадцатиричную цифру) влево. После этого цикл деления повторяется.

Количество шагов деления мантисс зависит от длины операндов, над которыми производится деление.

Перед началом деления мантиссы делителя и делимого проверяются на равенство нулю. При равенстве нулю мантиссы делителя операция не выполняется и происходит прерывание исполнения программы по причине некорректности деления с плавающей запятой. Делимое сохраняется неизменным. Если окажется, что мантисса делимого равна нулю, то сразу по адресу первого операнда записывается нулевой результат.

По окончании деления, кроме проверки на переполнение, производится сравнение характеристики частного с нулем. При обнаружении характеристики, равной или меньшей нуля, вырабатывается требование прерывания по причине исчезновения порядка. Прерывание произойдет, если в ССП имеется на это разрешение. При делении нормализованных мантисс может произойти нарушение

ние нормализации влево. Это устраняется сдвигом мантиссы частного на шестнадцатиричную цифру вправо и увеличением на единицу его характеристики.

При выполнении операции деления анализ результата для выработки его признака не производится. Код условия, полученный в ходе исполнения предыдущей команды, не меняется.

3.4 АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ НАД ДЕСЯТИЧНЫМИ ЧИСЛАМИ

Набор команд (см. приложение, табл. 10) для действий над десятичными числами дает возможность выполнять арифметические операции: сложения, вычитания, умножения и деления. Кроме того, в этом наборе имеются команды для преобразования десятичных данных из распакованного формата в упакованный и обратно, для сравнения операндов, их пересылки. Перечисленные операции выполняются над операндами, находящимися в основной памяти. Поэтому команды данной группы относятся к типу SS (память—память).

Форма представления десятичных данных, их возможные форматы и машинные коды были рассмотрены выше в пп. 2.2—2.5. Непосредственно в арифметических операциях десятичные числа участвуют в упакованном формате. Поля, которые они занимают в основной оперативной памяти, могут начинаться с любого байта и иметь длину до 16 байт (31 десятичная цифра, представленная в виде двоично-десятичной тетрады, плюс одна знаковая тетрада). В этих пределах операнды могут иметь разную длину. Допускается определенным образом перекрытие полей операндов арифметических и логических операций [5].

В процессе выполнения операций байты десятичных операндов преобразуются последовательно справа налево. Результат помещается на место первого операнда. Поэтому по разрядности он не должен превосходить длину поля, соответствующую первому адресу. Возможность нарушения этого условия при арифметических операциях и сравнении автоматически контролируется. Кроме того, во время выполнения операций постоянно контролируется правильность кодов двоично-десятичных цифр и знаков.

Признаки результатов устанавливаются только в конце операций типа сложения и сравнения.

Сложение и вычитание. В этих операциях десятичные целые числа — операнды — участвуют выравненными по правой границе, определяемой местоположением фиксированной запятой. Для выравнивания операндов в алгоритмах операций предусматриваются специальные действия. Например, в модели ЕС-1030 оно осуществляется путем сдвига операндов вправо таким образом, чтобы их правые крайние байты примыкали к границе слова. В модели ЕС-1050 в выбранных в АУ двойных словах местоположения правых границ операндов фиксируются специальными счет-

чиками. От положений границ ведется счет количеств байт, составляющих десятичные числа, над которыми производится операция.

В модели ЕС-1020 при каждом обращении к ООП выбирается полуслово, содержащее байт с четным и байт с нечетным адресами. В двух полусловах, в которых располагаются начала операндов, младшие цифры слагаемых могут располагаться различно: либо в четных, либо в нечетных байтах, либо в байтах различной четности. Это обстоятельство учитывается при организации выполнения операции. В микропрограмме операции автоматически выбирается такое направление, чтобы операнды в операции участвовали выравненными по младшим цифрам.

В упакованном формате младшие, правые крайние, байты операндов содержат знаковые тетрады. Эти тетрады (крайние правые в младших байтах) перед началом сложения или вычитания сравниваются. Если знаки одинаковые, то по команде СЛОЖИТЬ десятичные числа складываются в прямом коде. Результату присваивается знак операндов. При несовпадающих знаках в различных моделях ЭВМ действия выполняются по-разному. Например, в модели ЕС-1030 операнды перед сложением сравниваются друг с другом по абсолютной величине, затем из большего вычитается меньший, и результату присваивается знак большего операнда. Вычитаемое представляется в дополнительном коде (дополнение до 10^n ; см. п. 2.5). Результат всегда получается в прямом коде. В модели ЕС-1050 предварительное сравнение операндов не производится, а всегда при разных знаках из первого операнда вычитается второй, для чего первый операнд в прямом коде складывается с дополнительным кодом второго. Код результата устанавливается по наличию или отсутствию переноса из его старшего байта. Если при разных знаках операндов есть перенос, то это означает, что результат получился в прямом коде, в противном случае — в дополнительном. Для определения знака результата надо еще учесть, какие знаки имели исходные операнды. В модели ЕС-1020 всегда вычитается более короткий операнд.

Непосредственное сложение кодов операндов в моделях ЕС-1020 и ЕС-1050 выполняется побайтно, а в ЕС-1030 — пословно. В последнем случае в сложении одновременно участвуют по 4 байта от каждого слагаемого.

При выполнении команды ВЫЧЕСТЬ знак второго операнда предварительно изменяется на обратный, после чего действие над десятичными операндами производится так же, как и при алгебраическом сложении.

Когда складываются операнды с одинаковыми знаками, в коде с избытком 6 в моделях ЕС-1020 и ЕС-1050 представляется первый операнд, а в ЕС-1030 — второй; при разных знаках избыток содер­жится в дополнении.

Пример 3.22. Аналогично примеру 2.12, используя кодирование с избытком 6, выполнить арифметические операции сложения и вычитания.

1. Сложить числа $A = +82$ и $B = -75$ (+1000 0010 и -0111 0101). Дополнение 75 до 10^3 равно 25, тогда

$$\begin{array}{r}
 \begin{array}{l} 82 \text{ или } 1000 \ 0010 \ A \\ +25 \text{ или } +1000 \ 1011 \end{array} \\
 \hline
 \begin{array}{l} 07 \text{ или } 0000 \ 1101 \\ \text{п} \leftarrow 07 \quad \text{п} \leftarrow 0000 \end{array} \\
 \hline
 \begin{array}{l} 1101 \\ +1010 \end{array} \\
 \hline
 0000 \ 0111 \text{ Истинный результат } A + B.
 \end{array}$$

Дополнение B (с избытком 6 в каж-
дой тетраде)
Коррекция результата

В выполненных действиях дополнение второго числа определено путем инвертирования значений двоичных разрядов в тетрадах и прибавлением единицы к младшему разряду правой тетрады. Такое дополнение получается с потетрадным избытком 6. Поэтому, если при сложении (см. п. 2.5) возникает перенос из какой-либо тетрады, то это означает, что в ней получилась истинная цифра.

Если же при сложении нет переноса из тетрады, то она должна быть скорректирована. Коррекция осуществляется вычитанием из этой тетрады избытка 6, что заменяется прибавлением 1010 — дополнения 6 до 16.

Наличие переноса p из старшей тетрады при сложении A с дополнением B указывает на то, что результат получился в прямом коде. Так как первый операнд положительный, то результат имеет положительный знак, т. е. равен +7.

2. Вычтись из $A = 75$ $B = 82$, т. е. найти разность $75 - 82$. Вычитание заменением сложением $75 + (-82)$ или $75 + [-82]_{\text{доп}}$. Дополнение 82 до 10^3 равно 18, тогда

$$\begin{array}{r}
 \begin{array}{l} 75 \text{ или } 0111 \ 0101 \ A \\ +18 \text{ или } +0111 \ 1110 \end{array} \\
 \hline
 \begin{array}{l} 1111 \ 0011 \\ \text{п} \leftarrow \uparrow \text{п} \leftarrow \end{array} \\
 \hline
 \begin{array}{l} 1010 \\ 1001 \ 0011 \end{array} \\
 \hline
 93 \quad 1001 \ 0011 \text{ Десятичное дополнение числа } -7
 \end{array}$$

Дополнение B (с избытком 6 в каждой тетраде)
Нескорректированный результат в дополни-
тельном коде
Коррекция

Отсутствие переноса из старшей тетрады при сложении A с дополнением $-B$ свидетельствует о том, что результат получился в виде дополнения до 10^3 . Так как первый операнд положительный, то результат имеет отрицательный знак. Для получения прямого кода такого результата он вычитается из нуля (прибавляется к нулю двоичное дополнение дополнения) и корректируется, т. е.

$$\begin{array}{r}
 \begin{array}{l} 00 \\ + [93]_{\text{доп}} \end{array} \\
 \hline
 \begin{array}{l} 0000 \ 0000 \\ +0110 \ 1101 \end{array} \\
 \hline
 \begin{array}{l} 0110 \ 1101 \\ +1010 \ 1010 \end{array} \\
 \hline
 07 \quad 0000 \ 0111 \text{ Прямой код результата (без знака)}
 \end{array}$$

Двоичное дополнение десятичного дополне-
ния результата
Коррекция

К полученному результату должен быть приформирован знак минус, т. е. должно получиться -7 .

В каждом конкретном случае правило выработки знака результата зависит от того, как реализуется сложение операндов с разными знаками. В предыдущем примере, например, в обоих случаях вычитался второй операнд. Это должно учитываться при выработке правила определения знака результата.

Формируемый код знака зависит от того, какой код (ДКОИ или КОИ-8) используется для кодирования входного алфавита (см. п. 2.2).

Количество байт десятичного числа, выбираемое в АУ за одно обращение к оперативной памяти, зависит от конструкции последней. В модели ЕС-1020 это — 2 байта, в ЕС-1030 — 4 байта, а в

ЕС-1050 — 8 байт. В ЕС-1030 и ЕС-1050 операнды полностью выбираются из оперативной памяти, а в ЕС-1020 все время находятся в ООП. В ЕС-1030 после выравнивания они записываются в специальную местную память процессора, в ЕС-1050 — хранятся в регистрах. Поля 1-го и 2-го операндов могут перекрываться, но при этом адреса младших байт, где находятся знаки, должны совпадать. Если длины полей операндов не совпадают, то имеющий меньшую длину дополняется слева необходимым количеством нулевых байт.

Если все значащие цифры суммы не помещаются в поле первого операнда, то возникает требование прерывания исполняемой программы по причине десятичного переполнения. Переполнение также фиксируется, если происходит потеря переноса из старшей цифровой позиции поля результата. Нулевой результат всегда имеет положительный знак.

В моделях ЕС-1030 и ЕС-1050 результаты, которые могут иметь длину до 16 байт, полностью формируются в прямом коде со знаками в процессоре. Только после этого они отсылаются в основную оперативную память. В модели ЕС-1020 результат по мере выполнения операции над байтами заносится в ООП. Если необходимо взятие его дополнения, то он вновь по 2 байта вызывается в процессор, где вычитается из нуля и опять запоминается в основной памяти.

Умножение. Процессы нахождения произведений целых десятичных чисел реализуются аналогично тому, как это делается при вычислениях вручную с помощью обычных арифмометров. В общем случае суть их состоит в том, что к определенным образом сдвинутой текущей сумме частных произведений множимое последовательно прибавляется столько раз, сколько единиц содержится в очередной анализируемой цифре множителя. Сомножители представляются в прямом коде. Умножение начинается с младших цифр множителя. После очередного прибавления множимого цифра множителя, на которую производится умножение, уменьшается на единицу.

Пример 3.23. Найти произведение 17×23 .

Исходная сумма	0000	Множитель: 23	
частных произведений:	+	17	
	+	0170	
	+	0340	
	+	0510	
Сдвинутая вправо сумма	+	0051	
частных произведений	+	0221	
	+	0391	

$\downarrow 3 - 1 = 2$
 $\downarrow 2 - 1 = 1$
 $\downarrow 1 - 1 = 0$

$\downarrow 2 - 1 = 1$
 $\downarrow 1 - 1 = 0$

Произведение

Количество десятичных разрядов произведения не превышает суммарного количества цифр сомножителей. Эта сумма не должна быть больше максимально допустимого числа цифр в произведении, равного 31. Знак результата определяется по алгебраическим правилам и устанавливается в его правой тетраде младшего байта в конце выполнения операции.

В модели ЕС-1020 в процессе накопления сумм частных произведений для ускорения умножения, кроме самого множимого, используются его удвоенные значения, а в модели ЕС-1030 — еще и учетверенные. Они получаются путем специальных действий сдвига множимого в начале операции при вызове из ООП первого операнда. Для удвоения множимое сдвигается на один двоичный разряд влево и, если необходимо, корректируется. В ЕС-1030 удвоенное значение аналогично сдвигается еще на разряд влево для получения учетверенной величины.

Например, удвоенные и учетверенные двоично-десятичные значения множимого 17 (0001 0111₂₋₁₀) будут:

множимое	0001 0111
сдвиг на один разряд влево	+ 0010 1110
коррекция	0110
удвоенное множимое	0011 0100 (17·2 = 34)
учетверенное множимое	0110 1000 (34·2 = 68)

Если до сдвига в каких-либо тетрадах находились цифры, большие 4, то после сдвига производится коррекция. К полученным значениям цифр в этих тетрадах прибавляются цифры 6 с учетом переносов в соседние слева старшие тетрады.

Как множимое А, так и его удвоенное 2А и учетверенное 4А значения на время операции хранятся в специально отведенных для них (рабочих) ячейках местной оперативной памяти. В процессоре модели ЕС-1030 при умножении на очередную цифру множителя, в зависимости от ее величины, выполняются действия, указанные в табл. 3.4.

Т а б л и ц а 3.4

Очередная цифра множителя	Вариант суммирования при накоплении суммы частных произведений в модели ЕС-1030	Действия при накоплении суммы частных произведений в модели ЕС-1020	Очередная цифра множителя	Вариант суммирования при накоплении суммы частных произведений в модели ЕС-1030	Действия при накоплении суммы частных произведений в модели ЕС-1020
0	0	0	5	+4А+1А	+2А+2А+1А
1	+1А	+1А	6	+4А+2А	-2А-2А
2	+2А	+2А	7	+4А+2А+1А	-2А-1А
3	+2А+1А	+2А+1А	8	+4А+4А	-2А
4	+4А	+2А+2А	9	+4А+4А+1А	-1А

Следуя этим указаниям, процесс умножения, рассмотренный в примере 3.23, можно представить следующим образом:

	0000	Множитель:	2 3
	+		
	34		↓
	-----		+2A
	+0340		
	17		↓
	-----		+1A
	0510		
Сдвиг	+		↓
	0051		+2A
	34		

Произведение	0391		

В модели ЕС-1050 принят другой способ ускоренного умножения, основанный на комбинировании действий сложения и вычитания. Цифры множителя также анализируются, начиная с младшей. Для очередной цифры, меньшей или равной 5, множимое столько раз подсуммируется к сдвинутой на четыре разряда вправо сумме частных произведений, сколько единиц содержится в анализируемой цифре. Для цифр от 6 до 9 множимое, наоборот, вычитается из сдвинутой на четыре разряда вправо накапливаемой суммы. Количество вычитаний равно числу единиц в дополнении данной цифры до 10. Перед вычитаниями к сдвинутой вправо сумме частных произведений предварительно подсуммируется множимое, увеличенное в 10 раз. Вычитание производится путем сложения суммы с дополнением множимого.

Пример 3.24. Способом комбинированных действий найти произведение 17×91 .

Множимое:	17	Множитель:	9 1
Исходная сумма частных произведений	+ 00000		
	17		↓
	-----		1 - 1 = 0
1-я сумма частных произведений	01700		
Сдвинутая 1-я сумма	+ 00170		↓
17 · 10	17	←	10 - 9 = 1
	-----		Дополнение
	17170		цифры мно-
	+		жителя
Дополнение множимого	983	←	1 - 1 = 0

2-я сумма частных произведений	15470		
Сдвинутая 2-я сумма	01547		

	01547		

— произведение.

Множимое или его дополнение прибавляется к текущей сумме частных произведений до тех пор, пока соответственно очередная цифра множителя или ее дополнение не будет сведено к нулю. Суммирование выполняется с использованием двоично-десятичных кодов с избытком шесть.

Метод комбинированных действий принят и в модели ЕС-1020. Характер выполняемых действий (сложения или вычитания) для

каждого значения цифры множителя указан в табл. 3.4. В этой модели, если цифра множителя, на которую производится умножение, больше 5, то следующая цифра множителя увеличивается на единицу.

Деление. Как и для целых двоичных чисел, при делении целых десятичных чисел получается частное и остаток. Частное помещается в левую, а остаток — в правую части поля первого операнда. Частное имеет длину $L_1 - L_2$ (см. рис. 2.12), а остаток L_2 .

Многошаговые процессы десятичного деления в различных моделях ЭВМ Единой системы имеют свои особенности реализации. В модели ЕС-1030 оно выполняется методом последовательного вычитания делителя из делимого с восстановлением остатка. После каждого вычитания, если остаток неотрицательный, формируемая цифра частного увеличивается на единицу.

Пример 3.25. Пусть требуется разделить 78 на 6.

Исходное делимое	$+078$	
Дополнение делителя	$+94$	
	$+018$	→ 1 Первая цифра частного
	$+94$	
Остаток отрицательный	$+958$	
	$+06$	
Восстановленный остаток	018	
Сдвинутый влево остаток	$+180$	
	$+94$	
	$+120$	→ 11
	$+94$	→ 12
	$+060$	→ 13
	$+94$	
	000	
		} Формирование второй цифры частного
		}
		} <u>частное</u>

В данном примере вычитание делителя 06 заменяется прибавлением дополнения 94. Второй остаток получился отрицательным. Для восстановления к нему пришлось прибавить делитель.

В моделях ЕС-1020 и ЕС-1050 деление целых десятичных чисел выполняется по методу без восстановления остатка. Суть его состоит в следующем. Полученный при последовательных вычитаниях отрицательный остаток указывает на то, что при предыдущем вычитании закончено формирование определяемой цифры частного. При этом остаток не восстанавливается, а меняется вид действий: делитель сдвигается на десятичный разряд вправо и начинается его циклическое прибавление к отрицательному остатку. После первого прибавления в очередном разряде частного устанавливается цифра 9. При каждом последующем сложении содержимое этого разряда уменьшается на единицу. Появление теперь положительного остатка означает, что получена очередная цифра

частного. Далее делитель снова сдвигается и т. д. Процесс деления продолжается до тех пор, пока не будут получены значения всех разрядов частного.

Пример 3.26. Выполним процесс деления 162 на 6 по методу без восстановления остатка.

Исходное делимое	+	162	→	0
Дополнение делителя	+	94		
	+	102	→	1
	+	94		
	+	042	→	2
	+	94		
Сдвиг делителя	+	982		
		6		
	+	988	→	29
	+	6		
	+	994	→	28
	+	6		
		000	→	27

Знак частного определяется по правилам алгебры путем анализа знаков делимого и делителя. Знак остатка тот же, что и у делимого. В начале операции, как и при делении целых двоичных чисел, производится проверка корректности десятичного деления. Некорректным считается получение частного, у которого количество цифр больше допустимого, т. е. оно вместе с остатком не помещается на месте первого операнда. Для проверки корректности перед началом собственно деления производится пробное вычитание делителя из делимого. При этом делитель относительно делимого смещается так, что его старшая цифра оказывается расположенной под второй слева цифрой делимого. Требование прерывания возникает тогда, когда результат пробного вычитания положителен или равен нулю.

3.5. ЛОГИЧЕСКИЕ ОПЕРАЦИИ И ОСОБЕННОСТИ ИХ ВЫПОЛНЕНИЯ

Кроме арифметических операций, в группах команд всех форматов (см. приложение, табл. 10) имеются команды для логической обработки данных. В ЕС ЭВМ предусмотрена возможность выполнения логических операций сопоставления величин двоичных кодов (с целью выявления их равенства или неравенства), поразрядного логического умножения, логического сложения или суммирования по модулю два (соответственно операции *И*, *ИЛИ* и *Исключающее ИЛИ*), проверки, перекодирования, редактирования, сдвига и специальных видов пересылки.

Только в операциях редактирования данные — десятичные числа — рассматриваются как числа. В остальных логических

операциях операнды трактуются просто как группы восьмиразрядных двоичных кодов (байт), не имеющих количественного значения. В некоторых операциях действия производятся отдельно над левыми и правыми тетрадами байт. Байты операндов обрабатываются слева направо. Кроме сдвигов и пересылок, после каждой из остальных логических операций устанавливается код условия [5].

Соответственно форматам команд логическая информация может быть как фиксированной длины (8, 24, 32 или 64 двоичных разряда), так и переменной — до 256 байт. Во всех операциях, кроме редактирования, операнды имеют одинаковую длину.

Операции *И*, *ИЛИ* и *Исключающее ИЛИ* выполняются поразрядно. Над одноименными разрядами x_i и y_i полей операндов выполняются операции соответственно логического умножения $x_i y_i$, логического сложения $x_i \vee y_i$ или суммирования по модулю 2 ($x_i \oplus y_i = x_i \bar{y}_i \vee \bar{x}_i y_i$). Результат действия над каждой парой разрядов помещается в поле результата, замещающего первый операнд, в i -ю разрядную позицию. Если во всех разрядах поля результата оказываются записанными нули, то признак результата $ПР=0$; при наличии единицы хотя бы в одном разряде — $ПР=1$.

Операция сопоставления (сравнения) величин операндов также выполняется поразрядно. Начиная слева, коды, расположенные в двух полях, сравниваются как некоторые двоичные числа. Как только обнаружатся несовпадающие разряды, выполнение операции прекращается. При этом определяется большее из двух сравниваемых чисел. Если первый операнд больше второго, то $ПР=2$, в противном случае — $ПР=1$; при равенстве операндов — $ПР=0$.

Из остальных логических операций рассмотрим лишь назначение и особенности выполнения операций перекодирования и редактирования.

Перекодирование осуществляется на основе просмотра таблицы — словаря, в соответствии с которой байты первого операнда замещаются байтами второго операнда. Команды ПЕРЕКОДИРОВАТЬ имеют формат SS второго типа (рис. 2.12). База B_2 и смещение D_2 такой команды определяют начальный адрес словаря. Байты первого операнда называют аргументами, а второго — функциями. В процессе выполнения команды каждый байт — аргумент первого операнда, начиная с первого (считая слева направо), прибавляется к младшим разрядам начального адреса словаря. В результате последовательно формируются адреса, по которым из таблицы-словаря выбираются байты-функции, замещающие байты в поле первого операнда.

Словарь может содержать до 256 байт-функций. Это могут быть коды символов, соответствующие, например, таблицам кодирования КОИ-8, ДКОИ и т. п. Поэтому с помощью операции перекодирования можно осуществлять перевод буквенно-цифровой информации, занимающей поля до 256 байт, из одного кода в другой.

В таблице перекодирования (словаре) коды символов-функций размещаются так, чтобы их порядковые номера, начиная от границы, определяемой адресом $(B_2) + D_2$, равнялись числам, которыми закодированы символы-аргументы. Предположим, что осуществляется перевод данных из КОИ-8 в ДКОИ и очередным анализируемым символом-аргументом является буква t , согласно КОИ-8 (табл. 2.4) кодируемая условным числом 0111 0100 (116_{10}). В словаре код этой буквы, согласно ДКОИ (табл. 2.5) равный 1010 0011 (163_{10}), должен располагаться в 116 байте, начиная от границы, определяемой адресом второго операнда.

Одна из разновидностей рассмотренной операции — перекодирование с проверкой, позволяет не только заменять байты-аргументы байтами-функциями, но и разбивать тексты на предложения, выявлять знаки пунктуации. Такие действия необходимы для нахождения ограничителей при просмотре массивов данных в процессе транслирования программ и т. п.

Операции типа редактирования используются для приведения десятичных чисел к удобочитаемому виду и подготовки их таким образом к выдаче, например, на печать. В процессе выполнения операции второй операнд редактируется по образцу первого. Одновременно с этим редактируемые десятичные числа преобразуются из упакованного формата в формат с зоной (распаковываются). Результат записывается в основную память по первому адресу на место образца.

Собственно редактирование дает возможность уничтожать в числе незначащие нули слева, вставлять в определенные места разделители (десятичные запятые, точки), а также любые другие символы, заполнять пробелами нулевые поля или их части. В процессе редактирования можно различать нулевые, положительные и отрицательные числа и соответственно им устанавливать код условия. Если одной командой редактируется несколько числовых полей, то код условия относится только к последнему числу.

Команда ОТРЕДАКТИРОВАТЬ имеет формат SS второго типа (рис. 2.12). Длина L операнда, задаваемая командой, относится к образцу. Один байт образца дает возможность отредактировать одну цифру редактируемых данных. Поэтому второй операнд — редактируемые цифровые данные — обычно короче первого — образца.

Редактируемые данные называют исходными. Они представляют собой десятичные числа в упакованном формате, не содержащие разделителей. Например, смешанное положительное число 127,305 может быть размещено в четырех байтах следующим образом: 01 27 30 5С. Символом С закодирован знак плюс. Местоположение запятой в таком представлении подразумевается.

Первый операнд — образец, по которому происходит редактирование, еще называют шаблоном. Его поле содержит служебные

символы ДКОИ, которые указывают, как в него последовательно должны перемещаться цифры исходного числа. Одни символы служат для управления пересылкой и размещением цифр, другие сами заполняют определенные позиции в поле образца, третьи управляют процессом редактирования. Эти функции выполняют символы (см. приложение, табл. 9 и табл. 2.5):

- 1) заполнитель;
- 2) выбор цифры (ВЦФ);
- 3) начало значимости (НЗН);
- 4) разделитель полей (РП).

В качестве заполнителя обычно используется пробел. Как правило, этим символом начинается поле образца.

Символ выбора цифры управляет перемещением цифры исходного поля в поле образца. Действие этого символа не выходит за пределы одного байта образца. Если исходная цифра — значащий нуль, то она заменяется символом-заполнителем. Применяя в качестве заполнителя пробел, можно убирать нули перед старшими значащими цифрами чисел.

Символ начала значимости также замещается либо цифрой исходного поля, либо символом-заполнителем. Однако он имеет групповое действие. Выполняя те же функции, что и символ выбора цифры, он указывает, что все последующие цифры — значащие, в том числе и нули. Поэтому все последующие символы образца замещаются исходными цифрами.

Символ разделения полей используется в тех случаях, когда с помощью одной команды редактируется несколько полей упакованных десятичных чисел. Сам этот символ всегда замещается символом-заполнителем. Кроме того, он указывает, что далее идет следующее цифровое поле. Символы пробела, выбора цифры, начала значимости и разделения полей не имеют печатаемых изображений. Поэтому для записей полей образцов их часто обозначают соответственно как *b*, *d*, (*,*). В поле образца на первом месте слева всегда находится символ-заполнитель. Он сохраняется неизменным до конца операции.

В ходе редактирования оба операнда (исходный и образец) обрабатываются последовательно символ за символом. В результате анализа очередной пары символов осуществляется одно из трех действий.

1. К цифре исходных данных добавляется зона, и полученным символом замещается символ образца.

2. В данный байт образца помещается символ-заполнитель.

3. Символ образца не меняется.

В последнем случае в байте образца сохраняется специальный целевой символ. К таким символам относятся, например, символы-разделители (запятые, точки и т. п.), условные обозначения и т. п., вставляемые в результат. В поле образца их в необходимых местах чередуют с перечисленными выше управляющими символами.

Операция редактирования фактически начинается с анализа символа, следующего за символом-заполнителем в поле образца. Чаще всего им оказывается символ (d) выбора цифры. Он замещается либо первой цифрой исходного поля, либо символом-заполнителем. Для того чтобы определить вид замещения, анализируется значение цифры исходного поля. Значащая цифра помещается в поле образца, а вместо незначащей (нуля) ставится символ-заполнитель (делается пробел).

Факт обнаружения первой значащей цифры в исходном поле запоминается специальным триггером значимости числа (ТЗНЧ). Триггер значимости при этом устанавливается в единичное состояние и выдает сигнал $s = 1$. Пока этот триггер находится в единичном состоянии, все цифры исходного поля, включая нули, последовательно помещаются вместо символов d в поле образца. Перед началом анализа исходного поля ТЗНЧ находится в нулевом состоянии, выдавая сигнал $s = 0$. Поэтому до обнаружения первой значащей цифры символы d замещаются символами-заполнителями, например, пробелами.

Рассмотрим пример. Предположим, что имеем	
исходное поле	0 1 2 7 3 0 5 С и
поле образца	b d d d d , d d d d
Сигнал s принимает значения:	0 1 1 1 1 1 1 1 0.
Тогда результат будет:	b b 1 2 7 , 3 0 5 b.

Так как при выдаче на печать пробел — это пустое место, то будет отпечатано: 1 2 7, 3 0 5.

В примере первый 0 в исходном числе — незначащий, поэтому при его анализе ТЗНЧ останется в нулевом состоянии ($s = 0$), и первый символ d заменяется пробелом b . При анализе следующего символа d и очередной цифры исходного поля (1) вырабатывается $s = 1$ и цифра 1 помещается в поле образца.

Если триггер значимости ТЗНЧ остается в единичном состоянии, то в поле образца все символы, не относящиеся к управляющим, остаются на своих местах. Поэтому запятая остается в позиции, следующей за цифрой 7. При просмотре символа, соответствующего запятой, очередная цифра исходного поля (в данном случае 3) не анализируется. Ее анализ производится при выборе для просмотра следующего символа образца — d . Так как $s = 1$, то цифра 3 перемещается в поле образца. Аналогично, так как сохраняется $s = 1$, в поле образца помещается и следующая за цифрой 3 цифра 0 и т. д.

Триггер ТЗНЧ возвращается в нулевое состояние при анализе цифры С, означающей плюс. Поэтому в последней позиции символ d замещается пробелом b .

Выполнение команды заканчивается по заполнению поля образца отредактированными данными.

Признак результата устанавливается по значимости отредактированного числа и его знаку. Результатам, равным, меньшим и большим нуля, соответствуют признаки результата, соответственно имеющие значения 0, 1 и 2. Последние два значения вырабатываются по сигналу s . Знаком плюс триггер ТЗНЧ возвращается в нулевое состояние и s становится равным 0, а при знаке минус единичное состояние ТЗНЧ остается неизменным, т. е. сохраняется $s = 1$. Переключения триггера значимости происходят и при обнаружении в образце символов начала значимости и разделителя полей. Подробное описание применения этих символов можно найти, например, в [5].

ЭЛЕМЕНТНАЯ БАЗА И УНИФИКАЦИЯ КОНСТРУКЦИЙ ЭЛЕКТРОННОЙ АППАРАТУРЫ УСТРОЙСТВ ЭВМ

4.1. ТИПОВЫЕ КОНСТРУКТИВНЫЕ МОДУЛИ

При конструировании электронной аппаратуры ЭВМ Единой системы предусмотрено применение унифицированной конструктивно-технологической базы, основанной на широком использовании современных достижений микроэлектроники. Электронная аппаратура строится по блочному принципу. Основными ее элементами являются монолитные интегральные схемы (ИС).

Применение интегральных схем позволило использовать новые конструктивные решения. Теперь простейшей составной частью электронной аппаратуры машин и периферийных устройств стал модуль интегральной схемы 1 (рис. 4.1). Модули ИС монтируются на платах с печатным монтажом. Применяются двусторонние и многослойные печатные платы. Несущие модули ИС платы выполняются съемными. Они являются простейшими типовыми элементами замены (ТЭЗ) 2. Для удобства смены ТЭЗ снабжаются разъемами, с помощью которых они соединяются в блоки 3, получившие название блоков панелей.

Несущим элементом (шасси) конструкции блока панели является плоская многослойная печатная плата. На ней в определенном порядке крепятся разъемы и направляющие для ТЭЗ. Блок панели — коммутационная печатная плата с разъемами и направляющими — также представляет собой законченную конструктивно-монтажную часть конструкции, в которой обычно в два ряда устанавливается 40 ТЭЗ.

Блоки панелей крепятся в рамках 4, представляющих собой прямоугольные каркасы сварного типа из нормализованных профилей с перемычками и направляющими для установки блоков панелей. В рамках также размещаются блоки питания. Для осуществления межрамных соединений на рамках крепятся разъемы.

Рамы помещаются в защитные кожухи. В зависимости от размеров и назначения устройства рама с кожухом оформляются в виде шкафа 5, тумбы или стола.

Для более плотной компоновки электронного оборудования цифровой машины в шкафу (стойке) помещаются три рамы: две поворотные А и С и одна неподвижная В. С помощью неподвижной средней рамы обеспечивается жесткость конструкции стойки.

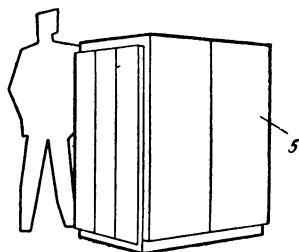
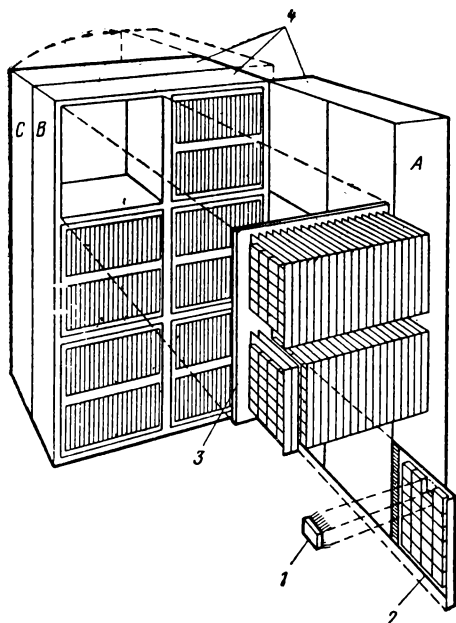


Рис. 4.1. Вариант типовой конструкции стойки ЭВМ:

1 — модуль интегральной схемы; 2 — типовой элемент замены (ТЭЗ); 3 — блок панели; 4 — рамы; 5 — стойка, оформленная в виде шкафа

Все рассмотренные составляющие части конструкций машин обладают конструктивной завершенностью и могут изготавливаться независимо друг от друга. Это обусловило целесообразность унификации таких частей, снабженных средствами механического и электрического взаимного сопряжения. Унифицированные части получили название *конструктивных модулей* того или иного уровня. Уровень модуля определяется количеством типов объединяемых в его составе более простых модулей. Самым простым модулем является наименьшая физически неделимая единица ЭВМ, имеющая стандартное конструктивное оформление. Такой единицей является, например, интегральная схема, электрорадиодеталь. Все остальные рассмотренные выше конструктивные модули являются сложными. В порядке старшинства они располагаются так, как дано в табл. 4.1.

Унификация конструктивных модулей различных уровней в наибольшей мере способствует удовлетворению требованиям, предъявляемым к конструкциям современных серийно выпускае-

Т а б л и ц а 4.1

Конструктивные модули

Уровень конструктивного модуля	Тип модуля
1	Интегральная схема (корпус ИС), электрорадиодеталь
2	Типовой элемент замены (ТЭЗ)
3	Блок панели
4	Рама
5	Стойка (шкаф), тумба, пульт

мых ЭВМ. Она позволяет применять прогрессивные технологические процессы, ведет к сокращению циклов производства и наладки машин, обеспечивает возможность быстрой модернизации и налаживания выпуска обширной номенклатуры машин без существенной перестройки процесса серийного производства, облегчает настройщикам, регулировщикам и эксплуатационникам освоение различных модификаций выпускаемых машин.

4.2. ОСНОВНЫЕ КОМПЛЕКСЫ ИНТЕГРАЛЬНЫХ СХЕМ, ПРИМЕНЯЕМЫХ В МОДЕЛЯХ ЭВМ ЕДИНОЙ СИСТЕМЫ

В моделях ЕС-1020—ЕС-1040 электронная аппаратура процессоров и каналов строится на интегральных транзисторно-транзисторных логических схемах (ТТЛ) серии 155. На ТТЛ-схемах этой серии также выполняются устройства управления аппаратуры ввода-вывода и накопителей всех моделей. В процессоре и каналах модели ЕС-1050 применяются более быстродействующие элементы — интегральные транзисторные логические схемы на переключателях тока с объединенными эмиттерами, получившие название эмиттерно-связанных логических схем (ЭСЛ-схемы). Используются ЭСЛ-схемы двух серий: 137 и 187.

Транзисторно-транзисторные логические схемы серии 155. Интегральные схемы этого типа изготавливаются на пластинах кремния по планарно-эпитаксиальной технологии. Кремниевые пластины с выполненными на них микросхемами заключаются в герметические пластмассовые корпуса *I* прямоугольной формы с 14-ю штыревыми выводами (рис. 4.1).

Функционально полную основу данного комплекса интегральных схем составляют универсальные элементы типа *И—НЕ*. Они являются логическими базовыми схемами, сокращенно обозначаемыми ЛБ. Разновидности элементов *И—НЕ* имеют дополнительные входы, позволяющие строить логические схемы *И—ИЛИ—НЕ*, расширяемые по входу ИЛИ. Принципиальная схема и условное изображение такого логического расширяемого элемента ЛР показаны на рис. 4.2, *a* и *б*.

Принципиальная схема содержит двухэмиттерный транзистор *T1*, на котором выполнена двухходовая входная схема *И*, и сложный инвертор, собранный на транзисторах *T2—T4*, диоде *D* и резисторах *R2—R4*. При напряжении питания $E_k = +5$ В высокий уровень выходного напряжения $U_1 \geq +2,4$ В, а низкий — $U_0 \leq +0,4$ В. Подаваемые на входы элемента напряжения u_{x1} и u_{x2} меняются в таких же пределах. Формы входного и выходного сигналов изображены на рис. 4.3. Обычно в ТТЛ-схемах высокий уровень напряжения U_1 ставится в соответствие коду 1, а низкий — U_0 — коду 0.

В модификациях базовых элементов комплекса имеются варианты, в которых входной транзистор *T1* имеет большее число эмит-

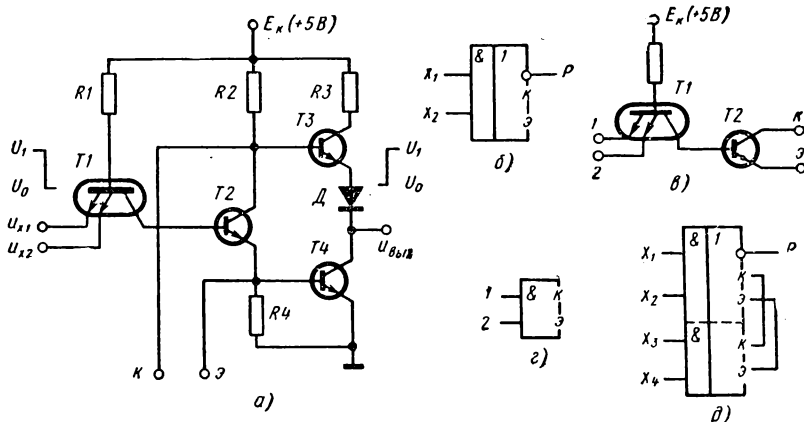


Рис. 4.2. Транзисторно-транзисторные логические схемы:

а, б — принципиальная схема и условное изображение элемента И—НЕ; в, г — принципиальная схема и условное изображение расширителя схемы И—ИЛИ—НЕ по входу ИЛИ; д — пример схемы И—ИЛИ—НЕ с подключенным расширителем

теров, например, 3, 4, 8. В зависимости от этого входная схема совпадения имеет соответствующее количество входов. Она работает следующим образом. Если на все эмиттеры транзистора Т1 подаются высокие уровни потенциала U_1 , то переходы база-эмиттер оказываются смещенными в обратном направлении, а переход база — коллектор — в прямом. Через резистор R1 и смещенный в прямом направлении переход база-коллектор протекает прямой ток базы транзистора Т2. Этим током последний открывается и вводится в режим насыщения. Одновременно полностью открывается и транзистор Т4, на котором выполнен выходной инвертор. С коллектора транзистора Т4 снимается низкий уровень выходного напряжения U_0 . Транзистор Т3 при этом находится в закрытом состоянии. Оно обеспечивается низким уровнем напряжения, установившимся на коллекторе транзистора Т2. Кроме того, надежному запираению Т3 способствует наличие диода Д. Это объясняется следующим образом.

При открытом транзисторе Т4 на его коллекторе устанавливается напряжение $+(0,1 \div 0,4)$ В, а на коллекторе открытого транзистора Т2 — напряжение $U_{б.эТ4} + U_{к.эТ2} \approx 0,7$ В $+(0,1 \div 0,4)$ В $= (0,8 \div 1,1)$ В. Разность между напряжениями

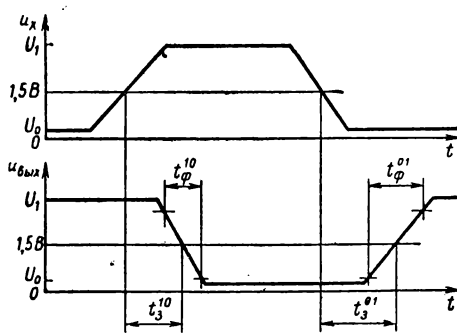


Рис. 4.3. Формы входного u_x и выходного $u_{\text{вых}}$ сигналов ТТЛ-элемента

на коллекторах транзисторов Т2 и Т4 составляет примерно 0,7 В. Этого перепада оказывается недостаточно для отпирания перехода база-эмиттер транзистора Т3 и диода Д, поскольку порог срабатывания только каждого из них составляет примерно (0,7 ÷ 0,8) В.

Схема переключается в состояние, при котором на выходе устанавливается высокий уровень напряжения U_1 , если хотя бы на один из эмиттеров транзистора Т1 подается напряжение низкого уровня U_0 . При этом соответствующий переход база-эмиттер транзистора Т1 смещается в прямом направлении, его базовый ток переключается в эмиттерную цепь и транзистор Т1 входит в насыщение. Сопротивление перехода коллектор-эмиттер транзистора Т1 резко падает, что обеспечивает быстрое рассасывание не основных носителей из базы транзистора Т2 и его запираение. Следом за Т2 закрывается и транзистор Т4. Транзистор Т3 открывается, так как напряжение на коллекторе закрывающегося транзистора Т2 стремится к E_k . При открытом транзисторе Т3 выходное напряжение $u_{\text{вых}}$ становится равным U_1 . Таким образом, при принятом выше условии о соответствии уровней U_1 и U_0 сигналам кода 1 и 0 схема реализует логическую функцию $P = \overline{x_1 x_2}$, где x_1, x_2, P соответственно $u_{x1}, u_{x2}, u_{\text{вых}}$. Сложный инвертор имеет достаточно высокую нагрузочную способность. Интегральные схемы серии 155, подобные рассмотренной, в среднем характеризуются следующими основными параметрами:

$$t_3^{10} = (15 \div 18) \text{ нс}; P_{\text{ср}} = (25 \div 35) \text{ мВт};$$

$$t_3^{01} = (25 \div 33) \text{ нс}; m \leq 8;$$

$$t_{\Phi}^{10} = (14 \div 16) \text{ нс}; l \leq 8;$$

$$t_{\Phi}^{01} = (18 \div 22) \text{ нс}; n \leq 10,$$

где t_3^{10} и t_3^{01} — задержки переключения выходного напряжения соответственно с уровня U_1 на уровень U_0 и с U_0 — на U_1 ;

t_{Φ}^{01} и t_{Φ}^{10} — длительности фронта и среза выходного сигнала;

$P_{\text{ср}}$ — средняя мощность рассеяния на один логический элемент;

m и l — коэффициенты объединения соответственно по входам И и ИЛИ;

n — нагрузочная способность.

Для того чтобы можно было строить схемы типа И—ИЛИ—НЕ, у расширяемых логических элементов предусматриваются дополнительные входы, на рис. 4.2, а обозначенные буквами κ и ε (выходы коллектора и эмиттера). Через эти входы к базовым схемам могут подключаться одноименные выходы логических подключаемых (ЛП) схем-расширителей по входам ИЛИ. Принципиальная схема двухвходового расширителя и ее условное изображение приведены на рис. 4.2, в, г. В результате параллельного

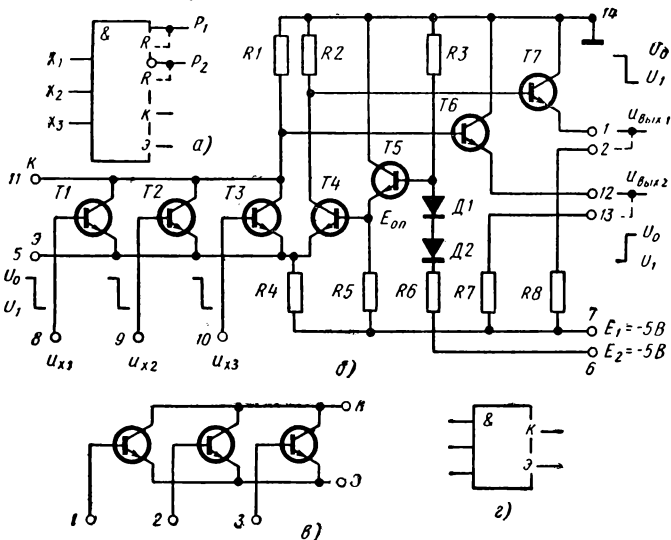


Рис. 4.4 Схема элемента на переключателе тока со связанными эмиттерами:

а, б — условное изображение и принципиальная схема универсального элемента И, И—НЕ; в, г — принципиальная схема и условное изображение расширителя для схемы И

включения транзисторов Т2 элементов ЛР и ЛП получаются логические схемы, подобные изображенной на рис. 4.2, д. Эта схема реализует логическую функцию $P = x_1 x_2 \sqrt{x_3 x_4}$. Применяя совместно расширяемые и подключаемые схемы с разными m и l , в пределах указанных выше возможностей, можно реализовать различные логические функции, выраженные в дизъюнктивной форме.

Комплекс серии 155 в своем составе содержит обширный набор модификаций интегральных схем, перечень которых и их технические и электрические параметры можно найти, например, в справочнике [31].

Транзисторные схемы со связанными эмиттерами (типа ЭСЛ). Схемы этого типа также изготавливаются по планарно-эпитаксиальной технологии. Корпусы микросхем — прямоугольные пластмассовые с 14-штыревыми выводами, как и у ИС серии 155.

Функционально полную основу комплексов элементов на переключателях тока составляют универсальные элементы типа И, И—НЕ, имеющие парафазные выходы. Принципиальная схема такого базового элемента приведена на рис. 4.4, б, а его условное изображение — на рис. 4.4, а. В данном элементе входная половина токового переключателя выполнена на транзисторах Т1—Т3 (рис. 4.4, б), другая — на транзисторе Т4. На транзисторе Т5, диодах Д1, Д2 и резисторах R3, R5 и R6 собран источник опор-

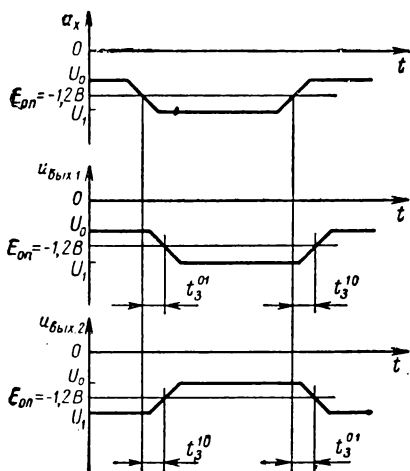


Рис. 4.5. Формы входного u_x и выходных $u_{\text{вых}1}$, $u_{\text{вых}2}$ сигналов элемента типа ЭСЛ

ходных сигналов элемента типа ЭСЛ приведены на рис. 4.5.

При низком уровне напряжения U_1 на всех трех входах $\delta-10$ (рис. 4.4, б) транзисторы Т1—Т3 закрыты, а транзистор Т4 открыт. Так будет до тех пор, пока на базу хотя бы одного из транзисторов Т1—Т3 не поступит сигнал высокого уровня U_0 . В этом случае транзистор, на вход которого поступил сигнал высокого уровня, открывается, а транзистор Т4 запирается. Ток, протекающий через транзистор Т4, переключается в открывшийся транзистор.

В кремниевых транзисторах падение напряжения на переходе база—эмиттер примерно равно $(0,7 \div 0,8)$ В. Примем это падение напряжения равным 0,7 В. Тогда при входном сигнале $U_0 = -0,95$ В, на эмиттерах транзисторов Т1—Т4 будем иметь напряжение $U_0 - U_{\text{б.э}} = -0,95 - 0,7 = -1,65$ В. При этом разность потенциалов между $E_{\text{оп}} = -1,2$ В и этим напряжением будет равна $E_{\text{оп}} - (U_0 - U_{\text{б.э}}) = -1,2 + 1,65 = 0,45$ В. Она недостаточна для отпирания транзистора Т4, так как порог его срабатывания находится в диапазоне $0,7-0,8$ В.

Если же, наоборот, открыт транзистор Т4, то на объединенных эмиттерах будет $E_{\text{оп}} - U_{\text{б.э}} = -1,2 - 0,7 = -1,9$ В. Это имеет место, когда на все входы подаются, например, сигналы уровней $U_1 = -1,45$ В. В этом случае разности потенциалов $U_1 - (E_{\text{оп}} - U_{\text{б.э}}) = -1,45 + 1,9 = 0,45$ В не хватает для отпирания транзисторов Т1—Т3.

В схемах рассматриваемого типа транзисторы работают в насыщенном режиме. Это способствует высокой скорости переключения таких элементов. Базовые элементы серии 137 характеризуются средним временем задержки распространения сигнала

ного напряжения $E_{\text{оп}} = -1,2$ В. Выходные сигналы элемента выдаются через эмиттерные повторители, выполненные на транзисторах Т6, Т7 и резисторах R7 и R8. При питающих напряжениях $E_1 = E_2 = -5$ В схема переключается входными сигналами и выдает на выходах сигналы, изменяющиеся от $U_0 = - (0,7 \div 0,95)$ В до $U_1 = - (1,45 \div 1,9)$ В. Эти значения практически симметричны относительно уровня $-1,2$ В, т. е. переключение схемы происходит входными сигналами с амплитудами $+ (0,25 \div 0,4)$ и $- (0,25 \div 0,7)$, симметричными относительно опорного напряжения.

Формы входного и вы-

лов $t_{з. ср} = 5$ нс при средней мощности рассеяния $P_{ср} = 70$ мВт, а серии 187 — $t_{з. ср} = 8$ нс при $P_{ср} = 35$ мВт.

Выходные эмиттерные повторители, формирующие выходные сигналы, обеспечивают высокую нагрузочную способность элементов $n \leq 15$. Они служат развязкой между переклюкателями тока и нагрузкой. Кроме того, они обеспечивают симметричность выходных сигналов относительно уровня — 1,2 В. На коллекторах транзисторов Т1—Т3 и Т4 средние уровни переключаемых напряжений оказываются смещенными в сторону повышения по отношению к средним уровням входных сигналов. Это смещение компенсируется смещением уровней выходных сигналов в сторону понижения за счет падения напряжений на переходах база—эмиттер транзисторов Т6 и Т7.

Если низкие уровни потенциала U_1 поставить в соответствие коду 1, а высокие U_0 — коду 0, то первый выход схемы (клеммы 1, 2) оказывается связанным с входами логическим соотношением $P_1 = x_1 x_2 x_3$, а второй (клеммы 12, 13) — соотношением $P_2 = x_1 x_2 x_3$, где x_1 соответствует u_{x1} , x_2 — u_{x2} , ..., P_2 — $u_{вых. 2}$. Очевидно, что $P_2 = \bar{P}_1$.

Для возможности увеличения количества входов у схем рассматриваемого типа в комплексах интегральных схем данных серий предусмотрены расширители. Пример схемы расширителя на три входа показан на рис. 4.4, в. При необходимости увеличения числа входов у схемы типа И, И—НЕ один или два трехвходовых расширителя клеммами κ и ε соединяются с одноименными клеммами токового переключателя. Пример такого включения показан на рис. 4.6. Предельное число получаемых таким образом входов у элементов обеих серий может достигать до $M = 9$.

Элементы на токовых переключателях могут рассматриваться и как схемы типа ИЛИ, ИЛИ—НЕ, если коду 1 поставить в соответствие высокий уровень потенциала, а коду 0 — низкий. При таком рассмотрении элементы входы у них изображаются инверсными, как показано на рис. 4.7. В этом случае выходные пере-

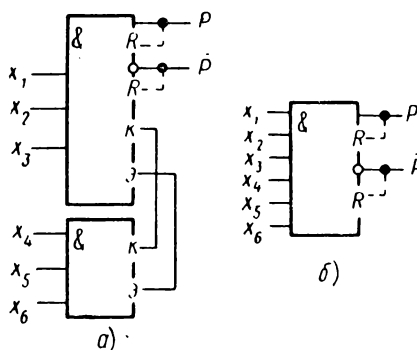


Рис. 4.6. Схема подключения расширителя:

а — логическая схема; б — условное изображение элемента И, И—НЕ с увеличенным числом входов

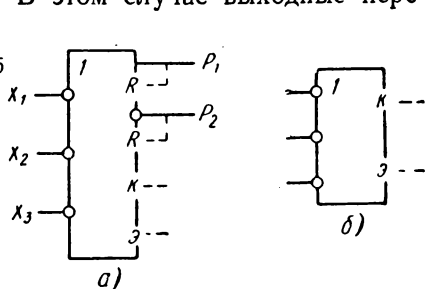


Рис. 4.7. Условные изображения элемента ИЛИ, ИЛИ—НЕ (а) и расширителя для схемы ИЛИ (б)

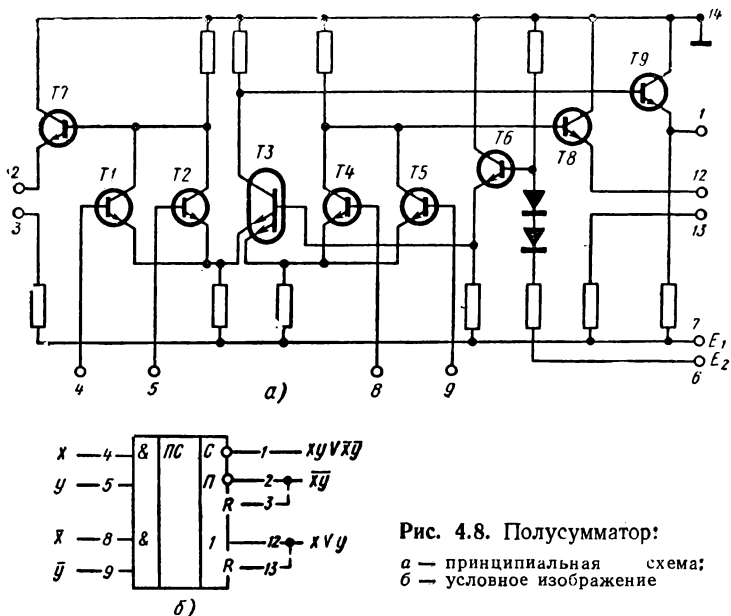


Рис. 4.8. Полусумматор:
 а — принципиальная схема;
 б — условное изображение

менные P_1 и P_2 на рис. 4.7, а будут связаны с входными $x_1 — x_3$ логическими соотношениями: $P_1 = \overline{x_1}\sqrt{x_2}\sqrt{x_3}$ и $P_2 = \overline{x_1}\sqrt{x_2}\sqrt{x_3}$. Здесь также $P_2 = \overline{P_1}$.

Для расширения входов схем ИЛИ применяются такие же расширители, как в схемах И. Поэтому допустимый предел расширения входов токовых переключателей выше был обозначен буквой M , в отличие от m — допустимого коэффициента объединения только по схеме И.

В комплексах ИС серий 137 и 187 имеется несколько модификаций элементов И, И—НЕ, различающихся количествами и типами входов и выходов, а также нагрузочной способностью. Кроме того, в комплексах предусмотрены различные модификации триггеров, а в серии 137 — еще два варианта таких схем, как полусумматор.

Принципиальная схема полусумматора приведена на рис. 4.8, а. Она состоит из двух двухвходовых токовых переключателей с общим опорным плечом, выполненном на двухэмиттерном транзисторе ТЗ. На входы 4 и 5 левой половины схемы подаются суммируемые переменные x и y , а на входы 8 и 9 правой половины — их инверсии. Транзистор ТЗ будет заперт только тогда, когда будут открыты по одному из транзисторов в парах: Т1, Т2 и Т4, Т5. Это соответствует случаю, когда переменные неравнозначны. При таком их сочетании на выходе 1 будет выдаваться высокий уровень потенциала, соответствующий значению $\overline{xy}\sqrt{xy} = xy\sqrt{xy}$, как показано на условном изображении полусумматора на рис. 4.8, б. Здесь, как и ранее, кодам 1 на входах и выходах ста-

вится в соответствие низкий уровень потенциала. На условном изображении выход инверсного значения суммы по mod 2, соответствующий клемме 1 на рис. 4.8, а, обозначен буквой С. На выходах П (клеммы 2, 3) и 1 (клеммы 12, 13) формируются сигналы, соответствующие инверсии логического произведения xy и логической сумме $x \vee y$.

При построении логических схем на токовых переключателях допускается объединение прямых и инверсных выходов одних элементов с прямыми и инверсными выходами других, а также прямого и инверсного выходов одного элемента. Допустимое количество объединяемых выходов обозначается буквой L . Допускается объединение выходов элементов только одной и той же серии. Для обеих серий $L \leq 5$.

При непосредственном соединении выходов эмиттерных повторителей (см. рис. 4.4, б) для сохранения уровней U_1 и U_0 выходных сигналов общее эмиттерное сопротивление не должно меняться. В случае соединения эмиттеров нескольких транзисторов надо иметь возможность использовать только один резистор R7 или R8. Поэтому в интегральных схемах типа ЭСЛ эмиттеры выходных эмиттерных повторителей и точки подключения соответствующих им резисторов выведены на отдельные клеммы корпусов ИС (например, 1, 2 и 12, 13).

Если объединить прямые выходы элементов И, И-НЕ (рис. 4.9, а), то получается вторая ступень И. Так как в таких схемах коду 1 ставится в соответствии низкий уровень потенциала, то объединенная схема реализует логическую формулу $P_1 = x_1 x_2 x_3 x_4 x_5 x_6$.

На функциональных схемах непосредственное соединение выходов нескольких элементов, благодаря которому реализуются новые логические связи, обозначают специальным элементом — псевдоэлементом ТОЧКА (сокращенно ТЧ, ТЧК)*. Изображение

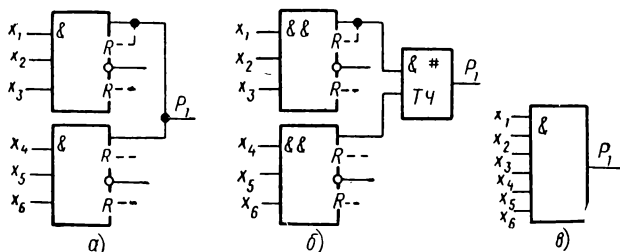


Рис. 4.9. Схема объединения прямых входов элементов И, И-НЕ:

а — непосредственное изображение объединения; б — изображение объединения с помощью псевдоэлемента; в — условное изображение расширенной схемы И

* Для обозначения псевдоэлементов ТОЧКА часто пользуются также термином DOT, заимствованным из иностранной литературы (по-английски dot — точка).

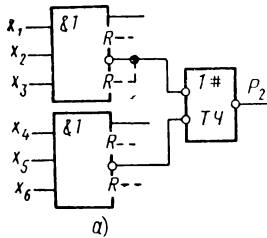


Рис. 4.10. Схема объединения инверсных выходов элементов:

а — изображение с помощью псевдоэлемента;
 б — условное изображение полученной схемы
 И — ИЛИ — НЕ

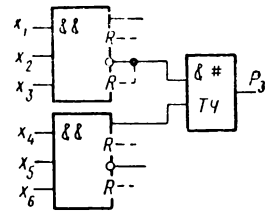
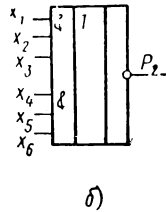


Рис. 4.11. Схема объединения прямого и инверсного выходов элементов

такого соединения с использованием псевдоэлемента показано на рис. 4.9, б. В данном случае реализуется конъюнкция («монтажное И»), поэтому в прямоугольнике псевдоэлемента, кроме сокращения ТЧ, реализуемая функция обозначена совокупностью символов &#. Элемент, соединяемый с псевдоэлементом, также обозначается двойным символом. Второй символ указывает тип псевдоэлемента, подключенного к данному выходу. С помощью непосредственного соединения может быть реализована и дизъюнкция («монтажное ИЛИ»). Это имеет место, например, при объединении инверсных выходов элементов, как показано на рис. 4.10, а. В данном случае $P_2 = x_1 x_2 x_3 \vee x_4 x_5 x_6$. Изображенный на рис. 4.10, а псевдоэлемент обозначен совокупностью символов 1#, а объединяемые — & 1.

При объединении инверсного и прямого выходов (рис. 4.11) реализуется функция $P_3 = x_1 x_2 x_3 x_4 x_5 x_6$. Поэтому, если на схеме, показанной на рис. 4.8, б, объединить выходы П и 1 (клеммы 2, 12 с 3 или 13 на рис. 4.8, а), то получим выход прямого значения суммы по mod 2, т. е. $xy (x \vee y) = xy \vee xy$.

Для логических схем И, И—ИЛИ—НЕ и др., получаемых в результате объединения входов или выходов элементов, применяют обобщающие условные обозначения, аналогичные показанным, например, на рис. 4.6, б, 4.9, в и 4.10, б. Такие обозначения, соответствующие более сложным схемам, по сравнению с простейшими (базовыми) элементами, содержащимися в корпусах интегральных схем, называют функциональными элементами (ФЭ). Они используются как графические единицы для построения функциональных схем.

4.3. ПОЛУПРОВОДНИКОВЫЕ ЗАПОМИНАЮЩИЕ ЭЛЕМЕНТЫ

В цепях передачи и преобразования информации процессоров, каналов и т. п. для запоминания и временного хранения информации используются триггеры. Они либо строятся из базовых и расширяемых логических элементов применяемого комплекса ин-

тегральных схем, как это имеет место, например, в ТТЛ-схемах, либо в составе комплекса предусматриваются специальные триггерные микросхемы, выполняемые в виде отдельных модулей. Такие модули имеются, например, в комплексах интегральных схем типа ЭСЛ.

В моделях ЭВМ Единой системы применяются симметричные потенциальные триггеры. Пример принципиальной схемы такого триггера, выполненной на базовых элементах типа ТТЛ, приведен на рис. 4.12. В этой схеме выходы инверторов двух логических элементов перекрестными обратными связями соединены с эмиттерами двухэмиттерных транзисторов, на которых выполнены входные схемы совпадения. Для нормальной работы такой схемы, при отсутствии запускающих сигналов, на входах установки в нулевое $У «0»$ и единичное $У «1»$ состояния должны присутствовать сигналы высокого уровня (u_0 и u_1). В этом случае сигнал высокого уровня с выхода инвертора одного элемента И—НЕ будет удерживать в открытом состоянии входную схему совпадения другого. В то же время сигнал низкого уровня с выхода инвертора последнего будет держать в закрытом состоянии входную схему совпадения первого.

Переключения триггера производится сигналами низкого уровня \bar{u}_1 и \bar{u}_0 , подаваемыми на входы $У «1»$ и $У «0»$. При этом одновременная подача сигналов низкого уровня на оба входа запрещена, т. е. $\bar{u}_1\bar{u}_0 = 0$. Во время переключения перекрестные обратные связи являются положительными, они способствуют развитию лавинообразного процесса. Тем не менее, для надежного переключения длительность входного запускающего сигнала должна превосходить время переходных процессов в триггере.

Запоминающие элементы, строящиеся на таких триггерах, различаются в основном типами входных цепей. Применяются схемы с однофазными и двухфазными входами. И те, и другие выполняются синхронизируемыми.

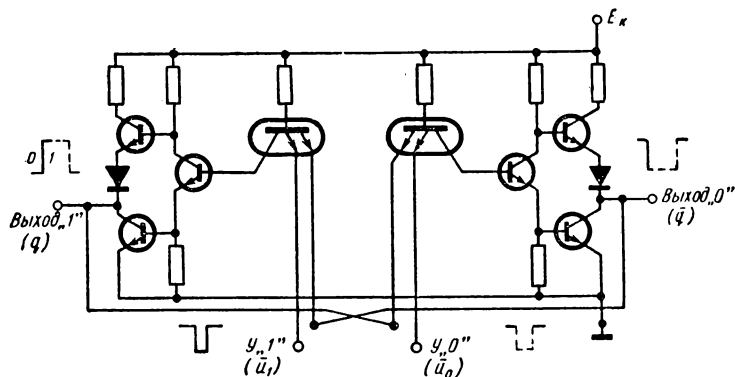


Рис. 4.12. Принципиальная схема триггера с кодowymi входами, выполненного на базовых ТТЛ-элементах

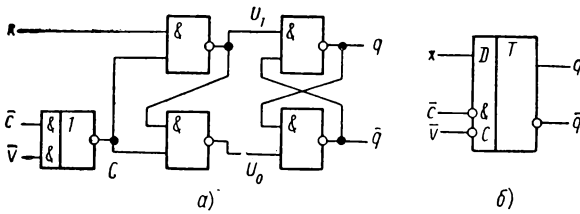


Рис. 4.13 Синхронизируемый триггер с однофазным входом:
 а — логическая схема;
 б — условное изображение

Пример логической схемы запоминающего элемента с однофазным входом приведен на рис. 4.13, а. Для того чтобы не усложнять терминологию, триггерные запоминающие элементы далее будем называть просто триггерами. Рассматриваемая схема является элементарным автоматом, функционирующим в соответствии с таблицей переходов (табл. 4.2).

Этот автомат является вполне определенным. Он обладает полной системой переходов, так как для перевода в каждое из устойчивых состояний имеется соответствующий входной сигнал. Согласно табл. 4.2, каноническое уравнение переходов триггера с синхронизируемым однофазным входом имеет вид

$$q_{t+1} = xC \vee q_t \bar{C}, \quad (4.1)$$

где q_t и q_{t+1} — состояния триггера в дискретные моменты времени t и $t + 1$;

x — входной кодовый сигнал;
 C — синхронизирующий сигнал.

Сигнал $C = \bar{c} \vee \bar{v}$, где c — сигнал синхронизирующей серии, а v — сигнал разрешения воздействия синхроимпульсов на триггер.

С учетом входной схемы И—ИЛИ—НЕ для формирования конъюнкции триггер с однофазным кодовым входом D может быть условно изображен, как показано на рис. 4.13, б. Такое изображение предназначается для функциональных схем.

Пример другой схемы синхронизируемого триггера, имеющего отдельные (парафазные) кодовые входы, показан на рис. 4.14, а. Такого типа триггеры строятся на ТТЛ-элементах на основе схем

Таблица 4.2

Таблица переходов

x	C	q _t	
		0	1
0	0	0	1
0	1	0	0
1	0	0	1
1	1	1	1

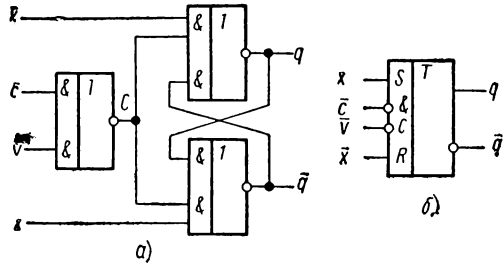
Таблица 4.3

Таблица переходов

x	\bar{x}	C	q _t	
			0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1

Рис. 4.14. Синхронизируемый триггер с парафазными входами:

a — логическая схема; *б* — условное изображение



И—ИЛИ—НЕ. Схема (рис. 4.14, *a*) работает согласно таблице переходов (табл. 4.3), в которой также $C = \overline{c \vee v}$. Этой таблице также соответствует каноническое уравнение вида (4.1). Условно данный триггер с парафазными входами может быть изображен, как показано на рис. 4.14, *б*. Здесь входы установки в единичное и нулевое состояния обозначены соответственно через *S* и *R*.

На токовых переключателях триггер с отдельными кодовыми входами строится по схеме, показанной на рис. 4.15. В этой схеме каждый из двух трехходовых логических элементов **И—НЕ** имеет два эмиттерных повторителя, один из которых является выходным, а другой предназначен для осуществления связи со входом другого элемента. Для организации перекрестных обратных связей в каждом из элементов используется по одному входному транзистору. Два других могут использоваться для приема входных сигналов. При отсутствии переключающих сигналов *У* «0» и *У* «1» на входах 8, 10 и 11, 12 должны быть низкие уровни потенциала. Переключается триггер входными сигналами высокого уровня. Для однозначности работы схемы сигналы высокого уровня не должны подаваться на входы противоположных плеч триггера одновременно.

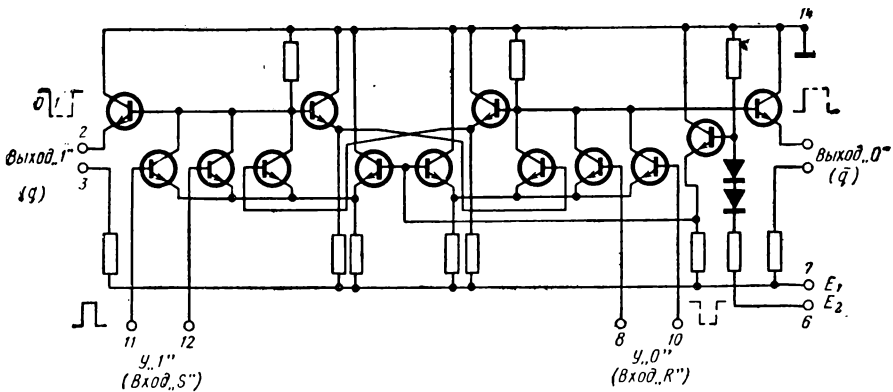


Рис. 4.15. Принципиальная схема триггера с кодовыми входами, выполненного на ЭСЛ-элементах

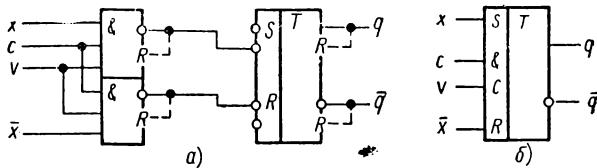


Рис. 4.16. Синхронизируемый триггер с парафазными входами:

a — логическая схема; *б* — условное изображение

В комплексе ИС серии 187 имеется интегральная схема такого триггера. Она служит основой для построения запоминающих элементов — синхронизируемых триггеров с парафазными и однофазными входами.

Схема синхронизируемого триггера с парафазными входами приведена на рис. 4.16, *a*. Собственно триггер *T* с кодовыми входами здесь соответствует схеме, показанной на рис. 4.15. Два входных элемента *И—НЕ*, также выполняемых на токовых переключателях, служат для организации синхронного переключения триггера в нулевое и единичное состояния. Формально логика работы всей схемы соответствует табл. 4.3. Здесь также через *c* обозначены синхронизирующие сигналы, а через *v* — сигналы управления. В отличие от схемы на элементах типа ТТЛ, в данной схеме коду 1 ставятся в соответствие сигналы низкого уровня, а коду 0 — высокого. Только при совпадении сигналов низкого уровня на всех трех входах верхней или нижней входной схемы совпадения на соответствующем входе собственно триггера появляется переключающий сигнал высокого уровня. Условное изображение *RS*-триггера, применяемое на функциональных схемах, приведено на рис. 4.16, *б*.

На рис. 4.17, *a* приведена схема с однофазным синхронным запуском. Здесь также сигналам кода 1 на входах *x*, *c*, *v* и выходе *q* ставятся в соответствие сигналы низкого уровня. Логика работы такой схемы соответствует табл. 4.2. На рис. 4.17, *б* буквой *D* обозначен кодовый вход, а буквой *C* — вход синхронизации, управляемый разрешающим сигналом *v* ($C = cv$). В комплексе серии 187 имеются специальные интегральные схемы, содержащие полностью выполненные синхронные триггеры с однофазными входами, включая входные цепи.

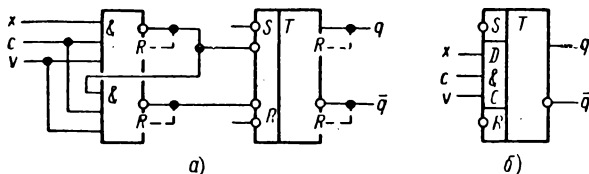


Рис. 4.17. Синхронизируемый триггер с однофазным входом:

a — логическая схема; *б* — условное изображение

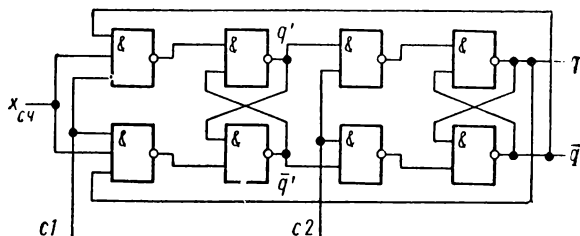


Рис. 4.18. Синхронизируемый двухступенчатый триггер со счетным входом

На интегральных схемах рассмотренных выше типов триггеры со счетными входами строятся двухступенчатыми. В каждой ступени используется синхронизируемый триггер с парафазными входами, например, как показано на рис. 4.18. Выходы q и \bar{q} правого триггера (2-я ступень) соединены перекрестными связями с входами входных элементов И—НЕ левого (1-я ступень). На другие входы этих входных элементов подаются информационные считаемые сигналы $x_{сч}$ и синхронизирующие сигналы $c1$. Сигналы с выходов q' и \bar{q}' триггера 1-й ступени вместе с синхронизирующими сигналами $c2$, сдвинутыми на полпериода относительно $c1$, подаются на входы входных вентилях правого триггера.

С одновременным приходом сигналов $x_{сч}$ и $c1$ (рис. 4.19) вначале в очередное состояние переключается триггер 1-й ступени. Состояние, в которое он должен переключиться, указывается сигналами с выходов q и \bar{q} триггера 2-й ступени. Последний выполняет роль элемента, запоминающего предыдущее состояние всей схемы на время перехода в новое состояние 1-й ступени. Предыдущее состояние сохраняется, пока отсутствует сигнал $c2$. Через перекрестные обратные связи удерживается в открытом состоянии только тот входной элемент И—НЕ, через который триггер 1-й ступени устанавливается в очередное состояние. Триггер 2-й ступени принимает это же состояние с приходом очередного сигнала $c2$. Наличие в рассмотренном триггере дополнительной запоминающей ступени обеспечивает необходимую задержку $t_{зq}$ (рис. 4.19) переключения выходных сигналов, управляющих входами 1-й ступени, по отношению к входным сигналам. Благодаря этому создаются условия для надежного пропуска входных сигналов через входные схемы совпадения триггера 1-й ступени и его переключения. По длительности входные сигналы должны превосходить время переходных процессов в этом триггере.

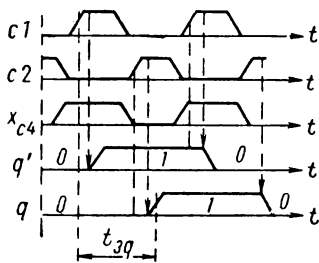


Рис. 4.19. Временная диаграмма работы триггера со счетным входом

4.4. ФУНКЦИОНАЛЬНЫЕ ЭЛЕМЕНТЫ

Большой объем и сложность электронного оборудования моделей ЭВМ Единой системы, а следовательно, и трудности в создании соответствующей ему графической документации обусловили необходимость в разработке специальной системы условных обозначений и соответствующей ей терминологии. В документации, отражающей принцип построения и процессы в реальном оборудовании, применяются понятия базового и функционального элементов.

Базовые элементы соответствуют определенным простейшим логическим функциям, согласно которым они осуществляют преобразование информации, и имеют конкретную реализацию в модулях интегральных схем. Условные обозначения базовых элементов применяются в качестве графических единиц для построения принципиальных схем. На основе таких обозначений строятся, например, принципиальные схемы всех ТЭЗ.

Функциональные элементы строятся на основе базовых и имеют свои условные обозначения. Однако последние несут большее функциональное содержание, соответствующее более сложным логическим функциям или их системам. Условные обозначения ФЭ в качестве графических единиц предназначаются для функциональных схем, которые также отражают процессы, происходящие в реальном оборудовании. Поэтому, являясь укрупненными по функциональному содержанию единицами, они в то же время имеют однозначную реализацию в базовых элементах. Это обеспечивается применением системы стандартных условных обозначений реализуемых функций и сокращенных наименований — идентификаторов, изображаемых составными буквенно-цифровыми символами. Идентификаторы указывают, как функциональные элементы реализуются с помощью базовых. При построении функциональных схем наряду с условными обозначениями функциональных элементов допускается применение базовых.

Если базовые элементы, применяемые в моделях ЕС-1020—ЕС-1050, реализуются в пределах модулей интегральных схем, то функциональные элементы, как правило, не ограничиваются этими пределами. Для их построения может потребоваться несколько десятков модулей ИС, как это имеет место, например, в модели ЕС-1050. В зависимости от требований к быстродействию схем могут налагаться условия на компоновку модулей, составляющих ФЭ. Например, в модели ЕС-1050 любой функциональный элемент должен целиком размещаться на плате одного ТЭЗ.

Функциональные элементы делятся на классы, каждый из которых охватывает группу схем, реализующих определенного типа логические соотношения. В каждой группе ФЭ различаются допустимыми количествами входов и выходов. Для элементов на токовых переключателях имеется 12 классов, соответственно содержащих:

- схемы *И*, *И—НЕ*;
- схемы *ИЛИ*, *ИЛИ—НЕ*;

- схемы И—ИЛИ, И—ИЛИ—НЕ;
- схемы, осуществляющие поразрядное сравнение;
- схемы, осуществляющие поразрядное сравнение и имеющие на выходе триггеры;
- схемы свертки (сложения кодов по mod 2);
- схемы свертки с триггерными выходами;
- схему полусумматора с триггерными выходами;
- схемы триггеров с однофазными и парафазными входами различной структуры;
- схемы усилителей, выполненных на типовых ИС комплекса, имеющих повышенную нагрузочную способность;
- схемы дешифраторов;
- схемы дешифраторов с триггерными выходами.

Примеры построения схем ФЭ типов И, И—НЕ; ИЛИ, ИЛИ—НЕ и И—ИЛИ, И—ИЛИ—НЕ были рассмотрены соответственно на рис. 4.6, 4.9, 4.11, 4.10. Согласно схеме рис. 4.9, б могут строиться схемы совпадения на 3—25 входов, согласно рис. 4.10, а — схемы И—ИЛИ—НЕ, содержащие 2—10, 2—15 или 4—25 входных схем совпадения и т. д.

Схемы поразрядного сравнения строятся таким образом, что на прямом выходе сигнал кода 1 вырабатывается в том случае, если сравниваемые коды поразрядно совпадают. Пример такой схемы для сравнения двухразрядных кодов, на выходе которой включен триггер, дан на рис. 4.20. При представлении сигналов кода 1 низким уровнем потенциала на выходе Р псевдоэлемента сигналы формируются согласно логическому соотношению

$$P = (a_1 b_1 \vee \bar{a}_1 \bar{b}_1) (a_2 b_2 \vee \bar{a}_2 \bar{b}_2). \quad (4.2)$$

Схема запоминания этих сигналов такая же, как была рассмотрена на рис. 4.17. Как показано на рис. 4.20, б, триггер может иметь дополнительные цепи установки его (штриховые линии) в нулевое и единичное состояния по входам R и S. С помощью модификаций функциональных элементов рассматриваемого типа можно производить сравнения 3—7- и 8—12-разрядных кодов.

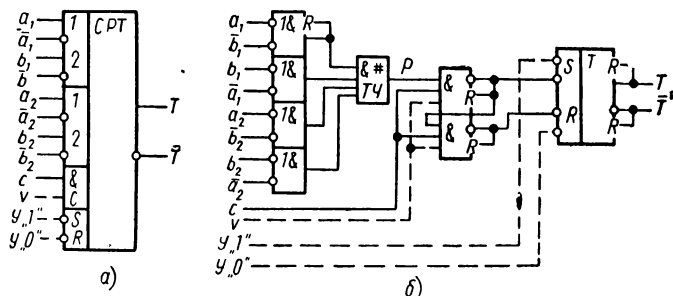


Рис. 4.20. Схема поразрядного сравнения с триггерным выходом:

а — условное изображение ФЭ; б — пример построения ФЭ на базовых элементах

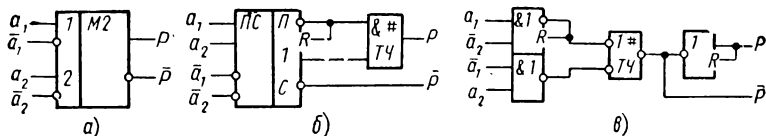


Рис. 4.21. Схема сложения по mod 2:

a — условное изображение ФЭ; *б* — построение на основе модуля полусумматора; *в* — построение на универсальных логических элементах

Простейшие примеры двух вариантов сумматора по mod 2 приведены на рис. 4.21. Схема, соответствующая условному обозначению, показанному на рис. 4.21, *a*, может быть выполнена на основе модуля полусумматора. Как было отмечено в п. 4.2, прямой выход суммы по mod 2 может быть получен соединением друг с другом в полусумматоре выходов *П* и *1*. В результате получается

$$P = (a_1 \vee a_2) \overline{a_1 a_2} = a_1 \bar{a}_2 \vee \bar{a}_1 a_2. \quad (4.3)$$

Во втором случае (рис. 4.21, *в*) имеет место соотношение

$$P = \overline{a_1 a_2 \vee \bar{a}_1 \bar{a}_2} = a_1 \bar{a}_2 \vee \bar{a}_1 a_2. \quad (4.4)$$

В наборе функциональных элементов имеются схемы свертки, позволяющие суммировать по mod 2 также 4—5- и 8—9-разрядные двоичные коды.

Класс триггеров содержит обширный перечень элементов памяти с однофазными и парафазными синхронизируемыми цепями ввода информации. Имеются схемы ФЭ, содержащие до четырех параллельно работающих триггеров. Такие элементы используются в сильно нагруженных схемах памяти. Для построения запоминающих схем, способных принимать информацию на хранение от большого количества различных источников, предусмотрены многоходовые триггерные схемы. Пример триггера с двумя независимыми однофазными синхронизируемыми входами и с двумя параллельно работающими RS-триггерами, аналогичными показанному на рис. 4.15, приведен на рис. 4.22. Здесь каждый из однофазных входов выполнен по схеме, показанной на рис. 4.17, *a*. На входах входных схем *И*—*НЕ* сигналам

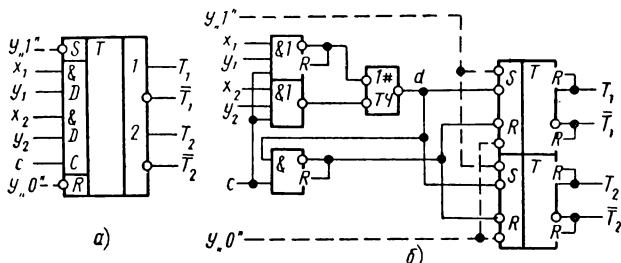


Рис. 4.22. Вариант схемы триггера с однофазным запуском: *a* — условное изображение ФЭ; *б* — пример построения на базовых элементах

кода 1 соответствуют низкие уровни потенциала. На входах R и S триггеров переключающими являются высокие уровни потенциала. В точке d они формируются согласно логическому соотношению

$$d = \overline{(x_1 y_1 \vee x_2 y_2)} c. \quad (4.5)$$

В схеме дополнительно могут быть использованы входы установки триггеров в единичное и нулевое состояния сигналами \mathcal{U} «1» и \mathcal{U} «0» (штриховые линии).

Среди ФЭ имеются триггеры, способные иметь до 15—25 входных схем совпадения, подобных изображенным на рис. 4.22, б.

Дешифраторы, рассчитанные на дешифрирование 2-, 3-, 4- и 5-разрядных двоичных кодов, также относятся к функциональным элементам. Пример схемы дешифратора на три разряда приведен на рис. 4.23. В этой схеме выходные сигналы связаны с входными следующими логическими соотношениями:

$$\begin{aligned} v_0 &= \overline{x_1 x_2 x_3}; & \bar{v}_0 &= \overline{\overline{\overline{x_1 x_2 x_3}}}; \\ v_1 &= \overline{x_1 x_2} x_3; & \bar{v}_1 &= \overline{\overline{\overline{x_1 x_2} x_3}}; \\ v_2 &= \overline{x_1} x_2 \overline{x_3}; & \bar{v}_2 &= \overline{\overline{\overline{x_1} x_2 \overline{x_3}}}; \\ v_3 &= \overline{x_1} x_2 x_3; & \bar{v}_3 &= \overline{\overline{\overline{x_1} x_2 x_3}}; \\ v_4 &= x_1 \overline{x_2} \overline{x_3}; & \bar{v}_4 &= \overline{\overline{\overline{x_1 \overline{x_2} \overline{x_3}}}}; \\ v_5 &= x_1 \overline{x_2} x_3; & \bar{v}_5 &= \overline{\overline{\overline{x_1 \overline{x_2} x_3}}}; \\ v_6 &= x_1 x_2 \overline{x_3}; & \bar{v}_6 &= \overline{\overline{\overline{x_1 x_2 \overline{x_3}}}}; \\ v_7 &= x_1 x_2 x_3; & \bar{v}_7 &= \overline{\overline{\overline{x_1 x_2 x_3}}}. \end{aligned} \quad (4.6)$$

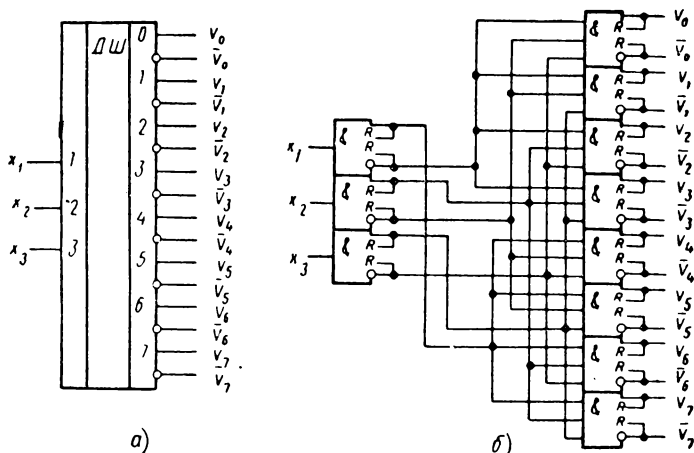


Рис. 4.23. Дешифратор:

a — условное изображение ФЭ; b — пример построения на базовых элементах

Сигнал кода 1 появляется только на том выходе, который соответствует 3-разрядному коду, поступившему на вход. Как и в предыдущих схемах, коду 1 на входах и прямых выходах ставится в соответствие низкий уровень потенциала.

Среди функциональных элементов имеются две разновидности дешифраторов: не содержащие элементов памяти и содержащие на выходах триггеры типа *RS*.

4.5. ТИПОВЫЕ ФУНКЦИОНАЛЬНЫЕ СХЕМЫ

Для более наглядного изображения процессов, протекающих в определенных частях устройств или в устройствах в целом, применяют так называемые структурные схемы. На этих схемах условными взаимосвязанными графическими единицами показываются группы функциональных элементов, имеющие определенное целевое назначение. Чаще всего эти группы предназначаются для выполнения простейших операций (микроопераций) над машинными словами или их частями: посылка, запоминание, сдвиг кода, сложение кодов и т. п. Такие группы элементов, часто называемые узлами или микрооперационными устройствами, принято подразделять на типовые и нетиповые. Типовые узлы в общем случае предназначаются для логического преобразования, временного запоминания или восстановления по форме и усиления по мощности стандартных сигналов, соответствующих машинным словам или их частям, содержащим несколько двоичных цифр. К таким узлам, наиболее часто используемым в устройствах, относятся:

- регистры для оперативного запоминания и временного хранения информации;
- сдвигающие регистры;
- счетчики;
- сумматоры;
- дешифраторы;
- схемы сравнения;
- схемы свертки.

Нетиповыми узлами обычно являются группы специальных усилителей, видоизменяющих сигналы, которыми представляются переменные, но в то же время не меняющих их информационного содержания. Такие узлы находят применение в запоминающих устройствах и устройствах ввода-вывода. Это — генераторы и формирователи токов и напряжений специальной формы, усилители слабых сигналов и т. п.

На структурных схемах узлы изображают в виде квадратов, прямоугольников, трапеций и других плоских фигур установленной стандартами формы. Например, в общем случае любая функциональная часть может быть изображена прямоугольником, а комбинационная схема, не содержащая элементов памяти, — трапецией, как показано соответственно на рис. 4.24, *а* и *б*. Внутри

указанных фигур даются полные или сокращенные условные названия узлов.

Типовые узлы строятся из логических и запоминающих элементов применяемых комплексов ИС. Особенности использования последних зависят от типа реализуемой схемы, от требований, предъявляемых к ее быстродействию и объему оборудования, от возможностей самих элементов комплекса, определяемых их техническими и электрическими параметрами. Рассмотрим примеры построения узлов на потенциальных элементах типа ЭСЛ.

Регистры для оперативного запоминания и временного хранения информации. Такие регистры строятся на триггерах как с однофазными входами, так и с парафазными. В зависимости от количества источников, от которых регистр должен принимать информацию, на входах триггеров включают соответствующие группы вентиляей.

Например, как показано на рис. 4.25, *а*, 8-разрядный регистр может принимать 8-разрядные коды по двум направлениям. Приемом кодов [X] управляют сигналы $v1$, а кодов [Y] — $v2$. Работа триггеров регистра синхронизируется сигналами $c1$ и $v3$.

Если не предусмотрена специальная цепь установки триггеров в нулевое состояние, то регистр на триггерах с однофазными входами может быть выполнен, как показано на рис. 4.25, *б*. На рисунке изображена функциональная схема нулевого разряда регистра. На входе включены две схемы совпадения, выходы которых объединены по схеме ИЛИ (см. рис. 4.10). Через входные схемы совпадения по сигналу $v1$ или $v2$ на вход триггера проходит соответственно переменная x_0 или y_0 . Входным и выходным сигналам кода 1 ставятся в соответствие низкие уровни потенциала. При этом условии работу схемы можно описать уравнением

$$q_{i+1}^0 = q_i^0 \bar{c} \bar{v} \bar{3} \vee (x_0 v1 \vee y_0 v2) c1 v3. \quad (4.7)$$

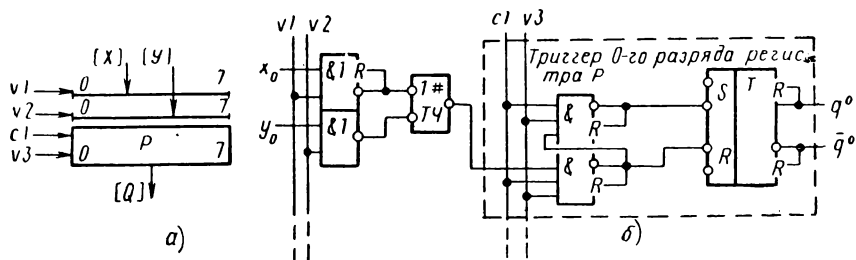


Рис. 4.25. Восьмиразрядный регистр для оперативного запоминания и временного хранения информации:

а — условное структурное изображение; *б* — функциональная схема 0-го разряда

Триггер нулевого разряда регистра выполнен по схеме, показанной на рис. 4.17, *а*. Такой триггер обеспечивает хранение информации минимум в течение периода следования синхроимпульсов *с1*. Продолжительность хранения зависит от поступления управляющего сигнала *υ3*.

Сдвигающие регистры, кроме запоминания кодов, осуществляют их сдвиги в разрядной сетке вправо или влево на необходимое количество разрядов. Для того чтобы сдвинуть число в регистре, в общем случае необходимо произвести следующие действия:

«прочитать» содержимое каждого триггера регистра и послать соответствующий считанному коду сигнал в определенный триггер справа или слева, в зависимости от направления сдвига и от количества разрядов, на которое происходит сдвиг;

подготовить триггеры к приему посланной информации;

принять в триггеры сдвинутый код.

При использовании одноступенчатых триггеров выдача информации с триггера и прием информации в триггер производятся синхросигналами разных серий, не совпадающими по времени. Поэтому для правильного выполнения перечисленных выше действий снимаемую с триггеров информацию надо где-то запоминать на время, на которое отстоят друг от друга эти синхросигналы. Такое временное запоминание в потенциальных схемах осуществляют вспомогательные триггеры, включаемые в цепи сдвига. Они образуют дополнительный регистр. В результате в целом сдвигающий регистр получается двухрядным, например, как показано на рис. 4.26, *а*.

Исходный код $[X]$ принимается в регистр $P1$ по сигналу ПК (прямой код). В триггеры регистра он заносится по синхросигналу *с1* при наличии разрешающего потенциала *υ2*. Для осуществления сдвига принятый код по синхросигналу *с2* засылается для промежуточного хранения в регистр $P2$. Затем из этого регистра он возвращается с необходимым сдвигом вправо или влево в регистр $P1$.

Собственно сдвиг производится комбинационной схемой, включенной между регистрами $P2$ и $P1$. В рассматриваемом сдвигающем регистре возможны однократные сдвиги влево и вправо на один, два, четыре и восемь разрядов ($L1, L2, L4, L8, P1, P2, P4, P8$). Способ построения комбинационной схемы, включенной на входе $P1$, и цепи подачи управляющих сигналов сдвига, опущенные на схеме рис. 4.26, *а*, поясняются функциональной схемой рис. 4.26, *б*.

На входе триггера *i*-го разряда в общем случае включается схема *И—ИЛИ*, имеющая девять двухвходовых схем совпадения. Первая сверху предназначена для ввода значения *x*, исходного операнда $[X]$, остальные — для передачи содержимого триггеров регистра $P2$ с соответствующим сдвигом. Сигналам кода 1 в схеме соответствует низкий уровень потенциала. Построение схемы

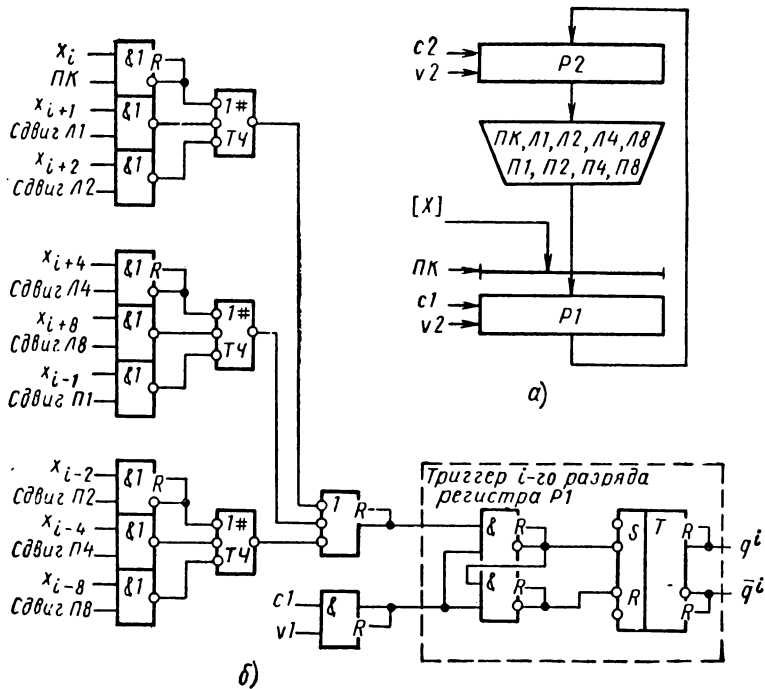


Рис. 4.26. Схема сдвига:

a — условное структурное изображение; *б* — функциональная схема *i*-го разряда регистра *P1* с входным коммутатором

И—ИЛИ аналогично рассмотренному на рис. 4.25, *б*. Элемент памяти регистра также выполнен одноходовым.

Счетчики. При использовании потенциальных элементов такие узлы, предназначенные для выполнения операций счета сигналов кода 1, последовательно подаваемых на их входы, строятся на двухступенчатых триггерах. Счетчики бывают суммирующие, вычитающие и реверсивные. Последние в зависимости от характера поступивших переключающих сигналов могут вести счет как в сторону увеличения суммы (прибавления), так и в сторону уменьшения (вычитания). В счетчиках на потенциальных триггерах обычно применяют либо цепи сквозного переноса, либо параллельного.

На рис. 4.27 приведен пример функциональной схемы двух младших разрядов *n*-разрядного счетчика. В данном случае младшим является двухступенчатый триггер *n*-го разряда. Двухступенчатые триггеры выполнены на одноходовых триггерах, однако в целом двухступенчатые схемы работают аналогично рассмотренному на рис. 4.18.

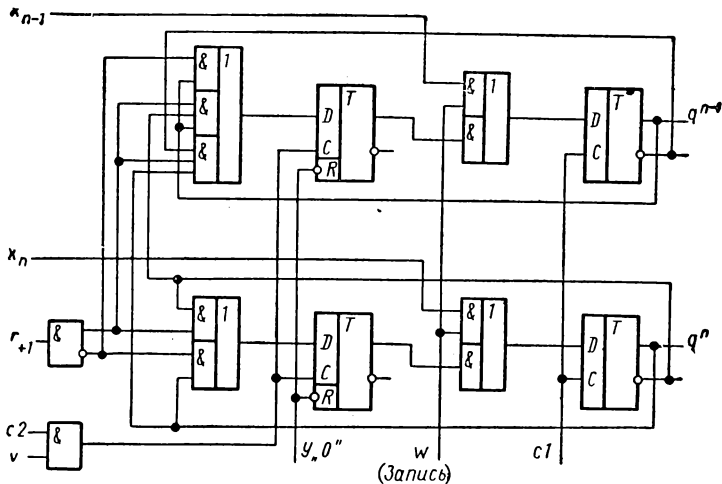


Рис. 4.27. Функциональная схема двух разрядов счетчика

Входными сигналами счетчика являются потенциалы низкого уровня. Они обозначены символом r_{+1} . Длительность их должна быть не меньше времени полного срабатывания триггеров первой ступени (на схеме — левые триггеры). Работа ступеней синхронизируется сигналами $c1$ и $c2$, сдвинутыми друг относительно друга на полпериода.

Цепь переноса между разрядами выполнена по параллельной схеме. Если триггер 1-й ступени n -го разряда работает согласно уравнению

$$q_{i+1}^n = [q_i^n \bar{r}_{+1} \vee \bar{q}_i^n r_{+1}] C, \quad (4.8)$$

то переключение триггера 1-й ступени $(n - 1)$ -го разряда происходит в соответствии с соотношением

$$q_{i+1}^{n-1} = [q_i^{n-1} \bar{r}_{+1} \vee q_i^{n-1} \bar{q}_i^n r_{+1} \vee \bar{q}_i^{n-1} q_i^n r_{+1}] C, \quad (4.9)$$

где в обоих случаях $C = c2v$. При последовательной подаче на вход сигналов r_{+1} состояния $q^{n-1}q^n$ последовательно принимают, например, значения 00, 01, 10, 11, 00, ... и т. д.

По управляющим сигналам w в счетчик могут заноситься исходные коды $x_{n-1}x_n$. Общая установка триггеров счетчика

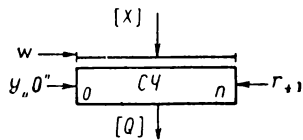


Рис. 4.28. Условное изображение счетчика

в нулевое состояние производится сигналами положительной полярности $У «0»$, подаваемыми непосредственно на входы R . В целом n -разрядный счетчик на структурной схеме может быть изображен, как показано на рис. 4.28. На условном изображении не показаны цепи сигналов синхронизации.

Кроме регистров и счетчиков большую роль в схемах ЭВМ Единой системы играют также другие узлы, перечисленные в начале параграфа. В моделях ЕС-1030 и ЕС-1050 сумматоры, например, составляют основу арифметико-логических устройств. Принципы построения и работа этих сумматоров рассматриваются в гл. 6.

Дешифраторы, схемы сравнения и свертки, во множестве используемые в различных устройствах машин, строятся, например, на элементах типа ЕСЛ так, как было показано в предыдущем параграфе. При побайтовом делении информации широко используются схемы, которые по количеству входов по сути дела относятся к функциональным элементам. Однако на структурных схемах во многих случаях их оказывается удобно выделять как самостоятельные узлы. Если требуются схемы, по количеству входов превышающие возможности функциональных элементов, перечисленных в каталогах, то они строятся из этих элементов многокаскадными, например по типу пирамидальных схем и т. п.

ОПЕРАТИВНАЯ И ПОСТОЯННАЯ ПАМЯТЬ

5.1. ПРИНЦИПЫ ПОСТРОЕНИЯ ОПЕРАТИВНОЙ ПАМЯТИ ТИПА 2,5D

В машинах Единой системы, как и вообще в современных ЭВМ, оперативная память, как правило, строится на ферритовых сердечниках с прямоугольной петлей гистерезиса. Она получила название МОЗУ — магнитное оперативное запоминающее устройство. Долголетие МОЗУ объясняется такими ее положительными качествами, как высокое быстродействие, большая емкость, высокая надежность работы, возможность хранения информации неограниченно долгое время без затраты энергии.

Известно, что в зависимости от организации системы выборки необходимой ячейки памяти различают следующие типы адресных МОЗУ:

а) трехмерные (трехкоординатные) МОЗУ, или МОЗУ типа 3D (от англ. «three dimensional» — три измерения, три координаты). В отечественной литературе они называются также МОЗУ матричного типа, МОЗУ с записью и считыванием по принципу совпадения полutoков;

б) двухмерные (двухкоординатные) МОЗУ, или МОЗУ типа 2D, иначе называемые МОЗУ с линейной выборкой, МОЗУ типа z . В таких устройствах запись производится по принципу совпадения полutoков, а считывание осуществляется полным током;

в) 2,5-мерные МОЗУ, или МОЗУ типа 2,5D, занимающие по своим свойствам и организации системы выборки промежуточное положение между двухмерными и трехмерными МОЗУ.

Все эти типы МОЗУ относятся к статическим ЗУ с адресным обращением и с произвольным доступом к ячейкам памяти.

В ЕС ЭВМ основная оперативная память строится по типу 2,5-мерного МОЗУ [36]. Рассмотрим сначала общие принципы построения МОЗУ типа 2,5D. Это удобнее сделать на конкретном примере построения МОЗУ небольшой емкости [15].

Пусть необходимо построить схему МОЗУ типа 2,5D для хранения 32 трехрядных чисел ($E = 32$, $n = 3$). В трехмерном МОЗУ такой емкости адресные шины по каждой координате проходили бы через 3 сердечника, а разрядные шины и шины запрета — через 32 сердечника. В двухмерном МОЗУ каждая

числовая шина (шина z) проходила бы через 3 сердечника, а каждая разрядная шина записи и разрядная шина считывания — через 32 сердечника (без учета стабилизирующих сердечников). Для более равномерного распределения сердечников между шинами в 2,5-мерном МОЗУ по сравнению с двухмерным уменьшено в k раз количество числовых (адресных) шин и во столько же раз увеличено количество разрядных шин. Величина k определяется из соотношения $k = \sqrt{\frac{E}{n}}$ и затем округляется до значения 2^p (p — целое число), что необходимо для удобства дешифрации.

В нашем примере $k = \sqrt{\frac{32}{3}} \approx 3,3$ или после округления $k = 4$. Теперь количество адресных шин будет $32 : 4 = 8$, а количество разрядных шин — $3 \times 4 = 12$. Каждая адресная шина проходит через 12 запоминающих сердечников, а каждая разрядная шина — через 8 сердечников. Количество сердечников, прошитых каждой адресной шиной, будет точно соответствовать количеству сердечников, через которые проходит каждая разрядная шина, если имеет место равенство

$$\frac{F}{n} = 2^{2p}. \quad (5.1)$$

Например, если $E = 8192$, $n = 32$, то $p = 4$, и тогда каждая адресная шина и каждая разрядная шина проходят через 512 сердечников (здесь $k = 16$). Общее число разрядных и адресных шин равно $512 + 512 = 1024$, что существенно меньше по сравнению с общим числом таких шин в двухмерном МОЗУ ($8192 + 32 = 8224$) той же емкости.

В схеме 2,5-мерного МОЗУ емкостью 32 трехразрядных числа (рис. 5.1) можно выделить следующие части.

1. Прямоугольная матрица, составленная из $8 \times 12 = 96$ запоминающих сердечников (в двухмерном МОЗУ матрица имела бы вид 32×3). По горизонтали расположены адресные шины, а по вертикали — разрядные. Сердечники, служащие для хранения одноименных разрядов всех запоминаемых чисел, обведены штриховой линией. Поскольку $n = 3$, все сердечники матрицы объединены в три группы. Сердечники, принадлежащие одной ячейке памяти, связаны с одной из адресных шин.

2. Регистр адреса РА, составленный из четырех частей, обозначенных через $РА_{x_1}$, $РА_{x_2}$, $РА_{y_1}$, $РА_{y_2}$. Число разрядов регистра РА равно 5, так как $E = 32$. Регистры $РА_{x_2}$, $РА_{y_1}$, $РА_{y_2}$ являются одноразрядными, а регистр $РА_{x_1}$ — 2-разрядным.

3. Дешифраторы адреса $ДШ_{x_1}$, $ДШ_{x_2}$, $ДШ_{y_1}$, $ДШ_{y_2}$. В рассматриваемой схеме дешифраторы $ДШ_{x_2}$, $ДШ_{y_1}$, $ДШ_{y_2}$, связанные с одноразрядными регистрами, включены для общности, так как управление работой соответствующих ключей и формирователей может осуществляться непосредственно сигналами,

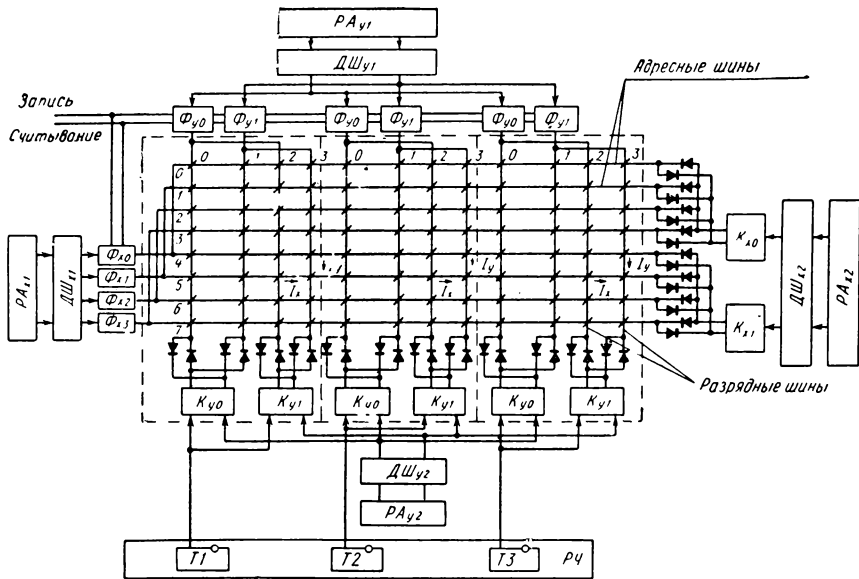


Рис. 5.1. МОЗУ типа 2,5D малой емкости

поступающими с регистров. В МОЗУ большой емкости эти дешифраторы обязательны. Например, для указанного выше примера, когда $E = 8192$ 32-разрядных числа, каждый из дешифраторов ДШ_{х2}, ДШ_{у1}, ДШ_{у2} строится на 8 выходов.

4. Ключи и формирователи, связанные с дешифраторами адреса.

5. Диодные схемы, связывающие адресные и разрядные шины с соответствующими ключами и обеспечивающие прохождение токов I_x и I_y при записи и считывании информации как в прямом направлении, так и в обратном.

6. Регистр числа РЧ, в котором фиксируется либо считанное число, либо записываемое. Необходимо подчеркнуть, что в схеме МОЗУ типа 2,5D формирователи и ключи по координате x являются общими для всех разрядов запоминаемых чисел. Шины выборки по координате y (разрядные шины) каждого разряда обслуживают отдельные формирователи и ключи.

Рассмотрим, как производится запись и считывание информации из выделенной ячейки памяти, адрес которой 10111, т. е. $x = 101$, $y = 11$ и в регистрах РА_{х1}, РА_{х2}, РА_{у1}, РА_{у2} будут записаны двоичные числа соответственно 10; 1; 1; 1.

Адресные шины в каждом разряде проходят не через один сердечник, как это имеет место в двухмерном МОЗУ, если не считать стабилизирующие сердечники, а через четыре (в общем случае через K сердечников). В связи с этим в 2,5-мерном МОЗУ, так же

как и в трехмерном, принцип совпадения полуточков используется не только при записи, но и при считывании.

При считывании числа, записанного в выделенной ячейке памяти, в результате дешифрации адреса числа возбуждаются формирователи Φ_{x1} , Φ_{y1} и открываются ключи K_{x1} , K_{y1} . Через пятую адресную и третьи разрядные шины всех разрядов проходят импульсы токов I_x и I_y , амплитуда которых равна $0,5I_m$ (направление токов показано стрелками), где I_m — ток, обеспечивающий создание перемещающего поля. Если в избранном сердечнике хранился код 1, то под действием магнитного поля, создаваемого токами I_x и I_y , он перемещается в нулевое положение, и в выходной обмотке (на рис. 5.1 она не показана), которая проходит через все сердечники одного разряда, наводится э. д. с. считывания. Если в избранном сердечнике хранился код 0, его магнитное состояние не изменяется, и в идеальном случае в выходной обмотке э. д. с. считывания отсутствует. Для уменьшения помех от полувозбужденных (полуизбранных) сердечников, подвергающихся действию одного из токов I_x или I_y , выходная обмотка, как и в трехмерном МОЗУ, пронизывает все сердечники одного разряда матрицы в диагональном направлении. Считанное число подается в регистр числа, а оттуда — в другие устройства ЭВМ.

При записи или регенерации числа в выделенную ячейку памяти дешифрация адреса осуществляется так же, как и при считывании, однако направление токов I_x и I_y изменяется на противоположное. Следовательно, все формирователи и ключи должны быть биполярными. Если в данном разряде ячейки памяти необходимо записать (восстановить) код 1, соответствующий сердечник ячейки подвергается действию токов I_x и I_y . При записи или восстановлении кода 0 в i -м разряде ток I_y не посылается. Для этого сигналом с триггера i -го разряда регистра числа закрываются соответствующие ключи K_y , поэтому сердечник i -го разряда ячейки памяти, подвергающийся действию только тока I_x , остается в нулевом положении.

Надежная работа МОЗУ обеспечивается при выполнении следующих условий:

$$I_{x \min} + I_{y \min} \approx 2I_{x \min} \geq I_m \quad (5.2)$$

$$I_{x \max} \approx I_{y \max} \leq I_n \quad (5.3)$$

где I_n — максимальное (пороговое) значение тока, при котором еще не происходит перемещения сердечника по полному циклу.

Выполнение условия (5.2) обеспечивает надежное перемещение сердечника избранной ячейки памяти при записи или считывании кода 1, а выполнение условия (5.3) — надежное сохранение состояния полуизбранных сердечников или избранного сердечника при записи в него кода 0.

Обозначим допустимые относительные отклонения токов I_x и I_y от их номинальных значений через δ_x и δ_y , причем

$$\delta_x = \frac{I_{x \max} - I_x}{I_x} = \frac{I_x - I_{x \min}}{I_x}; \quad (5.4)$$

$$\delta_y = \frac{I_{y \max} - I_y}{I_y} = \frac{I_y - I_{y \min}}{I_y}. \quad (5.5)$$

Тогда для случая максимальных допусков соотношения (5.2) и (5.3) можно записать в виде

$$(1 - \delta_x)I_x + (1 - \delta_y)I_y = I_m \quad (5.6)$$

$$(1 + \delta_x)I_x = I_n; \quad (5.7)$$

$$(1 + \delta_y)I_y = I_n. \quad (5.8)$$

Совместное решение уравнений (5.6)—(5.8) с учетом обозначения $\alpha = I_m/I_n$ позволяет получить соотношение

$$\frac{1 - \delta_x}{1 + \delta_x} + \frac{1 - \delta_y}{1 + \delta_y} = \alpha, \quad (5.9)$$

которое определяет взаимную зависимость допусков δ_x и δ_y на токи I_x , I_y при различных значениях параметра α , характеризующего материал сердечников, используемых в МОЗУ.

Для типичного значения $\alpha = 1,6$ в 2,5-мерном МОЗУ относительные допуски на отклонение амплитуд токов I_x и I_y составляют $\delta_x = \delta_y = 11\%$. В двухмерном МОЗУ при тех же условиях $\delta_x = \delta_{\text{зп}} = 24\%$ и в трехмерном $\delta_x = \delta_{\text{зпр}} = 7,5\%$. Этими цифрами подтверждается, что 2,5-мерные МОЗУ занимают промежуточное положение по сравнению с двухмерными и трехмерными МОЗУ по критичности к колебаниям амплитуды импульсов возбуждения.

По сравнению с трехмерными и двухмерными МОЗУ 2,5-мерная память имеет следующие преимущества:

- меньшее количество шин, проходящих через запоминающие сердечники, по сравнению с трехмерными МОЗУ;

- более простые и экономичные схемы, обеспечивающие запись, считывание и регенерацию информации, по сравнению с двухмерными МОЗУ той же емкости;

- меньшее число сердечников, прошитых одной разрядной шиной, по сравнению с числом сердечников, прошитых разрядной шиной в МОЗУ типа 3D или разрядной шиной считывания (записи) в МОЗУ типа 2D;

- отсутствие разрядного тока запрета (необходимого в трехмерных МОЗУ при записи или восстановлении кода 0 в данном разряде записываемого числа) и соответственно проблемы успокоения выходных разрядных линий после подачи импульса тока запрета.

Основные недостатки 2,5-мерных МОЗУ:

— невозможность форсированного перематничивания сердечников;

— использование принципа совпадения полуточков в такте считывания и необходимость принятия мер для уменьшения помех от полувозбужденных (полуизбранных) сердечников и для снижения некомпенсируемых помех.

5.2. СОСТАВ И НАЗНАЧЕНИЕ ОСНОВНЫХ БЛОКОВ ОПЕРАТИВНОЙ ПАМЯТИ ТИПА 2,5D ДЛЯ МОДЕЛИ ЕС-1020

Структуру и организацию функционирования 2,5-мерной оперативной памяти большой емкости рассмотрим на примере ОЗУ машины ЕС-1020 (шифр этого устройства ЕС-3220). Устройство ЕС-3220 предназначено для приема, хранения и выдачи информации при работе совместно с процессором ЕС-2020 модели ЕС-1020 ЕС ЭВМ. Условия его эксплуатации следующие:

— диапазон изменения температуры окружающего воздуха $+5 \div +40^\circ \text{C}$,

— относительная влажность при $t = 30^\circ \text{C}$ до 95%,

— атмосферное давление от 6 до 10 кПа.

Оперативная память ЕС-3220 состоит из трех типов логически самостоятельных частей: основной — ОП ОЗУ, локальной — ЛП ОЗУ и мультиплексной — МП ОЗУ. В зависимости от модификации ЕС-3220 емкость каждой из этих частей соответствует данным таблицы 5.1.

Т а б л и ц а 5.1

Емкость оперативной памяти модели ЕС-1020

Модификация	Емкость памяти в байтах			Примечание
	ОП ОЗУ	ЛП ОЗУ	МП ОЗУ	
ЕС-3200-1	64К	256	768	К = 1024
ЕС-3220-2	128К	256	1792	
ЕС-3220-3	256К	256	1792	

В каждый данный момент обращение может осуществляться только к одному из указанных типов памяти в соответствии с поступившим из процессора признаком по одной из шин ОП, ЛП или МП. При обращении к памяти производится запись или считывание одновременно двух байт информации в параллельном коде.

Полный цикл обращения делится на два самостоятельных такта: такт считывания и такт записи, начало которых определяется сигналами СЧИТЫВАНИЕ и ЗАПИСЬ, поступающими из про-

цессора. По сигналу СЧИТЫВАНИЕ в соответствии с адресом ячейки в одном из трех типов памяти выдается двухбайтовое число на кодовые шины (КШЧ). При этом информация в выбранной ячейке памяти стирается и не восстанавливается. По сигналу ЗАПИСЬ осуществляется запись двух байт информации, поступившей по кодовым шинам записи (КШЗ), в ту ячейку, из которой перед этими производилось считывание.

В течение полного цикла обращения к памяти выполняется одна из следующих операций «Считывание — регенерация», «Стирание — запись», «Считывание — обработка — запись». При выполнении операции «Считывание — регенерация» в такте считывания из выбранной ячейки памяти осуществляется перепись числа в регистр числа (РЧ), а затем в такте записи — перепись числа из регистра РЧ в ту же ячейку памяти. Операция «Стирание — запись» необходима только для записи числа в определенную ячейку памяти, поэтому в такте считывания эта ячейка освобождается от информации, но считанное число в регистр РЧ не записывается (регистр блокируется). В такте записи в ОЗУ подается команда ЗАПИСЬ и код записываемой информации по КШЗ. Выполнение операции «Считывание — обработка — запись» отличается от операции «Считывание — регенерация» тем, что в такте записи из процессора вместе с командой ЗАПИСЬ по КШЗ поступает уже обработанная информация, отличающаяся от считанной в предыдущем такте.

Минимальное время цикла обращения к оперативной памяти, определяемое временем между двумя последовательными командами СЧИТЫВАНИЕ, составляет 2 мкс. Время между следующими друг за другом командами СЧИТЫВАНИЕ и ЗАПИСЬ может быть произвольным, но не менее 1 мкс. Время выборки информации, определяемое как задержка появления считанной информации на КШЧ относительно командного сигнала СЧИТЫВАНИЕ, равно 1 мкс.

Основными функциональными частями оперативной памяти являются (рис. 5.2) блок регистра адреса и предварительной дешифрации (БРА) и (ПДШ), блок адресной выборки (БАВ), блок разрядной выборки (БРВ), магнитный накопитель (МН), блок считывания (БСч) и блок местного управления (БМУ).

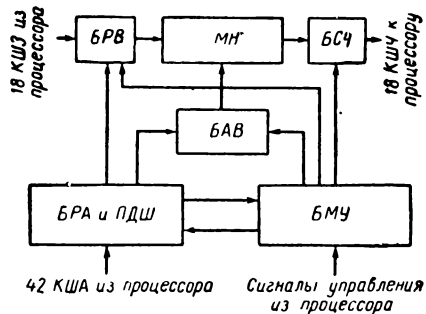


Рис. 5.2. Структурная схема оперативной памяти ЕС-3220

БРА и ПДШ предназначен для приема кода адреса, по которому осуществляется обращение к магнитному накопителю, предварительной дешифрации этого адреса и выдачи управляющих сигналов в блоки адрес-

ной и разрядной выборки, а также для выдачи в блок местного управления сигналов в случае несовпадения значения контрольных разрядов с числом значащих единиц в трех байтах адреса. БРА и ПДШ можно разделить на три блока: блок регистра, блок предварительной адресной дешифрации и блок предварительной разрядной дешифрации.

БАВ служит для выбора адресной шины, соответствующей адресу и признаку памяти, формирования стабилизированных импульсов считывания и записи и посылки их в выбранную адресную шину.

БРВ предназначен для выбора (в соответствии с адресом) одной из разрядных шин во всех группах разрядных шин, формирования стабилизированных токовых импульсов считывания во всех разрядах и стабилизированных токовых импульсов записи в тех разрядах, в которых необходимо записать единицу, и посылки этих импульсов в выбранные разрядные шины.

МН служит для хранения информации. Он состоит из запоминающих матриц на ферритовых сердечниках.

БСч предназначен для приема и усиления сигналов, поступающих из МН, отделения их от помех с помощью временного стробирования и амплитудной дискриминации, формирования выходных сигналов и посылки этих сигналов по КШЧ в процессор.

БМУ служит для выработки определенных серий управляющих сигналов и выдачи их в другие блоки оперативной памяти. Выработка этих серий осуществляется по сигналам управления, поступающим из процессора.

Рассмотрим более подробно структуру МН для модификации ЕС-3220-1 (рис. 5.3). Емкость его равна 64К байта (К-1024), причем с учетом одного контрольного разряда байт содержит 9 разрядов, или 32К двухбайтных (18-разрядных) слов. Магнитный накопитель состоит из 18 (по числу разрядов) ферритовых полей, каждое из которых служит для хранения одноименных разрядов всех запоминаемых 18-разрядных слов. Каждое ферритовое поле представляет собой прямоугольную матрицу размером 1024×32 , составленную, следовательно, из 32К ферритовых сердечников. Эта матрица разделена на две части (полуматрицы) размером 512×32 каждая; конструктивно эти полуматрицы расположены в двух разных плоскостях магнитного накопителя. Каждая полуматрица ферритового поля имеет 512 вертикальных координатных

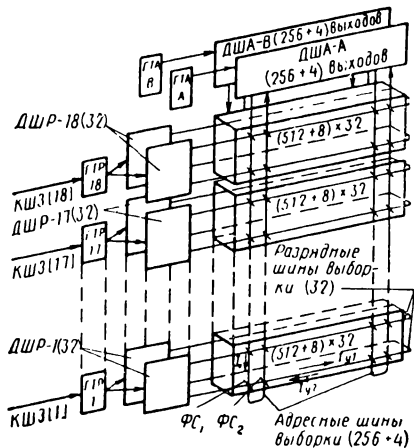


Рис. 5.3. Магнитный накопитель оперативной памяти ЕС-3220-1

шин, которые, будучи объединены попарно, образуют 256 адресных шин. Каждая адресная шина пронизывает 64 сердечника в каждой полуматрице — 32 прямой ветвью и 32 обратной.

Адресные шины, принадлежащие полуматрицам, расположенным в одной плоскости (с одной стороны) магнитного накопителя, связаны с выходами дешифратора адреса ДША-А или ДША-В. Каждый из этих дешифраторов имеет 256 выходов.

Кроме адресных дешифраторов, имеется 18 разрядных дешифраторов ДШР-1 — ДШР-18 на 32 выхода каждый. Выходы разрядного дешифратора i -го разряда ($i = \overline{1,18}$) связаны с разрядными шинами соответствующего ферритового поля. Ферритовое поле любого разряда содержит 32 разрядные шины выборки. Каждая разрядная шина состоит из двух ветвей, соединенных последовательно; одна из них пронизывает 512 сердечников одной полуматрицы, а другая ветвь — 512 сердечников другой полуматрицы. Формирование полутоков записи в разрядных дешифраторах управляется информацией, поступающей по кодовым шинам записи КШЗ [1] — КШЗ [18].

Адресные и разрядные дешифраторы возбуждаются сигналами с адресного и разрядного генераторов тока (ГТА и ГТР).

Выборка одного из 32К сердечников в каждом ферритовом поле осуществляется следующим образом. Пусть выбранными оказываются адресная и разрядная шины, проходящие через ферритовые сердечники ΦC_1 и ΦC_2 . По адресной шине проходит адресный ток I_x , величина которого равна $0,5I_m$. Через выбранную разрядную шину пропускается либо ток I_{y1} , либо ток I_{y2} противоположного направления, причем $I_{y1} = I_{y2} = 0,5I_m$. В первом случае избранным оказывается сердечник ΦC_2 , а во втором случае — сердечник ΦC_1 . Таким образом, за счет изменения направления разрядных токов в выбранной разрядной шине число выходов дешифратора ДША-А сокращается вдвое (256 вместо 512). То же самое относится к дешифратору ДША-В.

В каждой из двух плоскостей магнитного накопителя, кроме 256 адресных шин выборки основной памяти, имеются 4 дополнительных адресных шины, позволяющие получить необходимую емкость локальной (ЛП) и мультиплексной памяти (МП). Действительно, каждая адресная шина обеспечивает емкость в 64 двухбайтных слова, т. е. 128 байт. Следовательно, две дополнительные адресные шины (по одной в каждой плоскости) обеспечивают емкость 256 байт локальной памяти, а остальные 6 дополнительных шин (по 3 в каждой плоскости накопителя) — 768 байт мультиплексной памяти. С учетом дополнительных адресных шин дешифраторы ДША-А и ДША-В должны быть рассчитаны на $256 + 4$ выходов каждый.

В модификациях ЕС-3220-2 и ЕС-3220-3 имеются соответственно два и четыре магнитных блока, причем в обеих модификациях первый магнитный блок аналогичен рассмотренному, а во втором

блоке все 8 дополнительных адресных шин используются для получения емкости 1024 байта МП. Общая емкость мультиплексной памяти $768 + 1024 = 1792$ байта. В модификации ЕС-3220-3 в третьем и четвертом магнитных блоках дополнительные адресные шины не используются.

5.3. УСТРОЙСТВО И РАБОТА СОСТАВНЫХ ЧАСТЕЙ ОПЕРАТИВНОЙ ПАМЯТИ ТИПА 2,5D ДЛЯ МОДЕЛИ ЕС-1020

Блок регистра адреса. Блок регистра адреса предназначен для приема кода адреса, по которому осуществляется обращение из процессора, контроля этого кода на четность и выдачи управляющих потенциалов в другие блоки оперативной памяти.

Регистр адреса РА (рис. 5.4) является 21-разрядным, причем разряды объединены в три байта. Первые два байта 9-разрядные, включающие 8 информационных разрядов и один контрольный для контроля на четность. Третий байт неполный, 3-разрядный. Для всех типов памяти (ОП, ЛП и МП) адрес из процессора поступает импульсным парафазным кодом по 42 КША, по которым подается 18 информационных и 3 контрольных разряда адреса. Для адресации основной памяти используются 0—17 разряды РА, для адресации ЛП и МП — соответственно разряды 0—7 и 0—10 РА.

Функции отдельных разрядов и групп разрядов РА состоят в следующем:

- 0-й разряд — в дешифрации не участвует, он служит для выбора одного из двух байт считываемой информации устройства ЕС-2420;

- 1-й и 5-й разряды — для управления двумя предварительными разрядными дешифраторами ПДШР4;

- 2-й 3-й и 4-й разряды — для управления двумя предварительными разрядными дешифраторами ПДШР8;

- 6-й разряд — для определения направления тока I_y в разрядных шинах выборки; сигналы с триггера 6-го разряда РА поступают на вход дешифратора направления ДШ2Н;

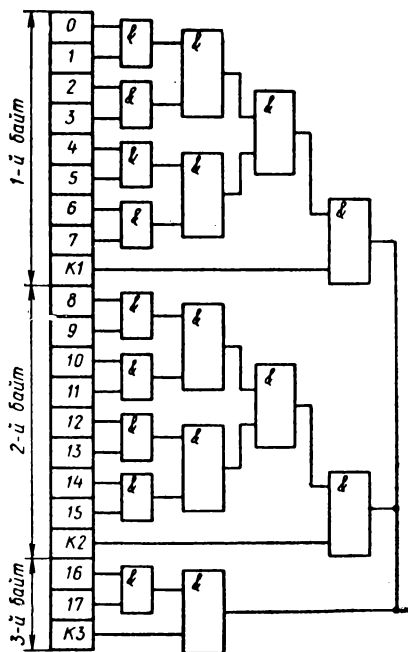


Рис. 5.4. Регистр адреса оперативной памяти ЕС-3220-1

— 7-й разряд — для выборки одного из двух (ДША-А или ДША-В) адресных дешифраторов в магнитном блоке;

— 8—11-й разряды — для управления предварительным адресным дешифратором ПДША16₁, который выбирает один из 16 координатных (адресных) ключей, определяющих выбор координаты (адресной шины) в группе адресных шин;

— 12—15-й разряды — для управления предварительным адресным дешифратором ПДША16₂, который выбирает один из 16 групповых ключей, определяющих выбор группы из 16 координат (адресных шин);

— 16—17-й разряды — для выбора номера магнитного блока при обращении к основной памяти, составленной из нескольких таких блоков;

— 10-й разряд — для определения номера магнитного блока при обращении к мультиплексной памяти;

— 5-й, 7—9-й, 14-й и 15-й разряды — совместно с признаками памяти используются для выбора одной из восьми секций обмотки считывания (выходной обмотки) магнитного блока;

— 11-й разряд — используется в качестве признака переполнения адресной сетки при обращении к мультиплексной памяти.

Каждый разряд регистра адреса представляет собой триггер с потенциальными связями и с парафазным занесением информации. Цепи для управления записью информации и цепи сброса в регистре отсутствуют.

Контроль на четность содержимого регистра осуществляется побайтно, причем в двух младших байтах с помощью четырехступенчатой схемы свертки (рис. 5.4). Первая ступень производит попарное сравнение разрядов адреса в байте, вторая и третья ступени — попарное сравнение потенциалов с выходов схем предыдущих ступеней. В четвертой ступени осуществляется попарное сравнение потенциалов с выхода третьей ступени с потенциалами триггеров контрольных разрядов. Контрольный разряд дополняет число значащих единиц в байте до нечетного. Если в контролируемый байт регистра адреса информация принята правильно, то на выходе элемента четвертой ступени устанавливается высокий потенциал, в противном случае — низкий.

Третий (трехразрядный с учетом контрольного разряда КЗ) байт проверяется на четность с помощью двухступенчатой схемы свертки. В первой ступени производится сравнение потенциалов с триггеров 16-го и 17-го разрядов, а во второй ступени — сравнение потенциалов с выхода элемента первой ступени и с выхода триггера контрольного разряда КЗ.

Результаты контроля всех трех байт поступают на элемент *ИЛИ*, расположенный в блоке местного управления, где, в случае неверного приема информации в РА, вырабатывается сигнал сбоя регистра адреса.

Блок предварительной разрядной дешифрации. Управление этим блоком осуществляется сигналами из блока регистра адреса

и блока местного управления, а предназначен он для выработки сигналов, управляющих работой блока разрядной выборки.

В состав блока предварительной разрядной дешифрации входят (рис. 5.5) два дешифратора на четыре выхода каждый — ПДШР_{4,1}, ПДШР_{4,2}, два дешифратора на восемь выходов каждый — ПДШР_{8,1}, ПДШР_{8,2}, один дешифратор направления на два выхода ДШ2Н и один дешифратор на два выхода ДШ2 для попарной выборки четных и нечетных блоков магнитных накопителей. Кроме того, в состав этого блока входят группы переходных элементов 4И и 8И, которые, принимая сигналы с выходов ПДШР и с выходов дешифратора ДШ2, преобразуют их, т. е. формируют по амплитуде и длительности, в сигналы, необходимые для запуска элементов блока разрядной токовой выборки.

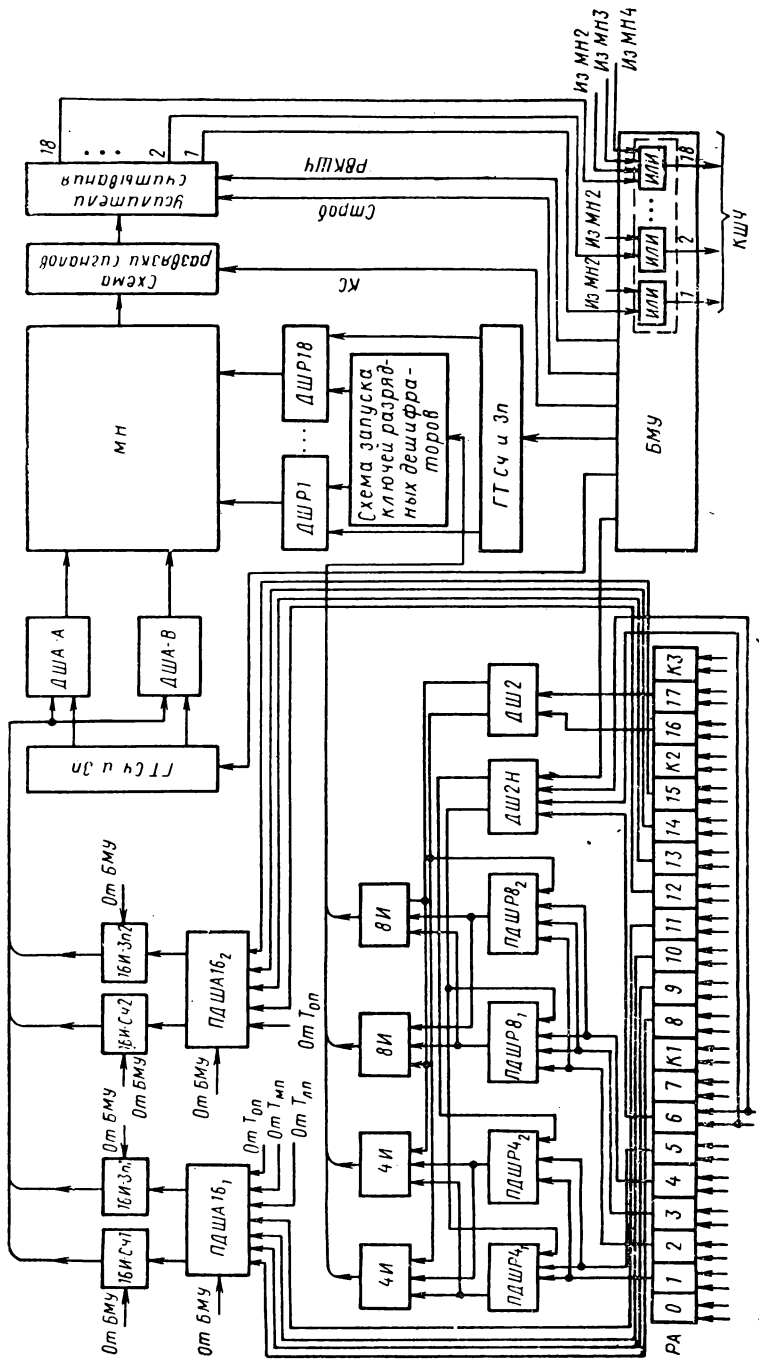
Дешифратор ДШ2Н служит для выбора направления разрядного тока I_y в разрядных шинах выборки магнитного накопителя. Он управляется, с одной стороны, потенциалами с выхода триггера 6-го разряда регистра адреса и импульсными сигналами с кодовых шин адреса этого разряда, что обеспечивает ускорение срабатывания дешифратора в такте считывания за счет исключения времени установления триггера 6-го разряда РА, и, с другой стороны, тактирующими импульсами из блока местного управления (БМУ). С одного из выходов дешифратора ДШ2Н сигналы поступают на входы дешифраторов ПДШР_{4,1} и ПДШР_{8,1}, а с другого выхода — на входы дешифраторов ПДШР_{4,2} и ПДШР_{8,2}. Следовательно, в зависимости от значения 6-го разряда РА срабатывают либо предварительные дешифраторы ПДШР_{4,1}, ПДШР_{8,1}, либо дешифраторы ПДШР_{4,2} и ПДШР_{8,2}, чем и определяется направление разрядного тока.

Дешифраторы ПДШР_{4,1} и ПДШР_{4,2} параллельно управляются сигналами с выходов 1-го и 5-го разрядов РА. В зависимости от распределения сигналов на выходах дешифратора ДШ2Н и от значения 1-го и 5-го разрядов РА будет возбужден один из четырех выходов либо дешифратора ПДШР_{4,1}, либо дешифратора ПДШР_{4,2}. Аналогично, в зависимости от распределения сигналов на выходах ДШ2Н и от значения 2-го, 3-го и 4-го разрядов РА возбуждается один из восьми выходов либо дешифратора ПДШР_{8,1}, либо дешифратора ПДШР_{8,2}.

Сигналы с выходов дешифраторов ПДШР_{4,1,2}, ПДШР_{8,1,2} поступают в блок разрядной выборки для управления работой ключей.

Блок предварительной адресной дешифрации. Этот блок предназначен для выработки сигналов, управляющих работой блока адресной выборки. Управление блоком предварительной адресной выборки осуществляется сигналами, поступающими с блока регистра адреса и блока местного управления БМУ.

В состав блока предварительной адресной дешифрации входят два дешифратора на 16 выходов каждый (ПДША16₁ и ПДША16₂) и группы переходных элементов считывания и записи 16И—Сч 1,2



42 КША из процессора

Рис. 5.5. Оперативная память ЕС-3220-1

и 16И—3п 1,2. Переходные элементы управляются сигналами с дешифраторов ПДША16₁ и ПДША16₂. Они необходимы для формирования управляющих сигналов и подачи их на токовые ключи адресных дешифраторов блока адресной выборки.

Выбор одного из 16 выходов дешифратора ПДША16₁ определяется значением 8—11-го разрядов регистра адреса. Управление этим дешифратором осуществляется, кроме того, потенциалами с выходов триггеров признаков памяти $T_{оп}$, $T_{мп}$, $T_{лп}$. При обращении к основной памяти с триггера $T_{оп}$ на дешифратор ПДША16₁ поступает потенциал, который разрешает выдачу сигналов с любого из 16 выходов дешифратора (нумерация выходов от 0 до 15). В этом случае номер выбранной шины будет определяться распределением потенциалов на выходах триггеров 8—11-го разрядов регистра адреса.

При обращении к мультиплексной памяти низким потенциалом с триггера $T_{мп}$ запрещается выборка выходных шин дешифратора ПДША16₁ с номерами от 4 до 15. Выбранной может быть одна из шин 0—3 в зависимости от значения 8-го и 9-го разрядов регистра адреса. Наконец, при обращении к локальной памяти потенциалом с триггера $T_{лп}$ блокируются все выходы дешифратора ПДША16₁ кроме нулевого. Следовательно, независимо от значения 8—11-го разрядов регистра адреса, в случае обращения к локальной памяти выбранным оказывается нулевая выходная шина дешифратора.

Дешифратор ПДША16₂ собран на трехвыходовых логических элементах. Каждый из этих элементов по двум входам управляется потенциалами с триггеров 12—15-го разрядов регистра адреса, а по третьему входу — потенциалом триггера основной памяти $T_{оп}$. При обращении к основной памяти высоким потенциалом с триггера $T_{оп}$ разрешается выборка любого из 16 выходов дешифратора, а какого именно — это зависит от значения 12—15-го разрядов регистра адреса. При обращении к МП или ЛП низким потенциалом с триггера $T_{оп}$ блокируется выбор любого из выходов дешифратора ПДША16₂.

Блок разрядной выборки. Блок разрядной выборки служит для формирования и послышки стабилизированных токовых импульсов считывания и записи в одну из 32 разрядных шин во всех 18 разрядах одного из четырех магнитных накопителей. Каждый накопитель обслуживается своим блоком разрядной выборки. В состав каждого из четырех блоков входят 18 однотипных разрядных токовых дешифраторов ДШР1—ДШР18 (рис. 5.5), схема одновременного запуска ключей этих дешифраторов, комплект из 18 разрядных генераторов тока (ГТ) считывания (Сч) и записи (Зп) со схемой запуска.

Разрядные токовые дешифраторы построены по принципу диодно-матричных переключателей с отдельными транзисторными ключами для токовых импульсов считывания и записи и двумя разделительными диодами на каждую разрядную шину

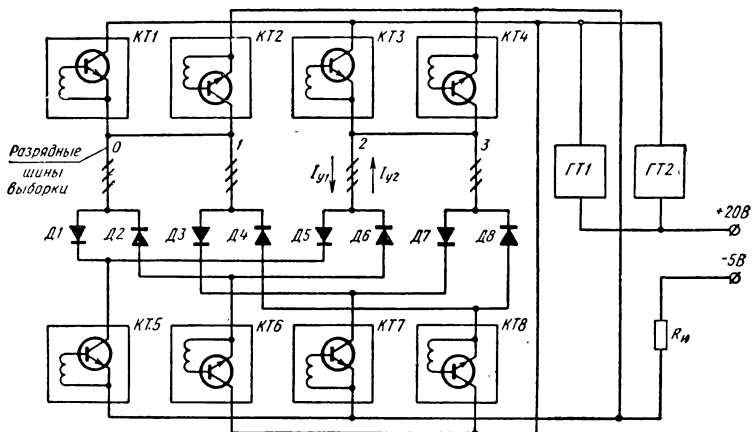


Рис. 5.6. Точковый дешифратор на 4 выхода

выборки. Принцип построения дешифратора рассмотрим на примере дешифратора на четыре выхода (рис. 5.6).

На рис. 5.6 изображены два генератора тока ГТ1 и ГТ2, групповые транзисторные ключи КТ1—КТ4, координатные транзисторные ключи КТ5—КТ8, разделительные диоды Д1—Д8, разрядные шины выборки 0—3, измерительный резистор R_n , источники питания +20 В и -5 В.

С помощью групповых ключей КТ1 и КТ3 осуществляется выборка двух групп разрядных шин (0—1 и 2—3), обеспечение прохождения импульсов токов положительной полярности. Групповые ключи КТ2 и КТ4 обеспечивают прохождение импульсов токов отрицательной полярности через те же группы разрядных шин. Выборка той или иной шины в группе производится координатными ключами: КТ5 и КТ7 — для тока одного направления и КТ6, КТ8 — для тока противоположного направления.

Пусть необходимо выбрать разрядную шину 2. Для этого необходимо по базе открыть ключи КТ3 и КТ5 (или КТ7). Путь тока I_{y1} через шину 2 будет проходить от источника +20 В через любой из генераторов тока ГТ1 или ГТ2, через ключ КТ3, шину 2, диод Д5, ключ КТ5 (или КТ7), резистор R_n и к источнику -5 В. Для пропускания тока через шину 2 в обратном направлении необходимо открыть ключ КТ4 и один из ключей КТ6 или КТ8. Тогда ток I_{y2} будет проходить от источника +20 В через любой из генераторов тока ГТ1 или ГТ2, через ключ КТ6 (или КТ8), диод Д6, шину 2, ключ КТ4, резистор R_n и к источнику -5 В. Таким образом, выбор направления разрядного тока через шину выборки определяется запуском соответствующей пары токовых ключей.

Объединение входов и выходов генераторов тока ГТ1 и ГТ2 устраняет необходимость выбора одного из них при переключе-

нии направления разрядных токов. Любой из этих генераторов может формировать токовые импульсы обеих полярностей. Однако целесообразно за одним генератором закрепить функции формирования стабилизированных импульсов тока одной полярности (разрядного тока считывания), а за другим — стабилизированных импульсов тока противоположной полярности (разрядного тока записи).

В каждом из 18 разрядных токовых дешифраторов имеется четыре пары групповых ключей, восемь пар координатных ключей и 32 пары разделительных диодов, оформленных в виде восьми интегральных диодных матриц — четырех с общим анодом и четырех с общим катодом. Четыре групповых ключа, соединенные с соответствующими разделительными диодными матрицами, выбирают одну из четырех групп разрядных шин, а восемь координатных ключей, соединенных с соответствующими разрядными шинами, обеспечивают выборку одной из восьми разрядных шин в выбранной группе для разрядного тока считывания. Другие четыре групповых ключа и восемь координатных ключей обеспечивают выборку одной разрядной шины для разрядного тока записи.

Запуск (открытие) одноименных ключей во всех 18 разрядных дешифраторах осуществляется от одного общего источника — схемы запуска. В состав схемы запуска входят ключи запуска, число которых (24) соответствует числу токовых ключей (групповых и координатных) в каждом разрядном дешифраторе. Ключи запуска разрядных дешифраторов в каждом магнитном накопителе разбиты на четыре группы, каждая из которых управляется по входным цепям сигналами, поступающими с соответствующих групп переходных элементов блока предварительной разрядной дешифрации.

Формирование стабилизированных импульсов тока считывания и записи в каждом разрядном дешифраторе производится двумя генераторами тока ГТ1 и ГТ2 с объединенными выходами. В каждом магнитном блоке для обслуживания разрядных дешифраторов имеется свой комплект из 18 пар генераторов тока и элементов их запуска. Срабатывание элементов запуска генераторов тока определяется совпадением двух сигналов из блока местного управления. Первый из этих сигналов поступает только на элементы запуска, относящиеся к выбранному магнитному накопителю, а второй, определяющий начало и длительность сигналов запуска генераторов тока, подается на элементы запуска всех магнитных накопителей. Формирование сигналов запуска генераторов тока в такте записи управляется потенциалами с кодовых шин записи. Поэтому в этом такте срабатывают только элементы запуска (а следовательно, и соответствующие генераторы тока, принадлежащие выбранному магнитному накопителю) в тех разрядах, куда необходимо записать единицу.

Блок адресной выборки. Как уже отмечалось, этот блок предназначен для выборки адресной шины и посылки в нее токовых

импульсов считывания и записи. В состав блока адресной выборки, в зависимости от модификации ЕС-3220, входят от 2 до 8 одинаковых адресных токовых дешифраторов ДША и от 4 до 16 генераторов тока со своими элементами запуска. Каждый магнитный накопитель обслуживается двумя адресными дешифраторами, расположенными на сторонах *A* и *B*. Адресный дешифратор служит для коммутации стабилизированных токовых импульсов в одну из 256 адресных шин при обращении к основной памяти или в одну из четырех дополнительных адресных шин при обращении к локальной или мультиплексной памяти.

Адресные дешифраторы построены по тому же принципу, что и разрядные дешифраторы. В состав адресного дешифратора ДША-*A* входят:

- 16 групповых ключей считывания $KT_{Cч1} — KT_{Cч16}$, соединенных с соответствующими диодными матрицами с общим катодом, и 16 групповых ключей записи $KT_{Эп1} — KT_{Эп16}$, соединенных с соответствующими диодными матрицами с общим анодом. Эти групповые ключи обеспечивают выборку одной из 16 групп адресных шин при обращении к основной памяти;

- один групповой ключ считывания $KT_{CчМЛ}$ и один групповой ключ записи $KT_{ЭпМЛ}$, соединенные с разделительными диодами. Эти ключи обеспечивают выборку группы шин, составленной из четырех адресных шин, при обращении к мультиплексной или локальной памяти;

- 16 координатных ключей считывания $K_{Cч1} — K_{Cч16}$ и 16 координатных ключей записи $K_{Эп1} — K_{Эп16}$, соединенных с соответствующими адресными шинами. С помощью этих ключей обеспечивается выборка одной из 16 адресных шин в выбранной группе адресных шин при обращении к основной памяти или одной из шин в группе из 4 адресных шин при обращении к мультиплексной или локальной памяти;

- 32 разделительных диода в цепи групповых ключей;

- 64 диода в цепи запуска ключей.

При открывании одного из групповых и одного из координатных ключей считывания замыкается цепь для прохождения через выбранную адресную шину адресного тока считывания. Для того чтобы через ту же шину пропустить адресный ток записи противоположного направления, необходимо открыть соответствующую пару ключей записи — один групповой ключ и один координатный.

Координатные ключи считывания в обоих адресных дешифраторах (ДША-*A* и ДША-*B*) каждого магнитного накопителя по одному из входов управляются сигналами с переходных элементов $16И—Сч1$ блока предварительной адресной дешифрации, запускаемых, в свою очередь, сигналами с предварительного дешифратора ПДША16₁. Управляющие сигналы с элементов $16И—Сч1$ поступают параллельно в два магнитных накопителя: в 1-й и 2-й или в 3-й и 4-й. Аналогично осуществляется управле-

ние координатными ключами записи и групповыми ключами считывания и записи, причем ключи $K_{3п1}—K_{3п16}$ управляются сигналами с переходных элементов 16И—3п1 блока предварительной адресной дешифрации, запускаемых сигналами с дешифратора ПДША16₁. Групповые ключи считывания и записи управляются сигналами с переходных элементов соответственно 16И—Сч2 и 16И—3п2, запускаемых сигналами с дешифратора ПДША16₂.

На другой вход всех групповых и координатных ключей каждого адресного дешифратора через диоды в цепи их запуска поступает отрицательный потенциал, разрешающий запуск выбранной пары ключей в одном из дешифраторов ДША-А или ДША-В. Разрешающие потенциалы для 8 адресных дешифраторов вырабатываются восемью эмиттерными повторителями, запуск которых осуществляется с выходов дешифратора блока местного управления, управляемого сигналами с триггера 7-го разряда регистра адреса и потенциалами с выходов дешифратора номера магнитного блока. При обращении к оперативной памяти срабатывает только один из восьми эмиттерных повторителей; следовательно, разрешается запуск ключей выборки в одном из двух адресных дешифраторов выбранного магнитного блока.

Для обслуживания каждого из восьми адресных дешифраторов имеется своя пара генераторов тока, один из которых формирует токовый импульс считывания, а другой — токовый импульс записи противоположной полярности. Запуск этих генераторов осуществляется так же, как и генераторов тока разрядных дешифраторов.

Блок считывания расположен в магнитном блоке. В состав блока считывания входят обмотки считывания ферритовых матриц магнитного накопителя, 18 усилителей считывания (по числу разрядов считываемого двухбайтного слова) и схема развязки сигналов, управляющих работой усилителей считывания. Обмотка считывания каждого разряда разбита на восемь секций, выходы которых связаны с входами восьми предварительных усилителей.

Каждый из 18 усилителей имеет 8 клапанируемых предусилителей и общий канал усиления, выпрямления, амплитудно-временной селекции и формирования выходного сигнала.

Схема развязки сигналов представлена восемью отдельными эмиттерными повторителями, выполненными на транзисторах. Клапанирующие сигналы (КС), выбирающие секции обмотки считывания, из блока местного управления поступают на вход (на базу) эмиттерных повторителей схемы развязки. С выходов эмиттерных повторителей (с эмиттеров транзисторов) сигналы подаются на соответствующие контакты усилителей считывания, сначала расположенных на стороне А магнитного блока, а затем на стороне В. Формирование клапанирующих сигналов производится в соответствии с адресом считываемой информации, т. е. при обращении по какому-либо адресу клапанирующий сигнал приходит на тот предварительный усилитель, на входе которого

должен появиться считанный кодовый сигнал, поскольку он связан с соответствующей секцией обмотки считывания данного разряда. Клапанирующие сигналы выполняют также функции предварительного стробирования, предотвращая прохождение на предусилители помех, возникающих по фронтам импульсов разрядных токов считывания.

В общем канале усилителя после усиления и выпрямления считанного кодового сигнала производится амплитудная дискриминация и временное стробирование. Амплитудная дискриминация осуществляется заданием порога срабатывания усилителя. Отделение полезного сигнала от помех (временное стробирование), обусловленных перемагничиванием полуизбранных ферритовых сердечников магнитного блока, осуществляется сигналом СТРОБ, который формируется в БМУ. Сигнал СТРОБ подается сначала на усилители считывания, относящиеся к стороне *A* магнитного блока, а затем — на усилители, относящиеся к стороне *B*. На выходе стробирующего каскада усилителя считывания сигнал появляется только в случае, если в течение времени, определяемого длительностью сигнала СТРОБ, сигнал на входе этого каскада превышает пороговый уровень. После стробирования считанный сигнал преобразуется по полярности и уровню и поступает на выходной каскад усилителя, где формируется по длительности.

Для формирования выходного сигнала по длительности на усилители считывания стороны *A*, а затем стороны *B* подается сигнал разрешения выдачи в кодовые шины чисел РВ КШЧ, который вырабатывается в БМУ. Передний фронт выходного сигнала усилителя считывания определяется передним фронтом сформированного по амплитуде считанного сигнала, а задний фронт — временем задержки заднего фронта сигнала РВ КШЧ.

С выхода усилителя считывания данного разряда сигнал поступает на элемент *ИЛИ* в БМУ. На этот же элемент поступают сигналы с усилителей считывания этого разряда других магнитных блоков.

Блок местного управления (БМУ) формирует сигналы, управляющие работой других блоков оперативной памяти. Формирование осуществляется по следующим сигналам, поступающим из процессора:

— СЧИТЫВАНИЕ — сигнал начала такта считывания;

— ОП, МП, ЛП — сигналы признака памяти соответственно основной, мультиплексной и локальной. В каждом такте обращения к оперативной памяти подается только один из этих признаков;

— ЗАПИСЬ — сигнал начала такта записи;

— АГ — аппаратное гашение.

Из блока местного управления в процессор подаются сигналы, представляющие собой признак отсутствия сбоя по адресации и признак сбоя регистра адреса.

По функциональному назначению выделяются следующие составные части БМУ:

- схема выработки серии управляющих сигналов такта считывания;
- схема выработки серии управляющих сигналов такта записи;
- узел триггеров управления;
- дешифратор номера блока;
- дешифратор секций обмотки считывания;
- схема приема сигналов с кодовых шин записи (КШЗ);
- схема запуска переходных элементов разрядной и адресной дешифрации.

Схема выработки серии управляющих сигналов такта считывания запускается командным сигналом СЧИТЫВАНИЕ из процессора. По сигналу СЧИТЫВАНИЕ срабатывают элементы этой схемы, на выходе которых появляется ряд импульсов, управляющих работой отдельных узлов и блоков оперативной памяти в такте считывания.

Назначение отдельных сигналов, формируемых схемой, состоит в следующем:

— сигнал РВ КШЧ поступает на все четыре магнитных блока для управления длительностью кодовых импульсов на выходе усилителей считывания;

— сигнал длительностью порядка 300 нс поступает на вход дешифратора секций обмотки считывания для задержки формирования переднего фронта сигнала клапанировки секций (клапанирующего сигнала) на время ~ 300 нс;

— сигнал, который поступает на вход дешифратора секций обмотки считывания, определяет своим задним фронтом окончание клапанирующего сигнала;

— сигнал, который поступает в схему формирования стробирующих сигналов, своей длительностью определяет время задержки выработки сигнала СТРОБ. В БМУ имеется возможность задавать начало стробирующих импульсов отдельно для каждого магнитного блока;

— сигнал, который поступает в блок предварительной разрядной дешифрации, в схему дешифрации номера блока и в схему запуска разрядных генераторов тока считывания, определяет начало и длительность выходных импульсов этих схем;

— сигнал, который поступает на вход дешифратора номера блока и в блок запуска переходных элементов разрядной и адресной дешифрации, определяет окончание стробирующих импульсов, а также начало и длительность сигнала запуска адресных генераторов тока;

— сигнал, который поступает на схему запуска формирователей, управляющих элементами предварительной адресной дешифрации, определяет начало и длительность адресных токов считывания.

Схема выработки серии управляющих сигналов такта записи запускается командным сигналом ЗАПИСЬ из процессора. По сигналу ЗАПИСЬ срабатывают элементы этой схемы, на выходе которых появляется ряд импульсов, управляющих работой отдельных узлов и блоков оперативной памяти в такте записи.

Выделим следующие сигналы на выходе схемы:

— сигнал, который поступает в блок предварительной разрядной дешифрации для управления выбором направления разрядных токов выборки;

— сигнал для запуска формирователей, управляющих по эмиттерным входам переходными элементами *И* предварительной разрядной и предварительной адресной дешифрации;

— сигнал для пропускания кодовых импульсов в схеме приема кодовых сигналов с КШЗ; этот же сигнал после инвертирования поступает в схему запуска генераторов тока, определяя длительность разрядных и адресных токов записи;

— сигнал для установки в исходное состояние триггеров в узле триггеров управления БМУ.

Узел триггеров управления состоит из триггеров признаков памяти $T_{оп}$, $T_{мп}$, $T_{лп}$ и других вспомогательных триггеров. При обращении к ЗУ в узел триггеров поступает один из сигналов признаков памяти (ОП, МП или ЛП), а также ряд кодовых сигналов. Сигнал признака памяти устанавливает соответствующий триггер $T_{оп}$, $T_{мп}$ или $T_{лп}$ в единичное состояние и сбрасывает в нулевое положение два других. Следовательно, в каждом такте считывания или записи в единичное состояние устанавливается только один из триггеров признаков памяти, который совместно с другими триггерами узла обеспечивает обращение к соответствующему виду памяти.

Дешифратор номера блока содержит схему выработки потенциалов номера магнитного блока, схему формирования сигнала СТРОБ и сигналов, указывающих на наличие или отсутствие сбоя адресации.

Основу схемы выработки потенциалов номера магнитного блока составляют дешифратор на четыре выхода, управляемый потенциалами триггеров 10-го, 16-го и 17-го разрядов регистра адреса и потенциалами триггеров $T_{оп}$, $T_{мп}$, $T_{лп}$. На одном из выходов этого дешифратора устанавливается потенциал низкого уровня при обращении к соответствующему магнитному блоку.

Дешифратор секций обмотки считывания предназначен для выработки одного из восьми управляющих сигналов, обеспечивающих выборку необходимой секции обмотки считывания во всех четырех магнитных блоках. Управление дешифратором осуществляется сигналами с 5, 7—9, 14 и 15-го разрядов регистра адреса, потенциалами триггеров признаков памяти и сигналами, поступающими из схемы выработки серии управляющих сигналов такта считывания.

Схема приема сигналов с КШЗ построена на ключевых логических элементах, каждый из которых представляет собой схему совпадения на два входа по сигналам низких уровней. Информация из процессора для записи в оперативную память поступает по 18 кодовым шинам записи, причем код 1 представляется сигналом низкого уровня, а код 0 — сигналом высокого уровня. На один из входов ключевого элемента по соответствующей КШЗ поступает кодовый сигнал, а на другой вход — стробирующий сигнал из схемы выработки серии управляющих сигналов такта записи.

На выходах тех элементов, на входы которых поступил код 1, формируется положительный сигнал для запуска формирователей. Импульсы с выхода этих формирователей, в свою очередь, запускают соответствующие разрядные генераторы тока записи.

Схема запуска переходных элементов разрядной и адресной дешифрации состоит из схемы запуска эмиттерных повторителей, адресных дешифраторов и формирователей, управляющих работой адресных генераторов тока считывания и записи, и схемы запуска формирователей, управляющих работой разрядных генераторов тока считывания и записи.

Первая из этих схем построена на восьми ключевых элементах, на входы которых поступают сигналы с выходов дешифратора номера блока и выходов триггера 7-го разряда регистра адреса. Запускающий сигнал высокого уровня появляется на выходе одного из этих восьми элементов. Этот сигнал обеспечивает запуск эмиттерного повторителя соответствующего адресного дешифратора, а также формирователей, управляющих срабатыванием одного из восьми адресных генераторов тока считывания и записи.

Элементы схемы запуска формирователей, управляющих работой разрядных генераторов тока считывания и записи, управляются потенциалами с выходов дешифратора номера блока. Сигналы с выходов этих элементов поступают на запуск формирователей, обеспечивая тем самым срабатывание разрядных генераторов тока считывания и записи соответствующего магнитного блока.

5.4. ФУНКЦИОНИРОВАНИЕ ОПЕРАТИВНОЙ ПАМЯТИ МОДЕЛИ ЕС-1020

В оперативной памяти выполняются чередующиеся друг за другом команды СЧИТЫВАНИЕ и ЗАПИСЬ, поступающие из процессора в блок местного управления. По команде СЧИТЫВАНИЕ осуществляется считывание 18-разрядного слова из выбранного блока оперативной памяти по адресу, поданному по 42-м кодовым шинам адреса, и выдача считанного слова на кодовые шины числа. По команде ЗАПИСЬ происходит запись информации, поступившей по кодовым шинам записи, в ту же ячейку оперативной памяти, из которой производилось считывание.

Содержимое регистра адреса между командами СЧИТЫВАНИЕ и ЗАПИСЬ не изменяется.

Рассмотрим порядок выполнения команды СЧИТЫВАНИЕ. Эта команда начинает выполняться после ее поступления в БМУ совместно с одним из трех признаков памяти ОП, МП или ЛП. Адрес ячейки, к которой производится обращение, поступает на регистр адреса либо заранее, либо вместе с командой СЧИТЫВАНИЕ. С приходом команды СЧИТЫВАНИЕ БМУ начинается формирование ряда распределенных по времени сигналов, предназначенных для запуска и управления работой элементов, узлов и блоков оперативной памяти. Одновременно с этим производится анализ поступившего адреса, признака памяти и признаков переполнения адресной сетки (с целью определения правильности адресации), а также контроль на четность содержимого регистра адреса.

В результате анализа содержимого 16 и 17-го разрядов регистра адреса вырабатывается один из четырех сигналов номера магнитного блока, который разрешает запуск элементов, управляющих работой соответствующего блока. При правильной адресации к магнитному блоку в БМУ формируется сигнал, представляющий собой признак отсутствия сбоя по адресации, и посылается в процессор. Выработка этого сигнала не производится, если магнитный блок, к которому производится обращение, в данной модификации оперативной памяти отсутствует или если имеет место переполнение адресной сетки при обращении к ОП и МП. Признаком переполнения адресной сетки при обращении к ОП служит отсутствие сигнала КША [—18] (т. е. сигнала по инверсной шине 18-го разряда адреса), а при обращении к МП — наличие кода 1 в 11-м разряде регистра адреса. В случае переполнения адресной сетки в БМУ производится блокировка выработки сигналов, необходимых для выполнения команды СЧИТЫВАНИЕ.

Если результат контроля поступившего адреса на четность оказывается отрицательным, в БМУ оперативной памяти формируется сигнал (признак сбоя регистра адреса), который отправляется в процессор. Однако считывание информации по этому адресу осуществляется. В процессоре по этому сигналу производится прием считанной информации, выдача ее по КШЗ совместно с командой ЗАПИСЬ (следовательно, считанная информация восстанавливается в той же ячейке памяти) и затем останов процессора с индикацией сбоя на пульте управления.

Дешифрация кода адреса, записанного в регистр адреса, осуществляется блоками предварительной адресной и предварительной разрядной дешифрации. В блоке предварительной разрядной дешифрации в зависимости от распределения сигналов на выходе дешифратора выбора направления разрядного тока ДШ2Н, управляемого потенциалами с выхода триггера 6-го разряда регистра адреса, а также в зависимости от значения 1—5-го раз-

рядов РА появляются разрешающие потенциалы на одном из четырех выходов либо дешифратора ПДШР_{4,1}, либо дешифратора ПДШР_{4,2} и на одном из восьми выходов либо дешифратора ПДШР_{8,1}, либо дешифратора ПДШР_{8,2}. Выходные шины дешифраторов ПДШР_{4,1,2} связаны с двумя группами переходных элементов 4И, одна из которых связана с нечетными магнитными блоками, а другая — с четными. То же самое имеет место в отношении дешифраторов ПДШР_{8,1,2} и двух других групп переходных элементов 8И. Одна группа элементов 4И и одна группа элементов 8И управляются сигналом с первого выхода дешифратора ДШ2, обеспечивающего попарную выборку четных и нечетных магнитных блоков, а другие две группы элементов 4И и 8И — сигналом со второго выхода этого дешифратора. Совпадение сигналов с дешифраторов ПДШР_{4,1,2} и ПДШР_{8,1,2} и с дешифратора ДШ2 происходит в одном из четырех переходных элементов одной из двух групп 4И и в одном из переходных элементов одной из двух групп 8И. На выходе этих выбранных элементов появляются сигналы, которые поступают в схемы запуска на отдельные входы ключей разрядных дешифраторов нечетных или четных магнитных блоков.

Объединенные входы всех этих ключей в МН1 управляются сигналом РА [—17], т. е. сигналом с инверсного выхода триггера 17-го разряда регистра адреса, а в МН3 — сигналом РА [17], т. е. сигналом с прямого выхода триггера 17-го разряда РА. Аналогичные связи имеются между объединенными входами ключей запуска в МН2 и МН4 и выходами триггера 17-го разряда РА. Следовательно, в зависимости от состояния триггера 17-го разряда регистра адреса сигнал на выборку магнитных блоков подается либо в схемы запуска ключей разрядных дешифраторов магнитных блоков МН1, МН2, либо в такие же схемы блоков МН3, МН4.

Для срабатывания ключей запуска разрядных дешифраторов необходимо совпадение сигналов с элементом И (с групп переходных элементов 4И и 8И) блока предварительной разрядной дешифрации и сигнала с выходов триггера 17-го разряда регистра адреса. Например, в модификации оперативной памяти ЕС-3220, где имеются все четыре магнитных блока, при выборе первого магнитного блока условия срабатывания ключей запуска разрядных дешифраторов будут выполняться только в МН1, так как:

в МН2 в схемы запуска ключей поступает сигнал с триггера 17-го разряда РА, но отсутствуют сигналы с элементов И блока предварительной разрядной дешифрации;

в МН3 в схемы запуска ключей поступают сигналы с элементов И, но отсутствует сигнал с триггера 17-го разряда РА;

в МН4 отсутствуют и те, и другие сигналы.

Выходы ключей запуска связаны со входами соответствующих групп токовых ключей выборки (групповых и координатных) всех 18 разрядных дешифраторов. Сигнал с выхода двух срабо-

тавших ключей запуска (один из них срабатывает по сигналу с выхода элементов $4И$ блока предварительной разрядной дешифрации, а другой — по сигналу с выхода элементов $8И$ этого блока) вызывают срабатывание пары соответствующих токовых ключей выборки во всех 18 разрядных дешифраторах. Выбранная пара токовых ключей в каждом разряде подсоединяет одну из 32 разрядных шин выборки к выходам генератора тока считывания данного разряда.

Запуск разрядных генераторов тока считывания ГТР-Сч осуществляется с помощью элементов их запуска Ф-Сч. Срабатывание элементов Ф-Сч происходит при совпадении двух сигналов, поступающих из схемы запуска переходных элементов разрядной и адресной дешифрации (точнее, из схемы запуска формирователей, управляющих работой разрядных генераторов тока считывания и записи) блока местного управления. Один из этих сигналов подается на элементы Ф-Сч только выбранного магнитного блока, а другой, определяющий начало и длительность импульсов запуска генераторов ГТР-Сч, на элементы Ф-Сч всех магнитных блоков. Следовательно, при срабатывании элементов Ф-Сч запускаются генераторы тока считывания всех 18 разрядов в выбранном магнитном блоке.

Направление тока считывания через выбранную разрядную шину в каждом разряде магнитного блока определяется значением 6-го разряда регистра адреса. Сигналы с выхода триггера 6-го разряда РА управляют дешифратором направления ДШ2Н. В зависимости от распределения сигналов на выходе дешифратора возбуждается одна из выходных шин предварительных дешифраторов ПДШР 4_1 и ПДШР 8_1 блока предварительной разрядной дешифрации или одна из выходных шин дешифраторов ПДШР 4_2 и ПДШР 8_2 . Это, в свою очередь, обеспечивает срабатывание соответствующих переходных элементов в группах элементов $4И$ и $8И$ блока предварительной разрядной дешифрации. Сигналы с выходов элементов групп $4И$ и $8И$ обеспечивают последовательный запуск сначала соответствующих ключей запуска, а затем одного из четырех групповых токовых ключей и одного из восьми координатных токовых ключей в каждом из 18 разрядных токовых дешифраторов выбранного магнитного блока. В зависимости от того, какая именно пара токовых ключей выбрана, направление разрядного тока будет тем или иным.

Помимо разрядных токов считывания I_y , формируемых в каждом из 18 ферритовых полей выбранного магнитного блока, для считывания информации из ячейки памяти необходимо сформировать токи считывания I_x в одной из 512 адресных шин выборки этого блока. Эта задача решается с помощью системы предварительной адресной дешифрации и системы адресной выборки.

В блоке предварительной адресной дешифрации при обращении к основной памяти магнитного блока осуществляется возбуждение по одной из 16 выходных шин дешифраторов ПДША16 $_1$

и ПДША16₂, управляемых сигналами с выходов триггеров соответственно 8—11-го разрядов и 12—15-го разрядов регистра адреса. При обращении к мультиплексной памяти производится блокировка всех выходов в дешифраторе ПДША16₂ и выходов 4—15 в дешифраторе ПДША16₁. Выбор одного из первых четырех выходов дешифратора ПДША16₁ с номерами 0—3 определяется состоянием 8—9-го разрядов регистра адреса. При этом, если обращение производится к МП в блоке МБ1, выбор нулевой шины дешифратора ПДША16₁ исключается. Если же обращение производится к МП, расположенной в блоке МБ2, возможен выбор любого из первых четырех выходов этого дешифратора. Наконец, при обращении к локальной памяти работа дешифратора ПДША16₂ полностью блокируется, а в дешифраторе ПДША16₁, независимо от значения 8—11-го разрядов регистра адреса, выбранной оказывается нулевая выходная шина.

Разрешающие потенциалы с выходов дешифраторов ПДША16_{1,2}, поступают на переходные элементы считывания 16И—Сч1,2, срабатывающие от сигналов с блока местного управления. Управляющие сигналы с переходных элементов 16И—Сч1,2 поступают на один из входов соответствующих групповых и координатных токовых ключей считывания адресных дешифраторов. На другие входы этих ключей подаются разрешающие потенциалы, формируемые в блоке местного управления с помощью дешифратора на восемь выходов (по числу адресных дешифраторов), который управляется сигналами с триггера 7-го разряда регистра адреса и потенциалами с выходов дешифратора номера магнитного блока. В результате обеспечивается открывание одного из групповых и одного из координатных токовых ключей считывания в одном из двух адресных дешифраторов выбранного магнитного блока.

При обращении к ОП открытый групповой ключ считывания из группы ключей КТ_{Сч}1 — КТ_{Сч}16 выбирает одну из 16 групп адресных шин (по 16 шин в каждой группе), а открытый координатный ключ считывания из группы К_{Сч}1 — К_{Сч}16 выбирает одну из 16 адресных шин в выбранной группе. При обращении к МП или ЛП открытый групповой ключ считывания КТ_{Сч}МЛ выбирает группу из четырех дополнительных адресных шин, а открытый координатный ключ считывания из группы К_{Сч}1 — К_{Сч}4 (т. е. в эту группу ключей входят первые 4 ключа из группы К_{Сч}1 — К_{Сч}16) — одну из адресных шин в этой группе.

Адресные дешифраторы выборки на сторонах А и В каждого магнитного блока обслуживаются своими парами генераторов тока, один из которых является генератором тока считывания ГТА-Сч, а другой — генератором тока записи — ГТА-Зп. Запуск этих генераторов осуществляется с помощью четырех групп переходных элементов (формирователей) 4Ф₁—4Ф₄, каждая из которых управляется сигналами с выходов дешифратора БМУ, определяющего номер и сторону МН, где расположена выбранная

адресная шина. В отличие от разрядных токов считывания, направление адресного тока считывания не зависит от значения адреса и одинаково во всех адресных дешифраторах.

В результате совместного воздействия токов выборки по разрядным и адресным шинам происходит считывание информации со всех ферритовых сердечников выбранной 18-разрядной ячейки оперативной памяти.

Считанные кодовые сигналы воспринимаются в каждом разряде одной из восьми секций обмотки считывания. Выходы восьми секций обмотки считывания каждого разряда связаны со входами соответствующих восьми клапанируемых предусилителей считывания, входящих в состав усилителей считывания. Клапанирующие сигналы, поступающие на входы предварительных усилителей, формируются в БМУ с помощью дешифратора секций обмотки считывания. Эти сигналы обеспечивают прохождение считанных сигналов в предусилителях всех 18 разрядов.

После прохождения через каскады усиления и выпрямления, которые являются общими для восьми секций обмотки считывания, считанный сигнал поступает на стробирующий каскад усилителя считывания. В стробирующем каскаде осуществляется временное стробирование по сигналу СТРОБ, который вырабатывается в четырех (по числу МН) схемах формирования этого сигнала. После стробирования считанный сигнал подается на выходной каскад усилителя, где по сигналу РВ КШЧ, выдаваемому схемой выработки серии управляющих сигналов такта считывания блока местного управления, он формируется по длительности.

Считанные сигналы, после формирования их по уровню и длительности, поступают на элементы *ИЛИ*, расположенные в блоке местного управления. Число этих элементов определяется разрядностью считываемой информации и равно 18. На каждый из них подаются считанные кодовые сигналы одноименных разрядов всех четырех магнитных блоков (обращение, естественно, производится только к одному из магнитных блоков). С выхода элементов *ИЛИ* считанные сигналы подаются на 18 кодовых шин числа (КШЧ).

Выполнение команды СЧИТЫВАНИЕ заканчивается после окончания всех сигналов, определяющих длительность импульсов считывания.

Команда ЗАПИСЬ начинает выполняться после ее поступления в блок местного управления. Адрес ячейки и признак памяти остаются теми же, которые были при выполнении предшествующей команды СЧИТЫВАНИЕ.

С приходом команды ЗАПИСЬ блок местного управления начинает выработку ряда распределенных по времени сигналов для запуска и управления работой элементов, узлов и блоков оперативной памяти. Блокировка выработки сигналов, необходимых для выполнения этой команды, в случае переполнения адресной

сетки производится точно так же, как и с приходом команды СЧИТЫВАНИЕ. Аналогично работает и система предварительной разрядной и адресной дешифрации с той лишь разницей, что во всех дешифраторах, управляемых сигналами СЧИТЫВАНИЕ и ЗАПИСЬ, будут выбраны другие выходы. Например, в блоке предварительной разрядной дешифрации срабатывают дешифраторы ПДШР₄₂ и ПДШР₈₂, если в такте СЧИТЫВАНИЕ срабатывали дешифраторы ПДШР₄₁ и ПДШР₈₁. В конечном счете это приведет к изменению направления разрядных токов в разрядных шинах выборки на обратное по сравнению с направлением этих токов при считывании. В системе адресной дешифрации будут выбраны другие группы переходных элементов, а также другие токовые ключи и генераторы тока (генераторы тока записи вместо генераторов тока считывания). Это приводит к изменению направления тока в выбранной адресной шине на обратное.

Запуск разрядных генераторов тока при выполнении команды ЗАПИСЬ существенно отличается от того, как производится их запуск во время считывания. При записи числа в оперативную память на один из входов 18 двухвходовых элементов совпадения, участвующих в формировании сигналов запуска разрядных генераторов тока (эти элементы находятся в схеме приема сигналов с КШЗ блока местного управления), подаются кодовые сигналы записываемого числа, а на второй вход — стробирующий сигнал из схемы выработки серии управляющих сигналов такта записи. В результате срабатывают только те элементы совпадения, которые подвергаются воздействию кодового сигнала 1 и стробирующего сигнала. На выходе сработавших элементов появляются сигналы, которые после их формирования с помощью формирователей подаются на запуск соответствующих разрядных генераторов тока записи. Следовательно, разрядные токи записи будут протекать в выбранных разрядных шинах только тех разрядов, где необходимо записать код 1.

В такте записи выработка сигналов выборки секций обмотки считывания, сигналов СТРОБ и РВ КШЧ не производится, вследствие чего сигналы, индуктируемые в обмотках считывания при перемагничивании ферритовых сердечников магнитного блока, через тракт усиления не проходят.

5.5. ОПЕРАТИВНАЯ ПАМЯТЬ МАШИНЫ ЕС-1030

Устройство памяти ЕС-3203 машины ЕС-1030 содержит собственную оперативную память, предназначенную для приема, хранения и выдачи информации и выполненную в виде двух автономных оперативных запоминающих устройств ОП1 и ОП2, и систему питания оперативной памяти, обеспечивающую необходимыми стабилизированными напряжениями ОП1 и ОП2.

Емкость одного ОЗУ равна 32К 36-разрядных слов, что составляет 128К байт. Время цикла записи-считывания равно 1,25 мкс (цикл с разрушением и регенерацией информации), время выборки слова — 0,8 мкс, тип запоминающего элемента — кольцевой ферритовый сердечник типа 5ВТ, принцип построения цепей выборки соответствует системе 2,5D. Для построения схем управления записью, считыванием и регенерацией использован комплекс интегральных схем серий 155 и 137. Параметры входных и выходных сигналов ОЗУ соответствуют параметрам сигналов комплекса интегральных схем серии 155.

Электропитание устройства осуществляется от трехфазной сети промышленной частоты 50 ± 1 Гц с номинальными значениями напряжения $3 \times 380/220$ В. Применяются следующие номиналы напряжений источников питания: $-2,4$ В; ± 5 В; $\pm 12,6$ В; ± 20 В; ± 27 В. Мощность, потребляемая от сети, равна 5 кВт. Габариты устройства (две типовые стойки ЕС ЭВМ) — $2425 \times 1060 \times 750$ мм, масса — не более 1500 кг.

Эксплуатация устройства может производиться при температуре окружающего воздуха от 5 до 40° С, относительной влажности воздуха при 30° С от 40 до 95%, атмосферном давлении от 6 до 10 кПа.

Устройство не имеет резонанса на частотах ниже 25 Гц.

Устройства ОП1 и ОП2 полностью идентичны как по составу, так и по техническим данным и принципу действия. Каждое из них состоит из магнитного блока (БЗМ — блок запоминающий магнитный) и электронных схем управления. Конструктивно устройства ОП1 и ОП2 расположены в одной стойке памяти, причем магнитные блоки БЗМ1 и БЗМ2 выполнены в виде конструктивно законченных устройств и электрически связаны с электронными схемами через разъемные соединения. Электронные схемы управления оформлены в виде съемных ТЭЗ на интегральных элементах серии 155 и частично серии 137. Используются 16 типов ТЭЗ.

Устройство питания ОП1 и ОП2 конструктивно выполнено в виде одной стойки, где расположены унифицированные блоки питания (УБП) и блок управления питанием (БУП). К этой стойке подключаются кабель первичного питания и кабель интерфейса питания.

Связь между процессором и устройством ЕС-3203 осуществляется через согласованную кабельную магистраль, причем к этой магистрали могут быть подключены два устройства ЕС-3203, т. е. четыре запоминающих устройства ОП1—ОП4.

Устройство ОП1 (рис. 5.7) состоит из следующих функциональных блоков:

- блок адресных регистров (БАР);
- блок дешифраторов (БДШ);
- блок адресных формирователей (БФА);
- блок разрядных формирователей (БФР);

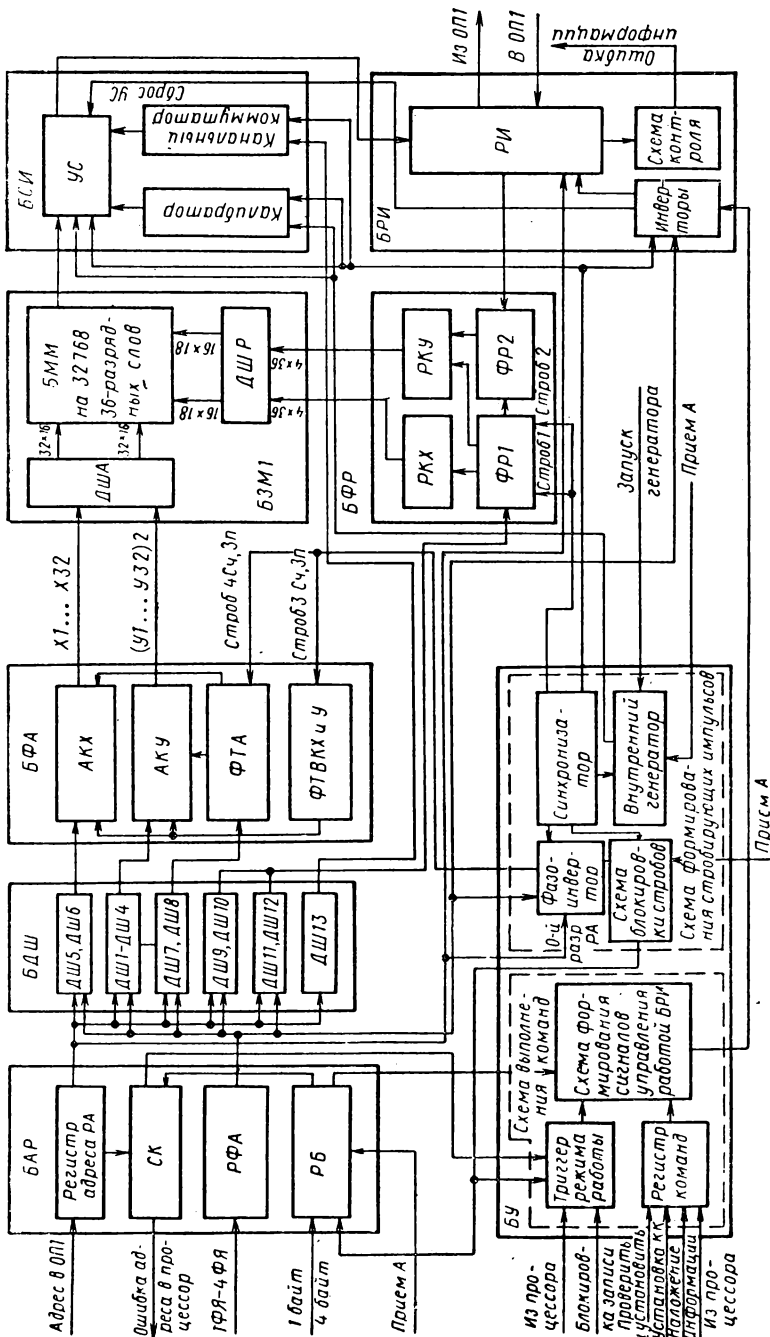


Рис. 5.7. Структурная схема запоминающего устройства ОП-1 машины ЕС-1030

- блок запоминающий магнитный (БЗМ1);
- блок считывания информации (БСИ);
- блок регистра информации (БРИ), или регистра числа;
- блок управления (БУ).

Рассмотрим назначение функциональных блоков устройства ОП1 и режимы его работы.

Блок адресных регистров предназначен для приема и запоминания адреса считываемой или записываемой информации, для запоминания признаков фиксированных адресов и признаков байт и для управления работой блока дешифраторов. В его состав входят регистр адреса (РА), регистр фиксированных адресов (РФА), регистр байт (РБ) и схема контроля (СК). Регистр адреса имеет 15 информационных и 2 контрольных разряда и служит для приема и хранения адреса ячейки памяти, к которой производится обращение, и управления работой блока дешифраторов. Регистр фиксированных адресов является четырехразрядным. Он служит для приема и хранения признаков фиксированных адресов 1ФЯ—4ФЯ и управления работой блока дешифраторов в режиме записи или считывания по фиксированным адресам. Регистр байт — пятиразрядный, он предназначен для приема и хранения четырех признаков байт (1 байт—4 байт) и значения одного контрольного разряда и для управления (через блок управления) работой регистра информации в режиме записи по байтам и в режиме «Проверить и установить» (см. ниже). Схема контроля служит для осуществления контроля содержимого регистра адреса и регистра байт. Контроль производится по байтам по схеме непрерывного контроля по модулю 2. При наличии ошибки сигнал ОШИБКА АДРЕСА выдается в процессор и в блок управления.

Блок дешифраторов (БДШ) служит для расшифровки 15-разрядного двоичного кода адреса и преобразования его в сигналы управления работой адресных и разрядных формирователей, а также усилителей считывания. В его состав входят 13 дешифраторов (ДШ), назначение которых заключается в следующем:

— ДШ1—ДШ4 на 16 выходов каждый — для управления работой адресных ключей Y;

— ДШ5, ДШ6 также на 16 выходов каждый — для управления работой адресных ключей X;

— ДШ7, ДШ8 на два выхода каждый — для управления работой формирователей адресного тока, а также для выбора одного из четырех дешифраторов ДШ1—ДШ4;

— ДШ9, ДШ10 на два выхода каждый — для управления работой разрядных ключей Y;

— ДШ11, ДШ12 на два выхода каждый — для управления работой разрядных ключей X;

— ДШ12 — для управления работой усилителей считывания. Управление осуществляется через дешифратор секций на восемь выходов (канальный коммутатор), расположенный в блоке считывания информации.

Блок адресных формирователей (БФА) служит для управления работой диодно-трансформаторных адресных дешифраторов ДША магнитного блока. В соответствии с кодом адреса он осуществляет выбор адресной шины БЗМ1 путем формирования в шине двуполярных импульсов тока считывания — записи положительной или отрицательной полярности. Блок адресных формирователей имеет адресные ключи X (АКХ) (32 пары ключей), адресные ключи Y (АКУ) (64 пары ключей), формирователи тока возбуждения ключей X и Y (ФТВК X и Y) и формирователи адресного тока (ФТА) считывания — записи. Адресные ключи X и Y предназначены для возбуждения соответственно шин X и Y адресных дешифраторов БЗМ1. При обращении к ОП1 в каждом цикле работы выбирается только одна пара ключей X и одна пара ключей Y .

Возбуждение адресных ключей X и Y осуществляется от формирователей тока возбуждения ключей, которые, в свою очередь, запускаются сигналами СТРОБ 3 Сч и СТРОБ 3 Зп, поступающими с блока управления.

В блоке имеется четыре формирователя адресного тока считывания—записи, управляемые дешифраторами ДШ7 и ДШ8 блока дешифраторов. В каждом цикле работы выбирается одна пара формирователей. Возбуждение формирователей осуществляется сигналами СТРОБ 4 Сч и СТРОБ 4 Зп, поступающими с блока управления.

Адресный ток считывания (записи) протекает от выбранного формирователя адресного тока через открытые ключи X и Y , диодно-трансформаторный ключ дешифратора адреса и адресную шину БЗМ1. Длительность и амплитуда тока задается формирователем адресного тока.

Направление адресного тока в шине БЗМ1 зависит от временного положения стробов при считывании и записи: при прямом направлении тока строб считывания (СТРОБ 3 Сч, СТРОБ 4 Сч) предшествует стробу записи (СТРОБ 3 Зп, СТРОБ 4 Зп), при противоположном направлении положение стробов меняется на обратное.

Блок разрядных формирователей (БФР) предназначен для управления работой диодно-трансформаторных разрядных дешифраторов ДШР магнитного блока. В соответствии с кодом адреса он осуществляет выбор разрядных шин по всем 36 разрядам путем формирования в шинах двуполярных импульсов тока считывания — записи или однополярных импульсов тока считывания в зависимости от управляющего кода, поступающего с регистра информации. В состав блока БФР входят разрядные ключи РКХ и РКУ (по четыре пары ключей), формирователи тока возбуждения ключей X и Y ФР1, формирователи разрядного тока считывания — записи ФР2. Разрядные ключи Y служат для возбуждения шин Y дешифратора одного разряда БЗМ1, а ключи X — для возбуждения шин X .

Возбуждение разрядных ключей X и Y осуществляется от формирователей тока $\Phi P1$, которые, в свою очередь, запускаются сигналами СТРОБ 1 Сч и СТРОБ 1 Зп с блока управления. Формирователи тока $\Phi P1$ по координате X управляются сигналами с дешифраторов ДШ11, ДШ12, а формирователи тока $\Phi P1$ по координате Y — сигналами с дешифраторов ДШ9, ДШ10.

Управление формирователями разрядного тока считывания — записи $\Phi P2$ осуществляется сигналами с регистра информации блока БРИ, причем в случае записи или регенерации нуля ток записи не формируется. Для возбуждения формирователей $\Phi P2$ используются импульсы с формирователей тока $\Phi P1$, при этом на входы последних с блока управления поступают сигналы СТРОБ 2 Сч и СТРОБ 2 Зп.

Сформированный импульс разрядного тока считывания (записи) протекает от формирователя $\Phi P2$ данного разряда через возбужденные ключи X и Y блока БФР, выбранный диодно-трансформаторный ключ дешифратора ДШР и разрядную шину БЗМ1. Параллельно возбуждаются 36 разрядных шин БЗМ1 — по одной в каждом разряде. Амплитуда и длительность импульса разрядного тока задаются формирователем тока $\Phi P2$.

Блок запоминающий магнитный (БЗМ) выполняет функции магнитного накопителя запоминающего устройства ОП1. В состав блока входят адресные и разрядные диодно-трансформаторные дешифраторы (ДША и ДШР) и блок магнитных матриц (БММ) на кольцевых ферритовых сердечниках типа 5ВТ размером $0,6 \times 0,4 \times 0,13$ мм. Емкость матриц — 32768 36-разрядных слов, принцип построения цепей выборки — по системе 2,5D с тремя проводами, т. е. полностью соответствует тому, который реализован в основной памяти машины ЕС-1020.

Выборка элементов памяти осуществляется по принципу совпадения импульсов тока в адресных и разрядных шинах БЗМ1, причем в возбуждаемые адресные шины подаются двуполярные (считывание и запись) импульсы тока, а в возбуждаемые разрядные шины — двуполярные при считывании и записи (регенерации) единицы и однополярные при считывании и записи (регенерации) нуля. На пересечении одной адресной и одной разрядной шины БЗМ1 находятся два ферритовых сердечника. Выборка одного из них производится путем изменения направления адресных токов в адресной шине. Считанные сигналы индуктируются в отдельных 8-секционных обмотках. В каждом разряде БЗМ1 имеется одна такая обмотка.

Адресный диодно-трансформаторный дешифратор ДША, общее число выходов которого равно 1024, предназначен для возбуждения адресных шин БЗМ1. Возбуждение координатных шин дешифратора ДША осуществляется сигналами с адресных ключей X и Y блока адресных формирователей.

Разрядный дешифратор ДШР, выполненный также по схеме диодно-трансформаторного матричного дешифратора, служит для

возбуждения разрядных шин БЗМ1. Схема дешифратора на один разряд имеет 16 выходов, следовательно, общее число выходов дешифратора ДШР равно 576 (16×36). Возбуждение координатных шин дешифратора ДШР осуществляется сигналами с разрядных ключей X и Y блока разрядных формирователей.

Блок считывания информации (БСИ) предназначен для усиления и выделения сигналов, считанных с блока БЗМ1, и формирования стандартных сигналов кодов 1 и 0. В состав блока входят усилители считывания (УСч), канальный коммутатор (дешифратор секций обмотки считывания БЗМ1) и калибратор (формирователи калибровочных сигналов). На вход БСИ поступают кодовые сигналы с БЗМ1, сигналы запуска калибратора и стробирующие импульсы с блока управления, а также сигналы на канальный коммутатор с дешифратора ДШ13 блока дешифраторов.

Усилители считывания обеспечивают усиление кодовых сигналов 1 и 0 и формирование их по амплитуде и длительности. Каждый усилитель имеет девять входов: восемь информационных (по числу секций обмотки считывания БЗМ1) и один вход калибровочного сигнала. В соответствии с кодом адреса на время воздействия считанных сигналов 1 и 0 открывается только один из восьми информационных входов, выбор которого производится с помощью дешифратора секций.

Калибровочный сигнал необходим для установки и динамической подстройки порога срабатывания амплитудного селектора усилителей считывания. Регулировка порога срабатывания амплитудного селектора производится путем автоматической регулировки амплитуды калибровочного сигнала на выходе калибратора.

Кодовые сигналы 1, 0 и калибровочный сигнал на входах усилителей считывания стробируются поступающими с блока управления стробами.

Формирование стандартного кодового сигнала 1 или 0 на выходе усилителя считывания осуществляется с помощью триггера. Стандартные сигналы 1 и 0 поступают в блок регистра информации (БРИ). Для установки триггера в исходное состояние подается сигнал СБРОС УСч.

Блок регистра информации выполняет функции запоминания и контроля информации, выдаваемой из ОП1 в режиме «Считывание» и поступающей в ОП1 в режиме «Запись». Кроме того, он служит для реализации дополнительных режимов работы «Запись РА в РИ», «Наложение информации», «Проверить и установить», «Установка контрольного кода». В состав блока входят регистр информации (РИ), схема контроля РИ и схема формирования управляющих сигналов, представляющая собой ряд инверторов, с помощью которых сигналы управления с блока управления размножаются и подаются на регистр информации.

Регистр информации 4-байтный, в каждом байте имеется девять разрядов — восемь информационных и один контрольный. Информация в регистре контролируется по байтам. Контроль

производится по схеме непрерывного контроля по модулю 2 (по нечетности). При обнаружении ошибки формируется сигнал ОШИБКА инф., который посылается в процессор.

Блок управления (БУ) предназначен для формирования сигналов управления работой ОП1. На вход блока поступают сигналы из процессора и блока адресных регистров, управляющие и стробирующие сигналы с выхода БУ распределяются на блоки БФА, БФР, БСИ, БРИ.

Основные части БУ — логическая схема выполнения команд и схема формирования стробирующих импульсов. Первая из них включает регистр команд «Проверить и установить», «Установка контрольного кода», «Наложение информации», триггер режима работы «Запись» — «Считывание» и логические схемы формирования сигналов управления работой БРИ.

Схема формирования стробирующих импульсов в своем составе имеет:

— синхронизатор блока управления, обеспечивающий формирование стробирующих и управляющих сигналов и выдачу их на блоки БФР, БФА (через фазоинвертор), БРИ, БСИ. Синхронизатор состоит из времязадающей цепочки на линиях задержки и формирователя стробов на триггерах;

— фазоинвертор адресных стробов, обеспечивающий в зависимости от значения нулевого разряда регистра адреса БАР, изменение на обратное временного положения импульсов СТРОБ 3 Сч и СТРОБ 3 Зп (аналогично для импульсов СТРОБ 4 Сч и СТРОБ 4 Зп), что необходимо для изменения направления адресных токов считывания — записи в шине БЗМ1;

— схему блокировки адресных и разрядных стробов, обеспечивающей сохранение информации в ОП1 при отключении питания;

— внутренний генератор, вырабатывающий стробирующие и запускающие импульсы для усилителей считывания и калибратора БСИ. Он осуществляет запуск калибратора и стробирование калибровочного сигнала в УСч. Внутренний генератор содержит линию задержки с инвертором, формирователи стробирующих и запускающих импульсов и управляющий триггер. Генератор может работать как в режиме внешнего запуска, так и в режиме самовозбуждения, благодаря чему осуществляется непрерывная автоматическая подстройка порога срабатывания амплитудного селектора усилителей считывания.

Режимы работы ОП1 (ОП2). Для устройства ОП1 (ОП2) предусматриваются следующие режимы работы.

1. «Запись» — в этом режиме производится запись информации в магнитный блок полным 36-разрядным словом или по байтам через регистр информации.

2. «Считывание» — осуществляется считывание из магнитного блока полного 36-разрядного слова, запись его на регистре информации (РИ) и выдача с РИ в процессор.

3. «Блокировка записи» производится прерывание режима «Запись» и переход в режим «Считывание» по сигналу из процессора.

4. «Проверить и установить» производится считывание информации с последующей записью всех «1» по заданному байту.

5. «Блокировка стробов» в этом режиме обеспечивается сохранение информации в устройстве при аварийном отключении питания.

6. Диагностические режимы работы: «Запись РА в РИ», «Запись (считывание) по фиксированному адресу», «Наложение информации», «Установка контрольного кода».

Режимы «Запись» и «Считывание» являются основными. В режиме «Запись», состоящем из такта считывания и такта записи, в ОП1 поступают:

- сигнал ЗАПИСЬ;
- сигнал разрешения приема адреса (ПРИЕМ А);
- код адреса на регистр адреса;
- код слова на регистр информации;
- признаки байт на регистр байт (1 БАЙТ—4 БАЙТ);

По сигналу ПРИЕМ А триггер режима работы устанавливается в состояние, соответствующее режиму «Запись», осуществляется прием кода адреса на регистре блока адресных регистров, запускается синхронизатор и выключается внутренний генератор блока управления. В соответствии с кодом адреса в блоке дешифраторов вырабатываются управляющие сигналы, которые поступают на адресные и разрядные формирователи тока и ключи X и Y, а также на каналный коммутатор блока БСИ. Синхронизатор блока управления в определенные моменты времени посылает стробирующие сигналы на формирователи токов в блоках БФА и БФР, на каналный коммутатор и усилители считывания.

На регистр информации импульсы приема информации поступают только по тем байтам, признаки которых переданы в регистр байтов. По другим байтам на регистр информации поступают сигналы управления регенерацией информации. В такте считывания режима «Запись» стробирующие импульсы на усилители считывания и соответственно кодовые сигналы с выходов усилителей на регистр информации поступают только по тем байтам, по которым нет записи новой информации, а производится регенерация считанной информации. После прохождения считанных кодовых сигналов через усилители считывания информационные входы усилителей запираются и открывается калибровочный вход, сигналом с синхронизатора запускается внутренний генератор, который начинает выдавать стробирующие и запускающие импульсы. Внутренний генератор работает до начала следующего цикла обращения к ОП1.

В такте записи режима «Запись» с каждого разряда регистра информации на разрядные формирователи тока записи ФР2 поступают сигналы, управляющие записью кодов 1 или 0 в данном разряде.

Таким образом, запись информации в ОП1 осуществляется по байтам, причем в режиме «Запись» на регистр информации принимаются кодовые сигналы тех байт, признаки которых переданы в блок адресных регистров. По другим байтам производится считывание и регенерация информации без выдачи ее в процессор.

В ОП1 предусмотрен автоматический переход с режима «Запись» в режим «Считывание», что бывает необходимо для сохранения информации в памяти. Такой переход осуществляется по сигналам ОШИБКА АДРЕСА и БЛОКИРОВКА ЗАПИСИ. По этим сигналам триггер режима работы блока управления переводится в состояние, соответствующее режиму «Считывание». По сигналу БЛОКИРОВКА ЗАПИСИ, кроме того, вырабатывается импульс, которым регистр информации вторично устанавливается в нулевое положение (первый раз он устанавливается в нулевое положение в начале цикла обращения к ОП1).

В режиме «Считывание» в ОП1 подаются сигналы: СЧИТЫВАНИЕ, ПРИЕМ А, ВЫДАЧА ИНФОРМАЦИИ ИЗ ОП1 и код адреса. По сигналу ПРИЕМ А триггер режима работы устанавливается в состояние, соответствующее режиму считывания, осуществляется прием кода адреса на регистр адреса, запускается синхронизатор и выключается внутренний генератор блока управления. В этом режиме регенерация и выдача информации из ОП1 производится только полным 36-разрядным словом.

Если запоминающее устройство работает в режиме «Считывание» и из процессора на регистр команд блока управления подан сигнал ПРОВЕРИТЬ и УСТАНОВИТЬ, то в соответствии с сигналом признака байта производится запись кода 1 по всем разрядам одного из четырех байтов.

В ОП1 предусмотрены четыре диагностических режима. В режиме «Установка контрольного кода» информация принимается из процессора на РИ и осуществляется установка кода в контрольном разряде каждого байта, после чего содержимое РИ переписывается в БЗМ1.

В режиме «Наложение информации» по правилу логического сложения осуществляется сложение кода РИ, сохранившегося от предыдущего цикла работы, и кода, поступающего из процессора на РИ по сигналу НАЛОЖЕНИЕ ИНФОРМАЦИИ. Результат сложения кодов переписывается в БЗМ1.

В режиме «Запись РА в РИ» код регистра адреса, сохранившийся от предыдущего цикла, передается на регистр информации и затем по сигналу ВЫДАЧА ИНФОРМАЦИИ ИЗ ОП1 — в процессор.

В режиме «Запись (считывание) по фиксированным адресам» имеется возможность произвести запись или считывание информации по четырем определенным адресам, которым соответствуют ячейки памяти с номерами 32, 33, 34 и 35. Обращение к этим ячейкам осуществляется по одному из четырех сигналов 1ФЯ—4ФЯ, являющихся признаками фиксированных адресов. На вход ре-

гистра фиксированных адресов поступает один из этих признаков, и, следовательно, обращение производится к одной из четырех фиксированных ячеек памяти независимо от значения кода в регистре адреса. Фиксированные ячейки БЗМ1 используются как обычные, если обращение к ним происходит через регистр адреса.

Контроль работоспособности устройства ЕС-3203 в составе модели ЕС-1030 осуществляется с пульта управления модели с помощью схем контроля оперативной памяти. Для этой цели используется также соответствующий программный тест.

5.6. ОПЕРАТИВНАЯ ПАМЯТЬ МАШИНЫ ЕС-1050

Оперативное запоминающее устройство ЕС-3205 машины ЕС-1050 включает собственно оперативную память (ОП) и систему питания ОП, обеспечивающую необходимыми стабилизированными напряжениями.

Емкость ОП равна 32К 72-разрядных слов, длина слова — восемь слогов, каждый слог имеет девять двоичных разрядов (восемь информационных и один контрольный). Время цикла записи-считывания равно 1,2 мкс, цикл с разрушением и регенерацией информации. Время выборки слова — 1,0 мкс. В качестве запоминающего элемента используется кольцевой ферритовый сердечник типа 5ВТ.

Схемы управления записью, считыванием и регенерацией построены в соответствии с системой 2,5D. В ОП используются интегральные схемы серии 155 и 137, причем параметры входных и выходных сигналов устройства соответствуют параметрам сигналов интегральных схем серии 137.

Конструктивно устройство ЕС-3205 состоит из двух стоек — стойки памяти и стойки питания. Связь между процессором и устройством осуществляется через согласованную кабельную магистраль. К ней могут быть подключены четыре устройства ЕС-3205.

Оперативная память состоит из двух магнитных блоков БЗМ (емкость каждого блока равна 32К 36-разрядных слов) и электронных схем управления, идентичных по составу, техническим данным и принципу действия.

Основные функциональные блоки оперативной памяти (рис. 5.8):

- блоки адресных регистров БАР1 и БАР2;
- блоки дешифраторов БДШ1 и БДШ2;
- блоки адресных формирователей БФА1 и БФА2;
- блоки разрядных формирователей БФР1 и БФР2;
- блоки запоминающие магнитные БЗМ1 и БЗМ2;
- блоки считывания информации БСИ1 и БСИ2;
- блоки регистра информации БРИ1 и БРИ2;
- блоки согласователей уровней БСУ1, БСУ2, БСУ3;
- блоки управления БУ.

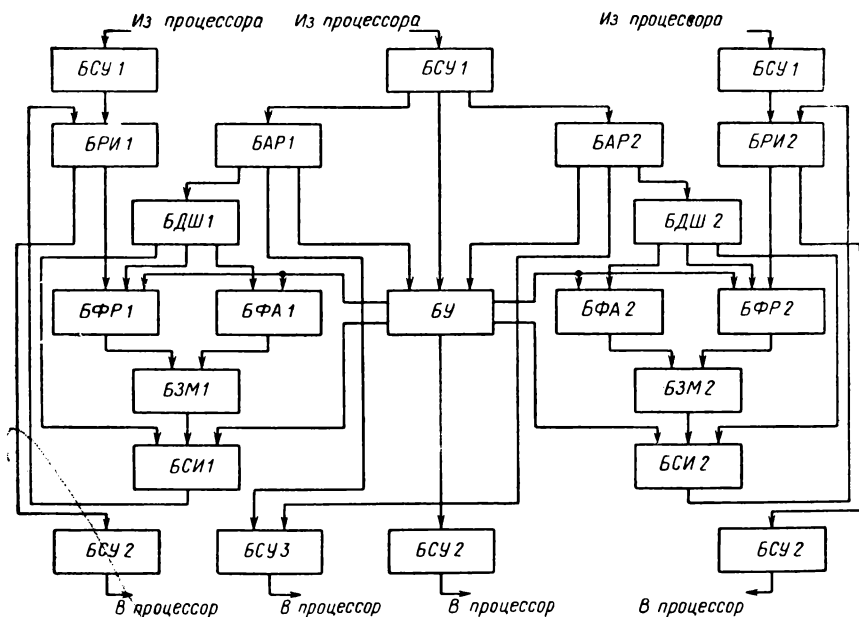


Рис. 5.8. Структурная схема оперативной памяти EC-3205 машины EC-1050

Эти блоки по своей структуре и выполняемым функциям аналогичны соответствующим блокам устройства EC-3203, поэтому ограничимся их краткой характеристикой.

Блоки адресных регистров имеют в своем составе два регистра адреса PA1 и PA2 (для каждого БЗМ — свой регистр), регистр маркеров и схемы контроля адреса. Регистр PA1 содержит 15 информационных разрядов и 2 контрольных. Он служит для приема и хранения адреса считываемой или записываемой информации в блок БЗМ1, а также для управления работой БДШ1. Регистр PA2 имеет 17 разрядов (15 информационных и 2 контрольных), он выполняет те же функции по отношению БДШ2 и БЗМ2. Регистр маркеров служит для приема и хранения признаков байт и контрольного разряда и управления работой регистров информации в режиме записи по байтам и в режиме «Проверить и установить». Контроль PA1 и PA2 и регистра маркеров осуществляется по схеме непрерывного контроля по модулю 2.

Каждый из блоков дешифраторов БДШ1 и БДШ2 содержит 13 дешифраторов. Он обеспечивает расшифровку 15-разрядного двоичного кода адреса и преобразования его в сигналы управления работой адресных и разрядных формирователей, а также усилителей считывания.

В блоке адресных формирователей имеются 64 пары ключей Y и 32 пары ключей X, служащих для возбуждения соответственно

шин Y и X адресных дешифраторов БЗМ, формирователи тока возбуждения ключей Y и X , формирователи адресного тока считывания — записи. В каждом цикле работы выбирается только одна пара ключей Y и одна пара ключей X в БФА1 и в БФА2. Изменение направления адресного тока в шине БЗМ1 и БЗМ2 осуществляется путем изменения временного положения адресных стробов при считывании и записи, т. е. так же, как и в устройстве ЕС-3203.

Блок разрядных формирователей, управляющий работой диодно-трансформаторных разрядных дешифраторов магнитного блока, в соответствии с кодом адреса осуществляет выбор разрядных шин БЗМ по всем 36 разрядам. Он состоит из разрядных ключей Y и X , формирователей разрядного тока считывания — записи и формирователей тока возбуждения ключей Y и X . Сигналы управления на формирователи разрядного тока записи подаются с регистра информации, причем при записи или регенерации нуля ток записи не формируется.

В составе каждого БЗМ имеются:

адресный дешифратор на 1024 выхода, выполненный по схеме диодно-трансформаторного матричного дешифратора и обеспечивающий возбуждение адресных шин магнитного блока;

диодно-трансформаторный разрядный дешифратор на 576 выходов (16×36). Дешифратор на один разряд имеет 16 выходов;

магнитные матрицы на кольцевых ферритовых сердечниках, общая емкость которых составляет 32 768 36-разрядных слов. Выборка запоминающего элемента осуществляется по принципу совпадения импульсов тока в возбужденных адресной и разрядной шинах. Через возбужденную адресную шину проходит двуполярный (считывание и запись) импульс тока, через разрядную шину — двуполярный импульс при считывании и записи (регенерации) единицы и однополярный при считывании и записи (регенерации) нуля. Для съема считанных сигналов в каждом разряде БЗМ используются отдельные 8-секционные обмотки.

Блок считывания информации, блок управления и блок регистра информации полностью идентичны соответствующим блокам устройства ЕС-3203. Регистр информации БРИ1 содержит 32 информационных разряда (от 0 до 31-го) и 4 контрольных (0К—3К), а регистр информации БРИ2, являющийся также 36-разрядным, имеет информационные разряды с 32-го по 63-й и контрольные разряды с 4К по 7К.

Блоки согласователей уровня БСУ1, БСУ2, БСУ3 предназначены для преобразования выходных сигналов оперативной памяти, представляющих собой стандартные сигналы интегральных схем серии 155, в входные сигналы процессора (стандартные сигналы интегральных схем серии 137), а также для преобразования выходных сигналов процессора в входные сигналы ОП.

Для устройства ЕС-3205 в составе модели ЕС-1050 предусмотрены следующие режимы работы:

- запись информации в БЗМ1 и БЗМ2 полным 72-разрядным словом или по байтам через регистр информации;
- считывание полного 72-разрядного слова из БЗМ на регистр информации и далее в процессор;
- прерывание режима записи и переход в режим считывания по сигналам из процессора, что бывает необходимо для сохранения информации в памяти;
- режим «Проверить и установить», в котором по заданному байту производится считывание информации с последующей записью всех единиц;
- диагностический режим «Наложение информации»;
- аварийный режим, в котором при аварийном отключении питания или выходе из строя процессора производится блокировка сигнала «Пуск ОП», вследствие чего информация в памяти не подвергается искажению.

5.7. ПОСТОЯННЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА

Постоянные запоминающие устройства (ПЗУ) используются в младших и средних моделях ЭВМ единой системы для хранения микропрограмм и включаются в состав процессоров. Это память трансформаторного типа, хранение двоичной информации в ней основано на наличии или отсутствии взаимной индукции между входными и выходными цепями.

Принципы построения и организацию функционирования ПЗУ рассмотрим на примере постоянной памяти, применяемой в процессоре машины ЕС-1020. Основные характеристики этой памяти: емкость 8192 64-разрядных слова, частота работы не более 1 МГц, время выдачи информации (слова) 650 нс, напряжение питания +5 В, +12,6 В, —5 В, потребляемый ток от этих источников питания соответственно равен 8; 1; 1,7 А.

Для хранения информации используются ферритовые сердечники с прямоугольной петлей гистерезиса (оксиферовые сердечники), работающие как линейные трансформаторы. Основой для построения ПЗУ трансформаторного типа служит числовая линейка (рис. 5.9, а), составленная из ферритовых сердечников, входных обмоток (шин) и выходных обмоток. Каждый сердечник предназначен для хранения одноименных разрядов запоминаемых чисел, следовательно, количество сердечников в линейке определяется разрядностью этих чисел. Емкость числовой линейки равна числу входных обмоток. Прошивка сердечников входными обмотками осуществляется в соответствии с информационным содержанием хранимых чисел: если в данном разряде запоминаемого числа содержится единица, соответствующая входная обмотка проходит через сердечник (на рис. 5.9, а это показано короткой наклонной линией, проходящей через точку пересечения обмотки и сердечника), если же в этом разряде содержится нуль, входная обмотка огибает сердечник (короткая наклонная линия на

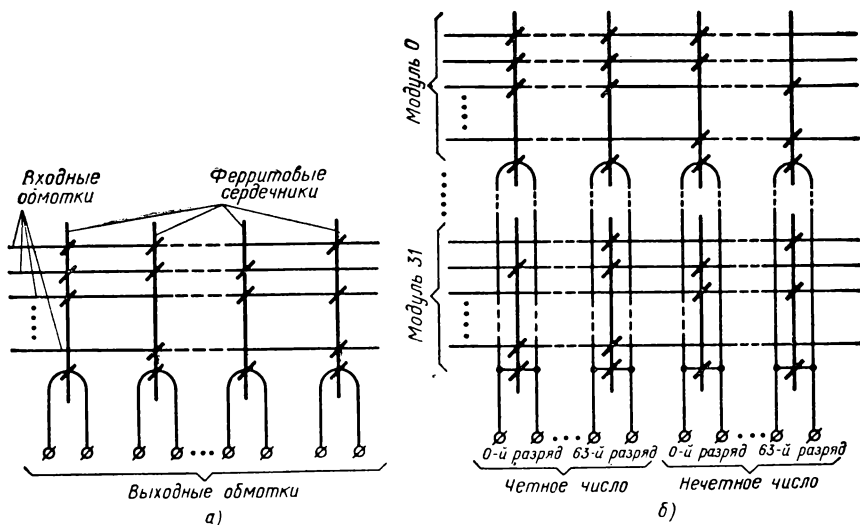


Рис. 5.9. Числовая линейка (а) и схема соединения выходных обмоток сердечников числовых линеек (б) в ПЗУ

рис. 5.9, а отсутствует). Таким образом, при записи информации во время монтажа ПЗУ машины ЕС-1020 используется принцип «прошито — не прошито». Выходные обмотки являются индивидуальными для каждого сердечника числовой линейки.

Емкость числовой линейки ограничивается не только площадью отверстия сердечника, но и взаимной наводкой между входными обмотками. При возбуждении одной из входных обмоток во всех остальных обмотках индуцируются токи, которые создают реакцию, противодействующую перемагничиванию соответствующих сердечников. Влияние этой реакции возрастает с увеличением числа входных шин. Значительное ослабление паразитных наводок в неизбранных входных шинах достигается при использовании более сложных способов прошивки сердечников, например, способа «прошито в одном направлении — прошито в другом направлении», а также прошивки двухпроводными (бифилярными) входными шинами.

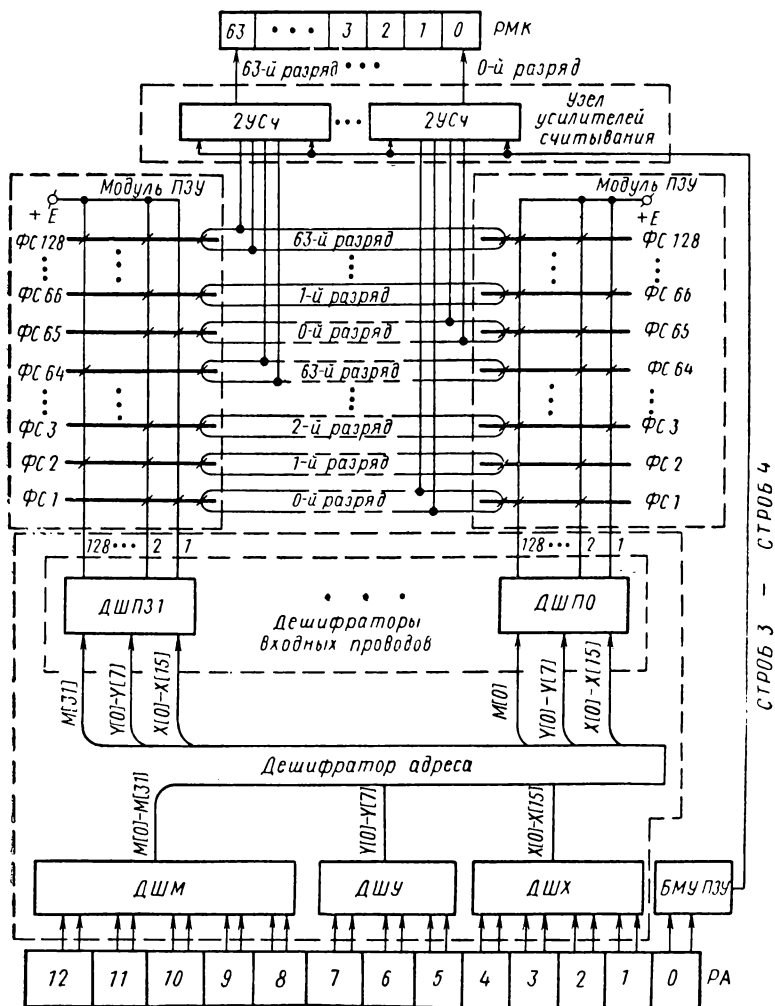
Постоянная память работает только в режиме считывания. Считывание может осуществляться неограниченное число раз, так как при считывании информация сохраняется.

Для считывания некоторого числа из ПЗУ в соответствующую входную обмотку числовой линейки подается импульс тока опроса. При этом в выходных обмотках сердечников, хранящих единицу, наводится э. д. с. сигнала. В выходных обмотках сердечников, хранящих нуль, сигнал отсутствует.

В ПЗУ машины ЕС-1020 одна числовая линейка служит для хранения 128 128-разрядных слов, т. е. в ее состав входит 128 ферритовых сердечников (ФС1 — ФС128), прошитых 128 входными ши-

нами. Такая числовая линейка представляет собой модуль постоянной памяти. Общее количество модулей в ПЗУ равно 32, нумерация модулей от 0 до 31. При обращении к ПЗУ из выбранного модуля считывается 128-разрядное слово, а так как передача в другие устройства осуществляются 64-разрядными словами, то считанное слово разбивается на два машинных слова — четное и нечетное. Выходные обмотки одноименных разрядов модулей соединены параллельно (рис. 5.9, б).

Схема ПЗУ машины ЕС-1020 приведена на рис. 5.10. Основные ее составные части следующие: регистр адреса РА, дешифратор



СТРОБ 3 — СТРОБ 4

Рис. 5.10. ПЗУ машины ЕС-1020

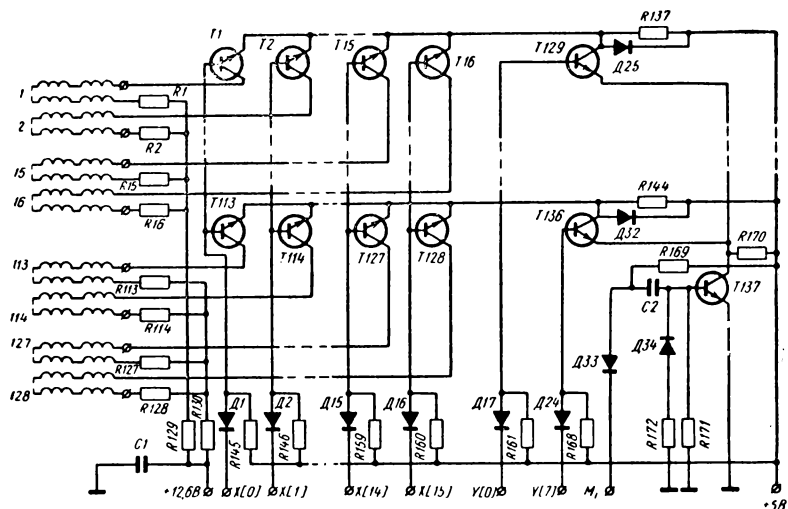


Рис. 5.11. Дешифратор провода ПЗУ машины ЕС-1020

адреса, узел хранения информации, узел усилителей считывания УСч, регистр микрокоманд РМк, узел контроля (на схеме не показан), блок местного управления БМУ.

Регистр адреса предназначен для приема, хранения и выдачи адреса считываемого четного или нечетного слова. Регистр содержит 13 информационных разрядов, пронумерованных от 0 до 12.

Дешифратор адреса обеспечивает выборку 64-разрядного четного или нечетного слова, соответствующего принятому адресу, и состоит из трех ступеней.

Первая ступень дешифрации имеет дешифраторы ДШХ, ДШУ и ДШМ, причем управление этими дешифраторами осуществляется сигналами, поступающими соответственно с [1—4] разрядов, с [5—7] разрядов и с [8—12] разрядов регистра адреса. Двухступенчатый дешифратор ДШМ с 32 импульсными выходами М [0] — М [31] определяет номер выбираемого модуля ПЗУ. Дешифраторы ДШХ и ДШУ обеспечивают выборку входной обмотки в избранном модуле. Выходы этих дешифраторов также импульсные.

Вторая ступень дешифрации предназначена для формирования импульса тока опроса в выбранной входной обмотке избранного модуля. Она состоит из 32 идентичных дешифраторов провода ДШП [0] — ДШП [31].

Дешифратор провода ДШП (рис. 5.11) представляет собой транзисторную матрицу 16×8, включающую транзисторы Т1—Т128. Кроме того, в состав дешифратора ДШП входят транзисторы Т129—Т136, в базовую цепь которых через диоды Д17—Д24 подаются импульсные сигналы Y [0]—Y [7] с дешифратора ДШУ, и транзистор Т137. Базовая цепь транзистора Т137 связана с соответствующим выходом дешифратора ДШМ. На вход транзи-

сторной матрицы сигналы $X [0]—X [15]$ поступают с дешифратора ДШХ. В коллекторную цепь транзисторов Т1—Т128 включены входные обмотки модуля памяти и токозадающие резисторы R1—R128.

В исходном состоянии все транзисторы дешифратора ДШП закрыты, так как на входы $X [0]—X [15]$, $Y [0]—Y [7]$, M_i поступают сигналы низкого уровня. Открывание одного из транзисторов Т1—Т128 матрицы осуществляется одновременной подачей трех положительных сигналов: с выхода дешифратора ДШХ на один из входов $X [0]—X [15]$, с выхода дешифратора ДШУ на один из входов $Y [0]—Y [7]$ и с i -го выхода дешифратора ДШМ на вход M_i . Например, если положительные сигналы поступили на входы $X [0]$, $Y [0]$, M_i , открываются транзисторы Т1, Т129, Т137 и замыкается цепь для импульса коллекторного тока, который проходит от источника +12,6 В через сопротивления R129 и R1, через первую входную обмотку модуля памяти и через указанные открытые транзисторы на землю. Таким образом, выбранными оказываются транзистор Т1 и первая входная обмотка модуля.

Третья ступень дешифрации необходима для выделения четного или нечетного 64-разрядного слова. В соответствии с состоянием нулевого разряда регистра адреса блок местного управления ПЗУ вырабатывает импульсный сигнал СТРОБ 3 или СТРОБ 4, который, поступая на соответствующие входы усилителей считывания, разрешает прохождение выбранного слова на регистр микрокоманд.

В узле хранения информации в качестве запоминающих элементов применяются малогабаритные ферритовые сердечники. Модули памяти этого узла объединены в четыре блока модулей БМ1—БМ4 по восемь модулей в каждом блоке. В состав каждого модуля, кроме 128 сердечников и 128 входных обмоток, входит соответствующий дешифратор провода. Входные обмотки выполнены методом печатного монтажа на фторопластовой подложке.

Узел усилителей считывания предназначен для разделения четного и нечетного слов, усиления, дискриминации и хранения считанных сигналов. Он состоит из 64 однотипных каналов, в каждом из которых имеется два независимых усилителя. На усилители, принадлежащие одному и тому же каналу, поступают сигналы одноименных разрядов четного и нечетного слов всех модулей памяти. Разрешение на прохождение выбранного слова осуществляется импульсом СТРОБ 3 или СТРОБ 4, причем первый из них поступает на выходы тех усилителей в каждом канале, на входы которых подаются сигналы разрядов четного слова, а второй — на выходы тех усилителей, на входы которых подаются сигналы разрядов нечетного слова. Сброс ранее записанной информации перед занесением новой производится импульсом сброса, поступающим с блока местного управления ПЗУ.

Регистр микрокоманд служит для приема и хранения считанного слова и содержит 64 разряда, из которых 62 разряда (от 0-го до 61-го) являются информационными, а два разряда (62-й и 63-й)

— контрольными. Занесение информации в регистр производится парафазными импульсными сигналами, поступающими от усилителей считывания. Передача информации из регистра микрокоманд в другие устройства процессора осуществляется также в парафазном коде.

Узел контроля предназначен для формирования и выдачи сигналов СБОЙ РА, СБОЙ РМк. Он состоит из узла контроля регистра адреса и узла контроля регистра микрокоманд. Узел контроля регистра адреса выполнен по стандартной схеме свертки. Он осуществляет сложение по модулю 2 содержимого разрядов 0—12 РА и 62-го разряда регистра РМк. В 62-м разряде регистра РМк хранится информация, дополняющая свертку разрядов РА до нечетного числа. Результат свертки передается в вычислительное устройство машины по шине СБОЙ РА. Узел контроля РМк осуществляет сложение по mod 2 содержимого разрядов [0—63] и также выполнен по стандартной схеме свертки. В 63-м разряде хранится информация, дополняющая свертку разрядов [0—62] до нечетного числа. Результат свертки передается в вычислительное устройство по шине СБОЙ РМк.

Блок местного управления ПЗУ предназначен для формирования в необходимые моменты времени управляющих импульсов определенной длительности.

АРИФМЕТИКО-ЛОГИЧЕСКИЕ
УСТРОЙСТВА ПРОЦЕССОРОВ6.1. ОСОБЕННОСТИ АРИФМЕТИКО-ЛОГИЧЕСКИХ
УСТРОЙСТВ ПРОЦЕССОРОВ РАЗЛИЧНЫХ
МОДЕЛЕЙ

В арифметико-логических устройствах (АЛУ) процессоров выполняются все арифметические и логические операции, набор которых определяется системой команд ЕС ЭВМ. Все модели ЭВМ Единой системы рассчитаны на выполнение единого универсального набора операций, включающего операции над операндами с фиксированной запятой, с плавающей запятой и над полями переменной длины, в том числе и над числами в десятичной системе счисления. Этим выполняется одно из требований обеспечения прямой программной совместимости моделей на уровне программ, представленных на машинном языке.

Обобщенная структурная схема АЛУ процессоров всех моделей может быть изображена в виде, приведенном на рис. 6.1. В основном она содержит четыре главные составляющие:

группу регистров R , предназначенных для приема и размещения надлежащим образом операндов, над которыми должны производиться действия при выполнении очередной операции;

операционную часть O , где осуществляются преобразования операндов согласно машинным алгоритмам арифметических, логических и других операций, на выполнение которых рассчитано АЛУ;

схемы контроля K , обеспечивающие непрерывный оперативный контроль работы АЛУ, а при обнаружении систематических сигналов ошибок — его диагностику с разрешающей способностью, соответствующей возможностям системы контроля, применяемой в модели ЭВМ;

схемы управления $У$, где вырабатываются управляющие сигналы $УС$, координирующие взаимодействие всех блоков АЛУ между собой и с другими блоками процессора, тем самым обеспечивая выполнение требуемых последовательностей микроопераций, соответствующих исполняемым операциям.

В моделях ЕС-1020 и ЕС-1030 применен микропрограммный принцип управления. Поэтому схемы управления АЛУ в них строятся на основе дешифраторов микрокоманд, выбираемых

в определенных последовательностях из ПЗУ микропрограмм. В модели ЕС-1050 реализован аппаратный принцип управления. В ее процессоре имеется блок управления, который принимает и дешифрирует код операции КОп очередной команды программы и обеспечивает выработку последовательности управляющих сигналов, необходимых для ее исполнения.

В моделях осуществляется гибкое управление выполнением операций. Последовательность действий по исполнению каждой команды зависит от особенностей операндов и получающихся промежуточных и окончательных результатов их преобразования. Для этого в операционной части АЛУ на разных этапах выполнения операций производится анализ преобразуемой информации. Результаты его в виде ответных сигналов-признаков СП поступают в схемы управления.

На основе анализа получающихся результатов в конце исполнения определенных команд схемы управления формируют признак результата ПР, который в виде двухразрядного кода условия заносится в слово состояния программы ССП. Переход к управлению исполнением каждой очередной команды строится по асинхронному принципу. При наличии в процессоре командной и преобразуемой информации действия в АЛУ по выполнению следующей операции могут начинаться сразу после завершения предыдущей операции. Для этого в схемах управления АЛУ формируется сигнал конца операции СКО. Управление выполнением следующей операции начинается по сигналу начала операции СНО, вырабатываемому в схемах центрального управления процессора.

В регистры Р операнды для очередной операции выбираются либо из местной оперативной памяти (регистров общего назначения РОН или регистров для операндов с плавающей запятой РПЗ), либо из основной оперативной памяти ООП. Результаты операций из регистров АЛУ отсылаются также в РОН, РПЗ или ООП.

Количество и разрядность регистров АЛУ в разных моделях ЭВМ Единой системы различны. В ЕС-1020 в основном регистры 8- и 16-разрядные, в ЕС-1030 — 32-разрядные, в ЕС-1050 — 64-разрядные. Это является одной из причин, обуславливающих различие в способах приема, размещения и оперирования данными в моделях в процессе выполнения операций.

По составу и устройству операционные части АЛУ разных моделей также различны (рис. 6.2). В модели ЕС-1050 операционная часть состоит из двух блоков с явно выраженными сумматорами в каждом из них (рис. 6.2, а):

— блока арифметического сумматора БАС и

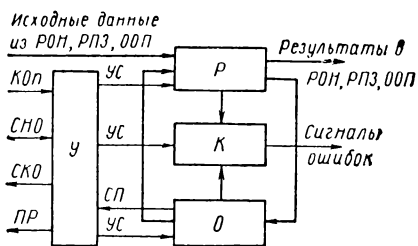


Рис. 6.1. Обобщенная структурная схема АЛУ процессоров

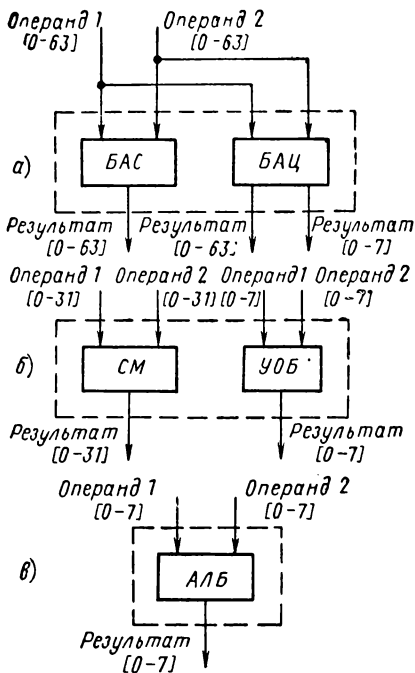


Рис. 6.2. Состав операционной части АЛУ процессора:

а — ЕС-2050; б — ЕС-2030; в — ЕС-2020

операнда одновременно по 8 байт. Блок может выдавать результат либо побайтно, либо параллельно по 64 разряда.

В модели ЕС-1030 основу операционной части (рис. 6.2, б), выполненной в едином блоке с регистрами, составляют 32-разрядный универсальный сумматор СМ и узел для логической обработки байт УОБ. В сумматоре выполняются действия как над двоичными, так и над десятичными числами. Схема УОБ предназначена для выполнения логических операций над операндами как фиксированных, так и переменного форматов. Она используется также для сопоставления величин порядков операндов с плавающей запятой.

В АЛУ модели ЕС-1020 операционная часть представляет собой универсальный коммутируемый 8-разрядный арифметико-логический блок АЛБ (рис. 6.2, в). Особенность его состоит в том, что он не имеет явно выраженного сумматора. Все арифметические и логические операции универсального набора, соответствующего системе команд ЕС ЭВМ, реализуются коммутируемой комбинационной схемой как совокупности элементарных действий: двоичного и десятичного сложения и вычитания, сдвига на разряд влево, вправо, поразрядных операций логического сложения, умножения и сложений по mod 2, а также операций сквозной

— блока цифровой десятичной арифметики БАЦ.

Блок БАС предназначен для выполнения арифметических и логических действий над операндами фиксированных форматов. Основу его составляет 64-разрядный двоичный сумматор параллельного действия. Благодаря возможности автоматической перекоммутации, блок БАС может принимать и обрабатывать операнды всех фиксированных форматов (от полуслова до двойного слова).

В блоке БАЦ производится только побайтовая обработка операндов. Для действий над десятичными числами в блоке имеется 8-разрядный десятичный сумматор. Кроме того, для логических операций над полями переменной длины предусмотрен 8-разрядный логический коммутатор. Два дополнительных 64-разрядных регистра позволяют блоку БАЦ принимать для обработки от каждого

передачи операндов через блок. В результате процессы выполнения операций над многобайтовыми операндами слагаются из последовательного выбора и преобразования пар их байт по микропрограммам этих операций. При этом из локальной или основной оперативной памяти операнды выбираются или записываются туда по два байта за одно обращение к ней.

Различия в составе и принципе действия АЛУ явились результатом разных требований к производительности моделей и желаемому объему их оборудования, предъявленных разработчикам при проектировании ЭВМ Единой системы. В свою очередь, особенности принятых решений обусловили разнообразие принципов построения и реализации в различных моделях машинных алгоритмов одинаковых операций (см. гл. 3). Особенности построения АЛУ обусловили также и различия в схемах контроля.

6.2. СТРУКТУРЫ И ОСНОВНЫЕ ИНФОРМАЦИОННЫЕ СВЯЗИ АРИФМЕТИКО-ЛОГИЧЕСКИХ УСТРОЙСТВ ПРОЦЕССОРОВ

АЛУ процессора ЕС-2050. Исключая схемы контроля и цепи управляющих сигналов, структурная схема АЛУ процессора ЕС-2050 может быть изображена, как показано в нижней части рис. 6.3. Основой операционной части являются сумматоры (двоичный СМ и десятичный ДСМ) и логический коммутатор ЛК в блоках арифметического сумматора БАС и цифровой десятичной арифметики БАЦ. В этих блоках имеется ряд регистров, из которых на схеме показаны: регистры результатов РР и РРБ, а в блоке БАЦ — еще два 64-разрядных входных регистра данных РВД1 и РВД2, два 8-разрядных входных регистра десятичного сумматора и логического коммутатора РСМ1 и РСМ2 и регистр десятичной коррекции РСМК.

Кроме перечисленных, в регистровую часть АЛУ входят четыре 64-разрядных регистра: входные — РВ1 и РВ2 и буферные РБ1 и РБ2. Они составляют основу блока арифметических регистров, который в целом предназначен для приема операндов в АЛУ и подготовки их для выполнения над ними операции в БАС, а также для формирования и выдачи результатов из АЛУ. Помимо этого, в БАЦ непосредственно выполняются логические операции *И*, *ИЛИ* и *Исключающее ИЛИ* над операндами, имеющими фиксированные форматы, а также осуществляются необходимые сдвиги операндов.

Операнды, требуемые для арифметических и логических операций, могут находиться либо в РОН или РПЗ, либо в ООП. В процессоре ЕС-2050 РОН и РПЗ образуют блок регистровой памяти БЦР. Из этого блока из регистров с адресами R_1 и R_2 операнды выдаются через выходные информационные регистры РИ1 и РИ2. Далее они поступают в регистры чисел РЧ1 и РЧ2

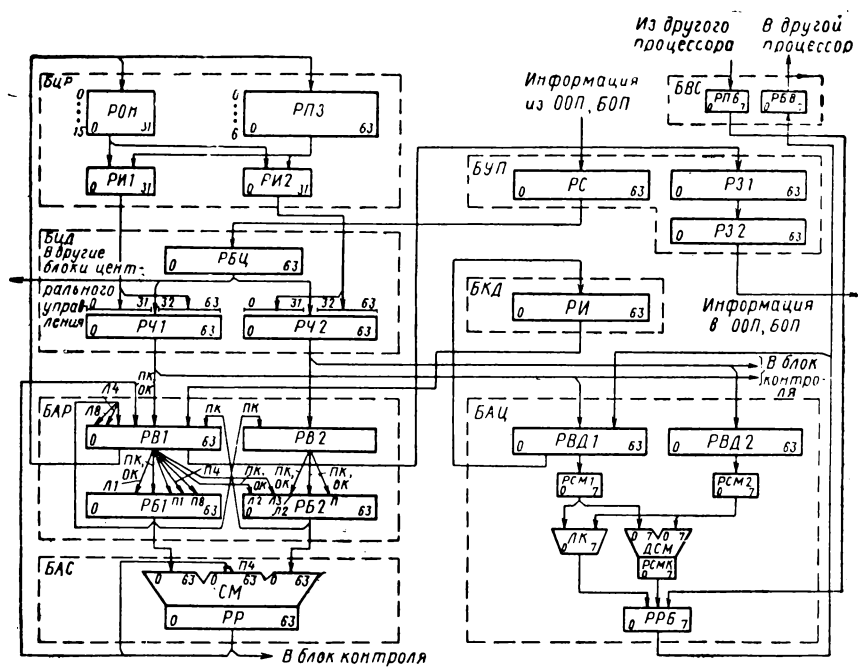


Рис. 6.3. Структурная схема АЛУ и его информационных связей с другими блоками процессора ЕС-2050

блока выбора данных БЦД процессора. Через этот блок проходит вся информация, которая должна обрабатываться в АЛУ. Из основной оперативной памяти в БЦД она поступает через регистр синхронизации РС блока управления оперативной памятью БУП в буферный регистр центрального управления РБЦ. В этот регистр поступают не только исходные операнды. Из ООП в него выбираются и участки программ, которые далее передаются в блоки центрального управления процессора.

Подготовка операндов к передаче в АЛУ производится в регистрах РЧ1 и РЧ2. Операнды, одновременно выбираемые из двух 32-разрядных регистров РОН, размещаются в РЧ1 или РЧ2 так, что в разрядах [0—31] РЧ находится операнд из РОН с четным адресом, а в разрядах [32—63] — из следующего с нечетным. Через РЧ1 или РЧ2 также передается операнд, принимаемый в РБЦ из ООП.

Операция в АЛУ начинается с приема в его входные регистры операндов из регистров РЧ1 и РЧ2. Для операций, выполняемых в БАС над операндами фиксированных форматов, информация принимается во входные регистры РВ1 и РВ2. Прием и размещение информации в этих регистрах зависит от кода операции и адресов операндов. Если операция должна выполняться в БАЦ, то опе-

ранды или их части из РЧ1 и РЧ2 поступают во входные регистры РВД1 и РВД2. Одновременно с этим из БЦД передаются: коды длин операндов и адреса начальных байт операндов в принятых на РВД1 и РВД2 64-разрядных словах.

Окончательная подготовка операндов перед передачей их в блок сумматора БАС происходит путем необходимых обменных пересылок между регистрами РВ и РБ и размещения в итоге на регистрах РБ1 и РБ2. Передачи из регистра в регистр могут происходить со сдвигами вправо П и влево Л на разные количества разрядов, как показано на схеме (П1, П4, П8, Л2, . . .).

Результат из регистра РР БАС посылается во входной регистр РВ1. Последний одновременно выполняет и роль выходного регистра АЛУ. С его выходов информация записывается либо в регистры РОН или РПЗ блока БЦР, либо в основную оперативную память через последовательно включенные регистры записываемой информации РЗ1 и РЗ2 блока управления оперативной памятью БУП.

В блоке БАЦ результаты выполнения операций над десятичными числами и над полями переменной длины, получаемые побайтно в регистре результатов байта РРБ, надлежащим образом размещаются в регистре РВД1.

Из этого регистра результат следует в регистр РВ1 блока БАР. Попадает он туда не непосредственно, а через регистр информации РИ блока контроля и диагностики БКД. При выполнении операции над операндами, не уместяющимися полностью в РВД1 и РВД2, они вызываются в БАЦ по частям, каждая из которых не превышает 8 байтов. В этом случае результат также вырабатывается по частям. Каждые очередные 8 байтов результата отсылаются в ООП через РИ и РВ1. Таким образом, регистр РВ1 является выходным и для блока БАЦ. Кроме того, регистр РРБ может непосредственно обмениваться байтами информации с блоком внешних связей БВС. Через этот блок происходит прием и передача информации при непосредственной связи с другим процессором. Посылаемые в этот процессор байты следуют из РРБ в выходной регистр байта РВВ блока БВС. Наоборот, принимаемые на обработку байты поступают на входы РРБ через приемный регистр байта РПБ.

С отсылкой результата в местную или основную оперативную память заканчивается выполнение операции в АЛУ.

Контроль правильности работы блоков БАР и БАС осуществляется путем одновременного выполнения тех же операций в блоке арифметического контроля БАК и сравнения результатов. Блок БАК, не показанный на рис. 6.3, по составу дублирует аппаратуру проверяемых блоков. Для получения контрольных результатов исходные операнды одновременно с подачей в РВ1 и РВ2 посылаются в аналогичные регистры блока АК. Проверка правильности работы оборудования блока АЦ основана на применении сквозного контроля по mod 2.

Рассмотренные связи АЛУ с местной и основной оперативной памятью и с другими блоками процессора через различные буферные регистры в блоках БЦД, БУП, БКД и БВС дают возможность осуществлять совмещение выполнения операций на разных уровнях. Это способствует увеличению допустимой плотности потока командной и обрабатываемой информации в процессоре, что соответственно повышает его производительность.

В процессоре ЕС-2050 приняты три уровня совмещения, одним из которых является выполнение действий над операндами в АЛУ и запись результата в местную или основную оперативную память. Одновременно с этим на других уровнях подготавливаются операнды на регистрах РЧ1 и РЧ2 для выполнения очередной команды и производится выбор в процессор следующей за ней команды.

Выводы о целесообразности объединения на одном уровне действий, производимых в АЛУ, и записи результата в память были получены в результате анализа процессов выполнения различных операций. Было установлено, что в среднем запись в основную оперативную память производится по командам загрузки, при исполнении которых не требуется работа АЛУ, и, наоборот, большинство арифметических операций кончается записью в местную память. Поэтому время выполнения команды на этом уровне в основном определяется временем выполнения операции в АЛУ (табл. 6.1).

Т а б л и ц а 6.1

Времена выполнения основных операций над операндами фиксированных форматов

Тип операции	Время выполнения операции (мкс) в модели		
	ЕС-1050	ЕС-1030	ЕС-1020
Короткие операции	0,56	5—11	20—30
Сложение — вычитание:			
с фиксированной запятой	0,56	9—11	20—30
с плавающей запятой	0,72—1,4	10—16	50—70
Умножение:			
с фиксированной запятой	2,32	32—38	220—350
с плавающей запятой	2,0—3,2	27—33	480
Деление:			
с фиксированной запятой	6,48	100—106	400
с плавающей запятой	5,76—12	46—52	400

На выполнение действий на всех уровнях отводится столько времени, сколько требуется. Это обеспечивается независимо работающими счетчиками тактов. Минимально возможный цикл счетчиков тактов всех уровней составляет шесть полутактов (полутакт равен 80 нс). Этот цикл определяется временем обращения к РОИ или РПЗ местной памяти.

При выполнении операции в блоке БАС запись в оперативную память происходит после выработки сигнала об окончании действий в АЛУ. В процессе же работы блока БАЦ за время исполнения одной команды формата SS может потребоваться несколько обращений к основной памяти. Операнд длиной в 256 байт может занимать в ООП 32 64-разрядные ячейки. Поэтому об окончании обработки очередных двойных слов, находящихся в РВД1 и РВД2 (или одного из них), указывается специальными сигналами. По этим сигналам готовая часть результата через РИ блока БКД отсылается в РВ1 и оттуда записывается в ООП. В то же время из ООП через блок БЦД принимаются на обработку продолжения операндов.

АЛУ процессора ЕС-2030. Для выборки из программы очередной команды и формирования исполнительных адресов операндов в процессоре ЕС-2050 имеются специальные блоки. Они работают параллельно не только с АЛУ, но и между собой. Например, это имеет место при выделении команды из участка программы, выбранного из ООП, и при действиях над кодами адресов предыдущей команды.

В процессоре ЕС-2030 нет такого параллельно работающего оборудования, а следовательно, и совмещения выполнения указанных действий. Выполнение очередной команды здесь складывается из последовательности следующих друг за другом этапов: выборки из ООП кода команды, формирования исполнительных адресов операндов и их выборки из МОП или ООП, выполнения операций и запоминания результата. При такой последовательности действий одно и то же оборудование процессора в разные интервалы времени используется для различных целей. Это в первую очередь относится и к узлам АЛУ. Вначале в его регистрах запоминаются части очередной выбранной команды, а затем с помощью сумматора формируются исполнительные адреса операндов. Эти адреса выдаются в регистр адреса РА основной оперативной памяти прямо из 8—31 разрядов сумматора АЛУ (рис. 6.4). Адреса регистров местной оперативной памяти МОП выдаются либо из регистра Р5, либо — из Р8 АЛУ. Непосредственно перед выполнением операции в регистрах АЛУ определенным образом размещаются регистры. После отсылки результата в оперативную память регистры АЛУ очищаются, и оно оказывается готовым к выполнению действий по приему и исполнению следующей команды.

Рассмотрим особенности построения и функционирования арифметико-логического устройства с точки зрения организации действий над операндами.

Как показано на рис. 6.4, регистровая часть АЛУ, в которой размещаются операнды или результаты действий над ними, в основном состоит из 32-разрядных регистров Р1—Р5, РА, РВ и 8-разрядных — РС, РD, Р7. Кроме этого, имеется еще ряд мало-разрядных регистров и множество триггеров, не показанных на рисунке. Они предназначены для запоминания различных

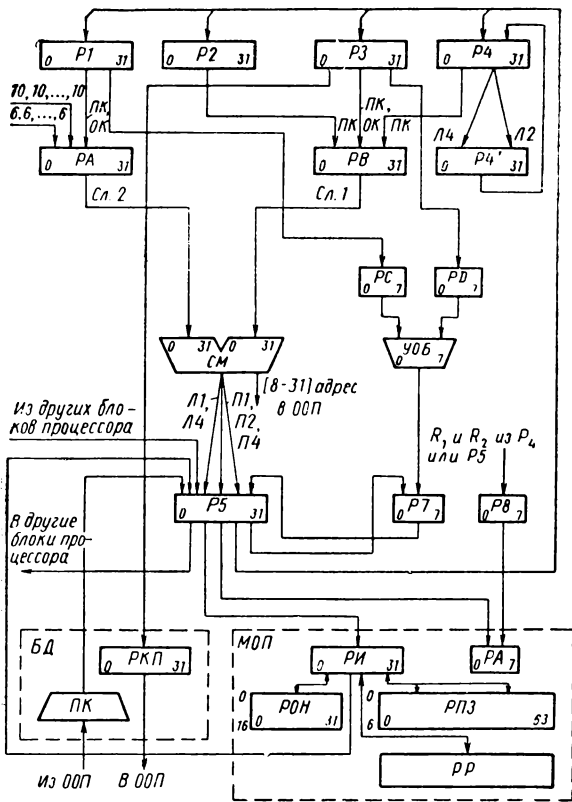


Рис. 6.4. Структурная схема АЛУ процессора ЕС-2030

кодов, сигналов, отражающих различные состояния, условия, результаты анализа преобразуемой информации, необходимые для правильного выполнения арифметических, логических и др. операций. Эти регистры и триггеры можно отнести к операционной части АЛУ, основу которой составляют сумматор СМ, узел логической обработки байтов УОБ и коммутаторы (группы вентилях), через которые осуществляется управление перемещением информации в АЛУ. Коммутаторы и цепи управления на структурной схеме также не показаны.

В регистрах P1 и P2 обычно размещается второй операнд, а в P3 и P4 — первый. Например, при сложении и вычитании в P1 или в P1 и P2 находятся соответственно 32- или 64-разрядное второе слагаемое или вычитаемое. Первое слагаемое при сложении и уменьшаемое при вычитании располагаются в регистрах P3 и P4. Короткие операнды (32-разрядные) занимают только регистр P3, а длинные (64-разрядные) — оба регистра P3 и P4. Регистры P4 и P4' образуют двухрядный сдвигающий регистр. В нем могут

производиться сдвиги влево на 4 разряда (Л4) и вправо на 2 разряда (П2). При умножении в Р4 помещается множитель.

Для непосредственного сложения в сумматоре СМ операнды или их части помещаются в регистры РА (второе слагаемое) и РВ (первое слагаемое). Эти регистры являются входными для комбинационного сумматора. Из Р1 и Р3 операнды в них пересылаются в прямом или обратном коде, а из Р2 и Р4 — только в прямом (на схеме отмечено соответственно ПК и ОК). В сумматоре могут складываться как двоичные, так и десятичные числа. В последнем случае для внесения избытков 6 и их изъятия из тетрад, из которых не было переноса при сложении, предусмотрены цепи занесения в РА кодов 6, 6, 6, . . . , 6₁₀ и 10, 10, 10, . . . , 10₁₀.

Результат сложения кодов в сумматоре запоминается в регистре результата Р5. Передача из СМ в Р5 может происходить как без сдвига, так и со сдвигами влево (Л1, Л4) или вправо (П1, П2, П4), как указано на схеме.

Для логических операций байты данных последовательно подаются во входные регистры РС и РD схемы УОБ. При сравнении порядков в операциях с плавающей запятой в РС и РD передаются характеристики чисел. Результаты обработки байт в УОБ запоминаются в восьмиразрядном регистре результата Р7. Из него они передаются в регистр Р5, последовательно заполняя последний.

На обработку в АЛУ информация 32-разрядными словами принимается через регистр Р5. Операнды, указываемые адресами R₁ и R₂, выбираются из РОИ или РПЗ и в регистр Р5 заносятся через регистр информации РИ местной оперативной памяти МОП. Из основной оперативной памяти ООП (из ее регистра информации РИ ООП) операнды в Р5 поступают через приемный коммутатор ПК процессорного блока диагностики БД. Поступившие в Р5 операнды затем, в соответствии с алгоритмом выполнения операции, передаются в регистры Р1—Р4.

Результаты операций 32-разрядными словами выдаются из АЛУ через регистры Р3, Р5. Из регистра Р3 производится запись информации в ООП. В РИ ООП записываемые слова передаются через регистр-коммутатор процессора РКП, расположенный в блоке диагностики. Запись в МОП происходит через регистр Р5, выходы которого непосредственно связаны с РИ МОП.

Кроме указанных цепей, узлы АЛУ связаны с другими блоками процессора, в том числе с блоком контроля. В модели ЕС-1030 применен сквозной контроль по mod 2.

Во временной диаграмме работы процессора на собственно суммирование отводится один машинный такт, равный 600 нс. Однако из-за последовательного характера действий, производимых при исполнении каждой команды, время, затрачиваемое на выполнение операций, в модели ЕС-1030 существенно больше, чем в модели ЕС-1050. В табл. 6.1 указаны средние времена выполнения операций над операндами фиксированных форматов (с учетом времени обращения к оперативным запоминающим устройствам).

АЛУ процессора ЕС-2020. Стремление уменьшить объем электронного оборудования процессора ЕС-2020 при побайтовой переработке информации привело к выбору регистровой структуры вычислительно-управляющего устройства. В этом устройстве большинство одних и тех же узлов используется как для выполнения арифметических и логических действий над преобразуемыми данными, так и над командами (в основном над их адресными частями).

Регистровая структура основана на применении ряда регистров, связанных друг с другом системой информационных магистралей через общий универсальный арифметико-логический блок АЛБ комбинационного типа, как показано на рис. 6.5. В цикле каждой операции управление выполнением в разные моменты времени различных действий на одном и том же оборудовании осуществляется микропрограммным способом.

Очередные байты данных, предназначенные для преобразования в комбинационных схемах АЛБ, заносятся в регистры РА и РВ. Эти регистры являются входными для АЛБ. Информация в них может поступать из всех регистров процессора, в которых в ходе выполнения каждой операции хранятся байты данных, команд и другая служебная информация. Регистры процессора, образующие блок регистров БР, в основном однобайтовые и двухбайтовые. Следует отметить, что это деление весьма условно, так как многие из них могут принимать и выдавать информацию не только побайтно, но и по несколько и даже по одному разряду. Кроме восьми информационных разрядов, на каждый байт команды или данных предусмотрен еще один (9-й) разряд, предназначенный для хранения контрольного кода байта. В модели ЕС-1020 применен сквозной контроль по mod 2 (на четность).

На структурной схеме изображены основные части адресных регистров (РФ, РЕ, РР, РИ, РТ и РУ) и процессорные регистры общего назначения (РЛ и РД), представляющие собой половину регистров блока БР.

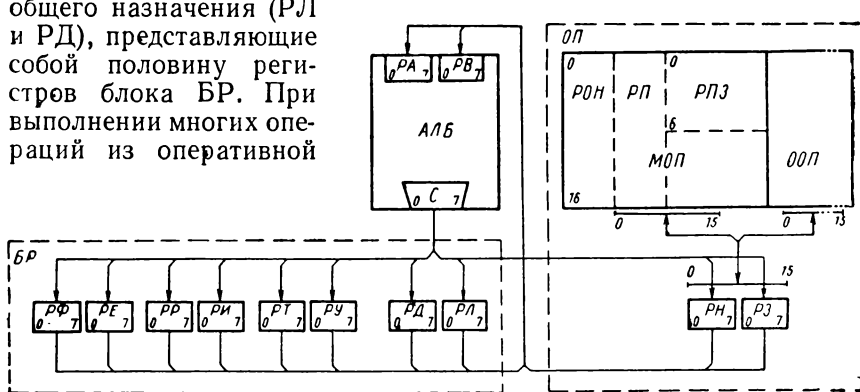


Рис. 6.5. Структурная схема информационных связей АЛБ с регистрами процессора и с оперативной памятью в модели ЕС-1020

памяти в РЛ и РД последовательно вводятся очередные байты данных. Например, в операциях типа сложения в них поочередно поступают пары байт первого слагаемого. Для ускорения выполнения таких операций, как умножение, деление, желательно, чтобы большие части операндов одновременно располагались в регистрах процессора. Поэтому для размещения операндов в таких операциях, кроме РЛ и РД, дополнительно используются перечисленные выше адресные регистры. Например, перед умножением операндов с фиксированной запятой в регистрах РД, РФ, РЕ и РЛ размещается 32-разрядный множитель, при делении в регистрах РР, РИ, РТ и РЛ — делитель. Перед приемом операндов эти регистры очищаются. Информация, которая может понадобиться в дальнейших действиях по выполнению операции, отсылается в определенные ячейки МОП, образующие рабочую память РП.

Все регистры блока БР могут соединяться друг с другом через АЛБ. Через соответствующие коммутаторы все они могут передавать информацию либо в оба регистра РА и РВ, либо в один из них. С помощью выходного коммутатора С блока АЛБ информация может быть введена в любой регистр блока БР. Для того чтобы можно было передавать информацию из регистра в регистр без изменений, в АЛБ предусмотрена возможность выполнения микроопераций транзитной пересылки (А ТРАНЗИТ, В ТРАНЗИТ).

Универсальный АЛБ предназначен для выполнения операций двоичной и десятичной арифметики, а также логических операций над байтами операндов. Кроме того, как следует из рассмотренного выше, АЛБ производит операции над адресами и осуществляет пересылку информации между регистрами процессора.

АЛБ связан с остальными блоками процессора, в том числе и со всеми регистрами блока БР, системой трех 9-разрядных шин — двух входных на входах РА и РВ и выходной шины на выходе коммутатора С. За один машинный такт длительностью 1 мкс в АЛБ выполняется одна микрооперация (сложение двух байт, сдвиг байта на разряд вправо или влево, транзитная передача байта в регистр и т. п.). Результат выполнения микрооперации заносится в один из регистров процессора.

Через АЛБ осуществляется информационная связь регистров процессора с оперативной памятью машины. В модели ЕС-1020 оперативная память представляет собой ферритовое ОЗУ емкостью 64—256К байт, в котором выделены области местной (локальной) оперативной памяти МОП емкостью 256 байт, памяти мультиплексного канала ПМК (на рис. 6.5 не показано), на которую может отводиться 768—1536 байт. Остальная емкость ОЗУ является основной оперативной памятью ООП. На рис. 6.5 условно показано разделение области МОП на подобласти регистров общего назначения РОН, регистров для операндов с плавающей запятой РПЗ и рабочей памяти РП, где в процессе выполнения операций, например, могут храниться частные произведения, часть 64-разрядного делимого и т. п. Остальная часть МОП предназначена для запоминания раз-

личной служебной информации: слова состояния программы ССП при прерываниях, частей кодов выполняемых команд, на месте которых в ходе выполнения операций в регистрах процессора размещаются операнды, и т. п. Область памяти мультиплексного канала на схеме не показана.

Прием в ОП и выдачи из нее информации осуществляется полусловами. При обращении к любой из трех областей МОП, ПМК или ООП передача информации осуществляется через двухбайтовый информационный регистр, половины которого обозначены РН и РЗ. Выбираемая из ОП информация вначале поступает в РН и РЗ, а затем побайтно через АЛБ в нужные регистры блока БР. Информация, подлежащая записи в ОП, побайтно заносится в регистры РН и РЗ с выходного коммутатора С АЛБ.

В процессе выполнения ряда операций в информационном регистре РН, РЗ могут храниться части одного из операндов, откуда они непосредственно подаются на обработку в АЛБ. Например, если при выполнении операции сложения с фиксированной запятой по команде формата RR в регистрах РЛ и РД находятся два очередных байта второго операнда, то в регистрах РН и РЗ — соответствующая им пара байт первого операнда. Очередная часть результата побайтового сложения частей операндов через информационный регистр записывается в соответствующую пару байт R_1 РОН.

Побайтовое преобразование информации в АЛБ, последовательное перемещение ее по частям из областей ОП в БР и обратно требует большего времени на выполнение операций, предусмотренных системой команд ЕС ЭВМ. Для сравнения времени выполнения операций над операндами фиксированных форматов имеются в табл. 6.1.

6.3. ОПЕРАЦИОННАЯ ЧАСТЬ АРИФМЕТИКО-ЛОГИЧЕСКОГО УСТРОЙСТВА ПРОЦЕССОРА ЕС-2050

Двоичный сумматор в блоке БАС, позволяющий одновременно производить сложение трех 64-разрядных слагаемых, выполнен двухступенчатым, как показано на рис. 6.6. Обе ступени (I ст. и II ст.) — комбинационные. В первой из них формируется двухрядный код, состоящий из кода S поразрядных полусумм (сумм по mod 2) и кода Q поразрядных переносов, получаемых в результате сложения трех двоичных чисел. С выходов этой ступени коды S и Q поступают в 64-разрядные регистры сумм РС и переносов РП.

Во второй ступени (СМ) сумматора осуществляется полное приведение переносов. Полученный таким образом окончательный однорядный код результата запоминается в регистре результата РР. Кроме блока контроля, из этого регистра он может быть послан либо в РВ1 (см. рис. 6.3), либо снова на вход сумматора

прямым кодом (ПК), обратным кодом (ОК) или прямым кодом со сдвигом на 4 разряда вправо (П4).

В сумматоре второй ступени полное приведение переносов выполняется по каскадно-параллельной схеме. Переносы как внутри групп (тетрады и байты), так и между группами (межбайтовые) приводятся параллельно по всем разрядам. Для осуществления этого в первой ступени специальными схемами СГП вырабатываются групповые переносы из тетрад и байтов, а также тетрадные и байтовые разрешения переносов. Эти переносы Q_6 и Q_7 и разрешения A_6 и A_7 в виде 8-разрядных кодов заносятся в регистры байтовых и тетрадных переносов РБП и РТП и регистры разрешения байтовых и тетрадных переносов РРБП и РРТП. Сигналы с выходов данных регистров используются для образования параллельных переносов в каждый разряд сумматора СМ.

Двухступенчатая схема сумматора с тремя входами и с промежуточным запоминанием поразрядных полусумм и переносов применена в модели ЕС-1050 для ускорения выполнения операций умножения и деления.

Три входа дают возможность на каждом шаге умножения суммировать сдвинутую на четыре разряда вправо текущую сумму частных произведений сразу с двумя очередными кратными множимого. Это позволило с помощью одного сумматора реализовать ускоренное умножение с одновременной расшифровкой четырех разрядов множителя. Промежуточное запоминание полусумм и переносов исключило необходимость на каждом шаге деления делать полное приведение переносов. Для определения знака очередного остатка переносы полностью приводятся только в старших пяти разрядах сумматора, включая два знаковых (см. п. 3.2). Это приведение осуществляется одновременно с получением двухрядного кода остатка. После анализа его знака частное либо прибавляется непосредственно к удвоенному двухрядному коду остатка, либо вычитается из него. Для этого содержимое РС и РП с соответствующими сдвигами влево может передаваться в РБ1 и РБ2 блока БАР. Делитель все время находится в РР, откуда в прямом или обратном коде поступает на третий вход сумматора. Окончательное полное приведение переносов в коде остатка производится только в конце деления. Таким образом, экономится время на исключении необходимости вычисления однорядного кода остатка на каждом шаге деления.

Формирование двухрядного кода S, Q происходит в два этапа. При сложении трех 64-разрядных двоичных чисел $X = x_0x_1 \dots \dots x_i \dots \dots x_{63}$, $Y = y_0y_1 \dots \dots y_i \dots \dots y_{63}$ и $Z = z_0z_1 \dots \dots z_i \dots \dots z_{63}$ вначале в каждом i -м разряде получаются промежуточные полусумма s'_i и перенос q'_i согласно (3.17) и (3.18). Они образуют 64-разрядные коды S' и Q' . Эти промежуточные коды вырабатываются входным трехвходным полусумматором ПСМЗ (рис. 6.6). Затем коды S' и Q' поступают в двухвходный полусумматор ПСМ2, где согласно (3.19) и (3.20) формируются значения

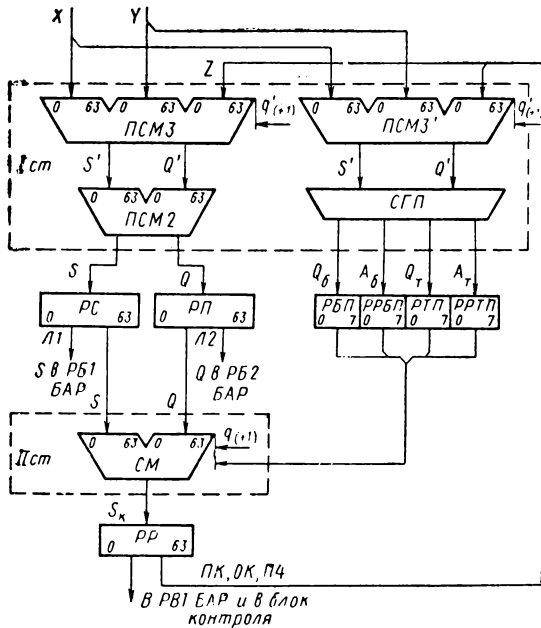


Рис. 6.6. Структурная схема БАС

s_i и q_i окончательного двухрядного кода, состоящего из 64-разрядных кодов S и Q .

Необходимость двухэтапного формирования кодов S и Q объясняется тем, что при сложении трех слагаемых в i -м разряде с учетом переноса из $(i + 1)$ -го разряда могут складываться четыре единицы. В этом случае требуется передача в $(i - 1)$ -й разряд сразу двух единиц переноса.

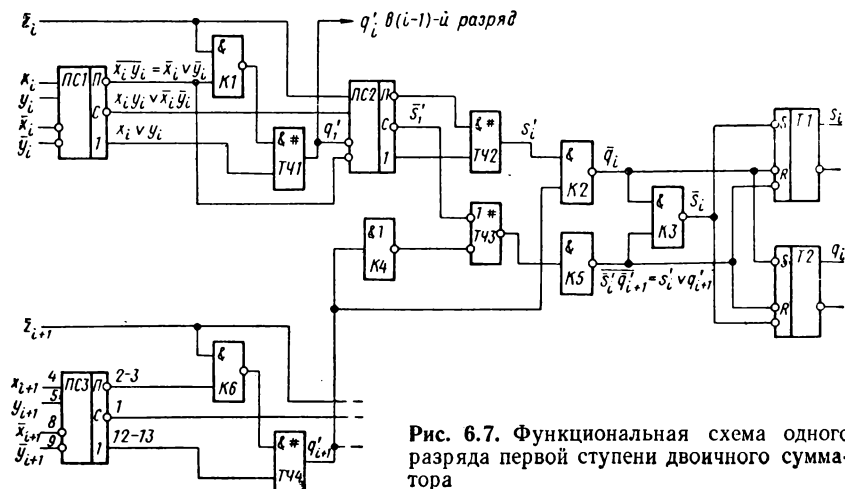


Рис. 6.7. Функциональная схема одного разряда первой ступени двоичного сумматора

На рис. 6.7 представлена функциональная схема i -го разряда первой ступени двоичного сумматора. В ЕС-2050 сумматор реализуется на полупроводниковых интегральных элементах 137 серии. В схемах на элементах такого типа коду l ставится в соответствие низкий уровень потенциала. Поэтому на рис. 6.7 единичным значениям $x_i, y_i, z_i, s'_i, q'_i, s_i$ и q_i соответствуют сигналы низкого уровня.

С помощью полусумматоров ПС1 и ПС3 и конъюнкторов К1 и К6 формируются переносы q'_i и q'_{i+1} , а также инверсии сумм по под $2 x_i y_i \vee x_i \bar{y}_i$ и $x_{i+1} y_{i+1} \vee x_{i+1} \bar{y}_{i+1}$. Перенос q'_i (q'_{i+1}) получается на объединенных выходах конъюнктора К1 и полусумматора ПС1 (К6 и ПС3). В полусумматорах используются прямые дизъюнктивные выходы l , на которых формируются дизъюнкции $x_i \vee y_i$ и $x_{i+1} \vee y_{i+1}$. На выходе К1 (К6) сигналы соответствуют логическому соотношению $\overline{x_i y_i z_i} = (\overline{x_i \vee y_i}) \bar{z}_i (\overline{x_{i+1} \vee y_{i+1}}) \cdot z_{i+1}$. Соединение накоротко (в точку) выходов l ПС1 и К1 дает возможность получить перенос q'_i в виде

$$q'_i = \overline{(x_i \vee y_i)} \bar{z}_i (x_i \vee y_i) = x_i y_i \vee x_i z_i \vee y_i z_i. \quad (6.1)$$

Аналогично вырабатывается и q'_{i+1} .

Инверсия \bar{s}'_i формируется непосредственно на выходе С полусумматора ПС2, т. е.

$$\bar{s}'_i = (x_i y_i \vee \bar{x}_i \bar{y}_i) \bar{z}_i \vee \overline{(x_i \vee y_i)} \bar{z}_i (x_i \vee y_i) (\bar{x}_i \vee \bar{y}_i), \dots, \quad (6.2)$$

а s'_i — путем соединения в точку его выходов П и l :

$$s'_i = \overline{(x_i y_i \vee \bar{x}_i \bar{y}_i \vee \bar{z}_i)} (\bar{x}_i \vee \bar{y}_i \vee \overline{(x_i \vee y_i)} \bar{z}_i (x_i \vee y_i)). \quad (6.3)$$

Элементы функциональной схемы рис. 6.7, с помощью которых вырабатываются q'_i, s'_i и \bar{s}'_i , относятся к ПСМ3 (см. рис. 6.6). Схему полусумматора ПСМ2 в каждом разряде составляют элементы, с помощью которых образуются s_i и q_i . Получаемое на выходе К3 значение s_i соответствует выражению

$$\bar{s}_i = \overline{s'_i q'_{i+1}} \overline{s'_i q'_{i+1}}, \quad (6.4)$$

а на выходе К2

$$\bar{q}_i = \overline{s'_i q'_{i+1}}. \quad (6.5)$$

Нетрудно убедиться, что эти логические функции представляют собой инверсии выражений (3.19) и (3.20).

Сигналами, соответствующими \bar{s}_i и q_i , осуществляется переключение i -х триггеров (Т1 и Т2) типа RS регистров сумм РС и переносов РП. Триггеры переключаются высокими уровнями потенциала. Формирование в рассмотренной комбинационной схеме и занесение в триггеры значений s_i и q_i , или в целом S и Q, осу-

шестьвается за один полутакт синхронизации длительностью 80 нс. В этом же полутакте вырабатываются и запоминаются в соответствующих регистрах групповые переносы и разрешения переносов. Их выработка также производится в два этапа. Вначале формируются коды полусумм S' и переносов Q' . Для этой цели оказалось удобнее использовать дополнительный полусумматор на три входа ПСМЗ' (см. рис. 6.6), в общих чертах повторяющий ПСМЗ, но имеющий несколько большую скорость работы. По полученным для всех разрядов значениям s'_i и q'_i для каждой j -й 8-разрядной группы сумматора вырабатываются: групповой перенос из ее второй тетрады $q_{T_{2j}}$, тетрадное разрешение переноса $a_{T_{2j}}$, групповой перенос из этого j -го байта q_{6j} и байтовое разрешение переноса a_{6j} (рис. 6.8). Сигналы, соответствующие этим значениям, формируются в схеме групповых переносов СГП согласно формулам:

$$q_{T_{2j}} = s'_{i_1} q'_{i_5} \vee (s'_{i_4} \vee q'_{i_5}) s'_{i_5} q'_{i_6} \vee (s'_{i_4} \vee q'_{i_5}) (s'_{i_5} \vee q'_{i_6}) s'_{i_6} q'_{i_7} \vee \vee (s'_{i_4} \vee q'_{i_5}) (s'_{i_5} \vee q'_{i_6}) (s'_{i_6} \vee q'_{i_7}) s'_{i_7} q'_{i_{(j+1)}}, 0; \quad (6.6)$$

$$a_{T_{2j}} = (s'_{i_4} \vee q'_{i_5}) (s'_{i_5} \vee q'_{i_6}) ((s'_{i_6} \vee q'_{i_7}) (s'_{i_7} \vee q'_{i_{(j+1)}}, 0); \quad (6.7)$$

$$q_{6j} = s'_{j_0} q'_{j_1} \vee (s'_{j_0} \vee q'_{j_1}) s'_{j_1} q'_{j_2} \vee (s'_{j_0} \vee q'_{j_1}) (s'_{j_1} \vee q'_{j_2}) (s'_{j_2} q'_{j_3} \vee \vee (s'_{j_0} \vee q'_{j_1}) (s'_{j_1} \vee q'_{j_2}) (s'_{j_2} \vee q'_{j_3}) s'_{j_3} q'_{j_4} \vee \vee (s'_{j_0} \vee q'_{j_1}) (s'_{j_1} \vee q'_{j_2}) (s'_{j_2} \vee q'_{j_3}) (s'_{j_3} \vee q'_{j_4}) q_{T_{2j}}; \quad (6.8)$$

$$a_{6j} = (s'_{j_0} \vee q'_{j_1}) (s'_{j_1} \vee q'_{j_2}) (s'_{j_2} \vee q'_{j_3}) (s'_{j_3} \vee q'_{j_4}) a_{T_{2j}}, \quad (6.9)$$

где цифровой индекс, следующий за индексом j , указывает номер разряда в j -м байте; $j = 0, 1, 2, \dots, 7$.

Величина $q'_{(j+1)}$ в 7-м байте (рис. 6.8) представляет собой дополнительную единицу, прибавляемую к значению младшего разряда. Подсуммирование этой единицы необходимо, когда для одного из слагаемых, поступающих на вход сумматора, приходится брать дополнение.

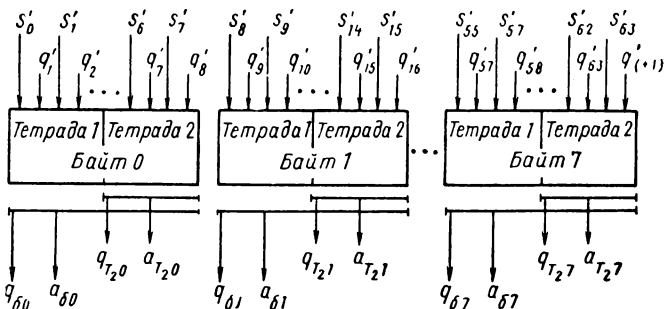


Рис. 6.8. Распределение входных и выходных сигналов по байтам сумматора в схеме формирования групповых переносов СГП

Значения тетрадных и байтовых переносов, а также разрешения тетрадных и байтовых переносов, образующие 8-разрядные коды соответственно $Q_T = q_{T,0}q_{T,1} \dots q_{T,7}$, $Q_6 = q_{6,0}q_{6,1} \dots q_{6,7}$, $A_T = a_{T,0}a_{T,1} \dots a_{T,7}$ и $A_6 = a_{6,0}a_{6,1} \dots a_{6,7}$, заносятся в соответствующие регистры (см. рис. 6.6). При определении окончательной суммы в сумматоре СМ II ступени двоичного сумматора БАС эти значения совместно с полусуммой S и переносами Q используются для формирования по каскадно-параллельной схеме входных переносов $q_{вх. i}$ в каждый разряд сумматора, где $i = 0, 1, 2, \dots, 63$. По полученным входным переносам $q_{вх. i}$ и значениям полусумм s_i полное приведение переносов осуществляется путем суммирования по модулю $2s_i$ и $q_{вх. i}$, т. е.

$$s_{ik} = s_i \oplus q_{вх. i} \quad (6.10)$$

Значения s_{ik} , полученные для всех разрядов, образуют код окончательной суммы S_k , который заносится в регистр результата РР (см. рис. 6.6).

По каскадно-параллельной схеме может достичь, например, нулевого разряда сумматора СМ либо любой перенос q_1, q_2 или q_3 , возникший в первой тетраде нулевого байта (рис. 6.8), либо групповой тетрадный перенос $q_{T,0}$ из второй тетрады этого же байта, либо входной перенос нулевого байта $q_{вх. 60}$, переданный в данный байт из других групп, т. е.

$$q_{вх. 0} = q_1 \vee s_1 q_2 \vee s_1 s_2 q_3 \vee s_1 s_2 s_3 q_{T,0} \vee s_1 s_2 s_3 a_{T,0} q_{вх. 6, 0} \quad (6.11)$$

где

$$q_{вх. 6, 0} = q_{61} \vee a_{61} q_{62} \vee a_{61} a_{62} q_{63} \vee \dots \vee a_{61} a_{62} a_{63} a_{64} a_{65} a_{66} a_{67} \vee \vee a_{61} a_{62} a_{63} a_{64} a_{65} a_{66} a_{67} q_{(+1)}; \quad (6.12)$$

$q_{(+1)}$ — дополнительная единица в 63-й разряд, которую можно рассматривать как входной перенос 7-го байта.

Функциональная схема формирования окончательного значения s_{0k} в нулевом разряде суммы S_k приведена на рис. 6.9. Левая часть этой схемы с полусумматором ПС предназначена для образования входного переноса нулевого байта $q_{вх. 60}$, правая — для получения входного переноса нулевого разряда $q_{вх. 0}$ и его суммирования по mod 2 с s_0 согласно (6.10). Это суммирование выполняется в сумматоре М2. Результат заносится в нулевой разряд регистра результата РР [0], в триггер типа RS. Управляет занесением управляющий сигнал УС.

Полное приведение переносов в каскадно-параллельной схеме СМ II ступени осуществляется за полутакт синхронизации, равный 80 нс. Таким образом, общее время суммирования в двухступенчатой схеме БАС осуществляется за два полутакта, т. е. за 160 нс.

В блоке десятичной арифметики БАЦ основу операционной части составляет десятичный сумматор ДСМ (рис. 6.10). Это — 8-разрядный комбинационный сумматор, позволяющий склады-

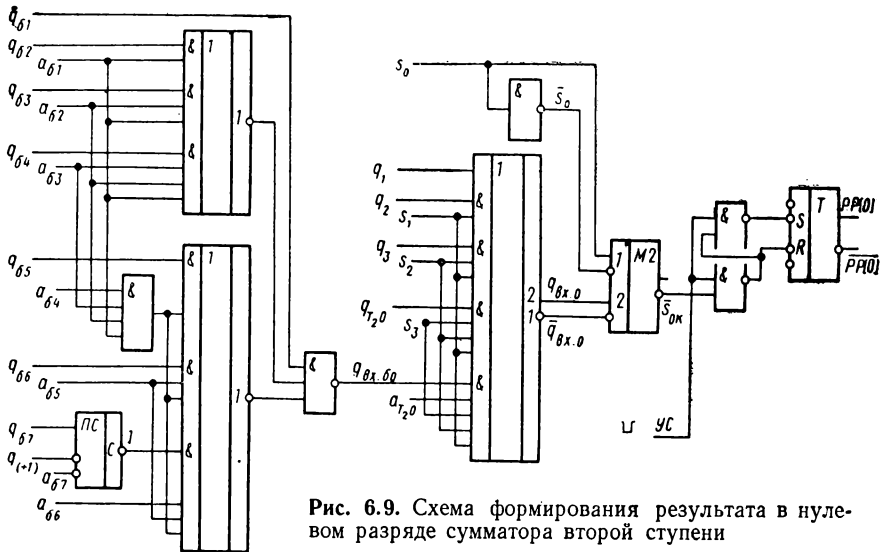


Рис. 6.9. Схема формирования результата в нулевом разряде сумматора второй ступени

вать как положительные, так и отрицательные десятичные числа, представленные в двоично-десятичной форме в коде с избытком 6. Отрицательные числа при сложении и вычитании представляются

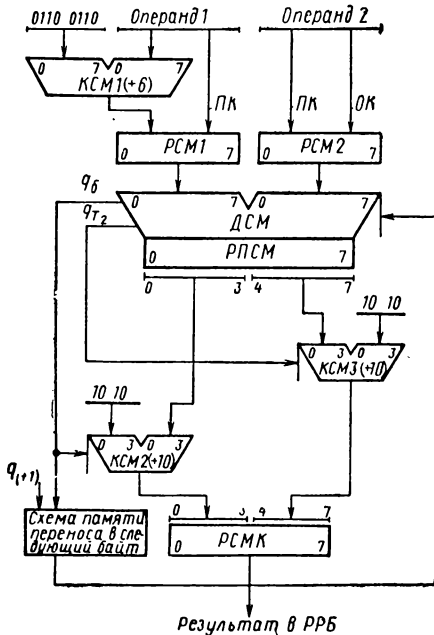


Рис. 6.10. Структурная схема десятичного сумматора

в дополнительном коде (см. п. 2.5). При сложении кодов чисел разных знаков пары тетрад второго операнда подаются на вход ДСМ в обратном коде. Для образования дополнительного кода второго операнда на вход переноса 7-го разряда сумматора при сложении младших тетрад операндов подается сигнал $q_{(+1)} = 1$. Эта дополнительная единица поступает из схемы памяти (триггера) переноса в следующий байт, в которую она предварительно заносится по сигналу из блока управления.

Одновременно из РВД1 и РВД2 (см. рис. 6.3) через РСМ1 и РСМ2 на входы ДСМ подаются два очередных байта, в каждом из которых содержатся две тетрады, представляющие собой две десятичные цифры. Если складываются числа одинако-

вых знаков, то очередные байты заносятся в РСМ1 и РСМ2 (рис. 6.10) в прямом коде. При этом байт первого числа из РВД1 следует через первый корректирующий сумматор КСМ1, где к каждой его тетраде прибавляется избыток +6 (0110). Когда знаки слагаемых разные, байт первого операнда поступает в РСМ1 в прямом коде, минуя КСМ1, а байт второго операнда — в обратном коде. Обратный код байта получается его поразрядным инвертированием. При этом автоматически в его тетрады вводятся избытки +6, так как для цифр от 0 до 9 берутся дополнения до 15 (1111).

Сложение двоично-десятичных кодов, один из которых в каждой тетраде имеет избыток 6, может быть выполнено обычным двоичным сумматором. Для того чтобы можно было определить необходимость последующей коррекции результата и провести ее, надо лишь запомнить значения межтетрадного q_T , и межбайтового q_6 переносов.

Таким сумматором является схема ДСМ на рис. 6.10. Результат сложения в ней двух пар тетрад заносится вначале в регистр промежуточной суммы РПСМ. Переносы q_6 и q_T , для нужд коррекции запоминаются в специальных триггерах, не показанных на схеме. Если они равны нулю, то в корректирующих сумматорах КСМ2 и КСМ3 к содержимым разрядов [0—3] и [4—7] РПСМ прибавляется код 1010, являющийся дополнением 6 до 16. Таким образом, из каждой тетрады изымается избыток +6 и скорректированная сумма запоминается в регистре десятичной коррекции РСМК. Указанный избыток в процессе сложения автоматически удаляется из тетрады, если возникает перенос из ее старшего разряда. Поэтому, если равен единице q_T , или q_6 , в сумматоре КСМ3 или КСМ2 соответственно коррекция не производится.

Двухвходовой десятичный сумматор ДСМ, i -й разряд которого представлен на рис. 6.11, имеет цепи параллельного распространения переносов. Окончательное значение суммы в каждом разряде с помощью полусумматора и входной логики триггера типа D формируется как

$$s_{ik} = (s'_i \bar{q}_{i+1} \vee \bar{s}'_i q_{i+1}) c_i, \quad (6.13)$$

где $\bar{s}'_i = x_i y_i \vee \bar{x}_i \bar{y}_i$ берется непосредственно с выхода с полусумматора ПС1, а $s'_i = (x_i \vee y_i) \bar{x}_i \bar{y}_i = x_i \bar{y}_i \vee \bar{x}_i y_i$ — получается путем

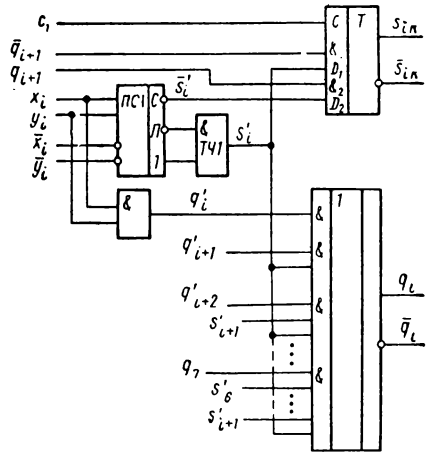


Рис. 6.11. Функциональная схема одного разряда ДСМ

объединения в точку (псевдоэлемент ТЧ1) выходов Π и I того же полусумматора; q_{i+1} — входной перенос в i -й разряд, формируемый из предыдущих разрядов параллельными цепями; c_1 — синхронизирующий сигнал.

Структура параллельных цепей переноса показана на примере формирования окончательного значения переноса в i -м разряде. Как следует из схемы

$$q_i = q'_i \vee s'_i q'_{i+1} \vee s'_i s'_{i+1} q'_{i+2} \vee \dots \vee s'_i s'_{i+1} \dots s'_6 q_7, \quad (6.14)$$

где

$$q'_i = x_i y_i;$$

$$q_7 = x_7 y_7 \vee s'_7 (q_6 \vee q_{(7+1)}).$$

Для выполнения логических операций таких, как *И*, *ИЛИ*, *исключающее ИЛИ* (отрицание равнозначности), в блоке БАЦ имеется логический коммутатор ЛК (см. рис. 6.3). Эта схема также рассчитана на побайтовую переработку информации. В каждом из восьми разрядов она содержит логические схемы, реализующие логические операции:

$$P_{1i} = x_i y_i;$$

$$P_{2i} = x_i \vee y_i;$$

$$P_{3i} = x_i \bar{y}_i \vee \bar{x}_i y_i. \quad (6.15)$$

Результаты логических действий над одноименными разрядами очередной пары байтов операндов заносятся непосредственно в регистр результата обработки байтов РРБ.

6.4. ОПЕРАЦИОННАЯ ЧАСТЬ АРИФМЕТИКО-ЛОГИЧЕСКОГО УСТРОЙСТВА ПРОЦЕССОРА ЕС-2030

Сложение двоичных и десятичных чисел (см. рис. 6.4) в процессе выполнения арифметических операций происходит в 32-разрядном сумматоре комбинационного типа с групповой структурой (рис. 6.12, *а*). Разряды с 1-го по 31-й распределены по восьми группам, каждая из которых, кроме первой, имеет по 4 разряда. Первая группа — 3-разрядная. Нулевой (знаковый) разряд не включен в эту группу. Кроме основных 32 разрядов, в сумматоре АЛУ имеются 4 дополнительных разряда (на схеме не показаны), предназначенных для действий над цифрами, выходящими за пределы разрядной сетки сумматора в сторону старших разрядов. Это бывает необходимо при выполнении десятичных операций

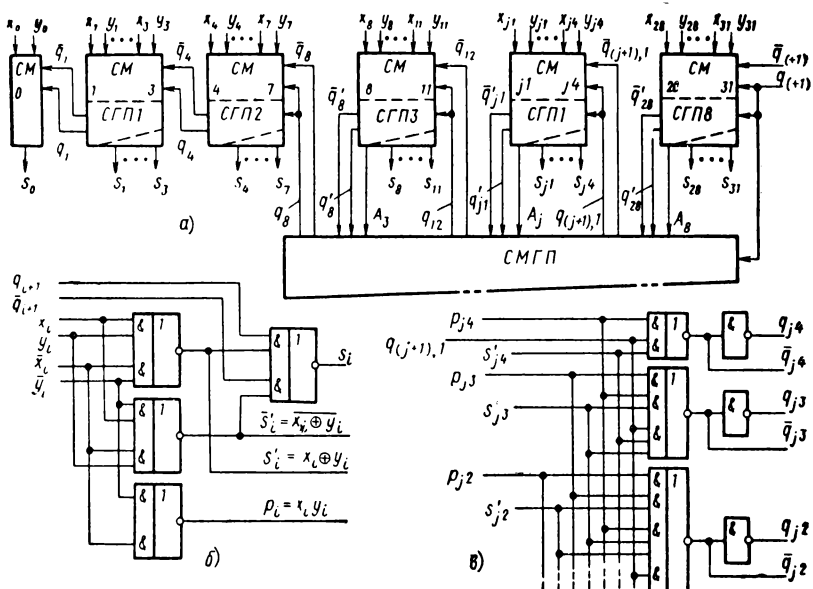


Рис. 6.12. Сумматор процессора ЕС-2030:

а — структурная схема; б — функциональная схема одного разряда сумматора;
 в — часть схемы групповых переносов

и операций над операндами с плавающей запятой. Перенос в дополнительные разряды происходит из нулевого разряда основного сумматора.

Каждая группа сумматора состоит из поразрядных схем формирования полусумм и цепей параллельного образования и распространения переносов. Функциональная схема одноразрядного сумматора, предназначенного для сложения i -х разрядов слагаемых с учетом переноса из $(i + 1)$ -го разряда, изображена на рис. 6.12, б. Как и все электронные схемы ЕС-1030, она реализуется на монолитных интегральных схемах типа ТТЛ серии 155.

В I ступени элементами И—ИЛИ—НЕ формируются сумма по mod 2 x_i и y_i и ее отрицание, а также конъюнкция $x_i y_i$, во II ступени — окончательное значение суммы s_i в i -м разряде сумматора. В целом s_i получается в виде следующего выражения:

$$s_i = \overline{x_i y_i} \vee \overline{x_i y_i} q_{i+1} \vee x_i y_i \vee x_i y_i q_{i+1}, \quad (6.16)$$

где q_{i+1} — перенос из $i + 1$ -го разряда сумматора;
 $i = 0, 1, \dots, 31$.

Полусумма $s'_i = x_i \oplus y_i$ и конъюнкция $p_i = x_i y_i$ используются в схеме групповых переносов СГП. Часть такой схемы показана

*.

на рис. 6.12, в. Цепи параллельного образования и распространения переносов строятся в СГП согласно логическим формулам

$$q_{j4} = \overline{p_{j4} \vee s'_{j4} q_{(j+1), 1}}; \quad (6.17)$$

$$q_{j3} = \overline{p_{j3} \vee s'_{j3} p_{j4} \vee s'_{j3} s'_{j4} q_{(j+1), 1}}; \quad (6.18)$$

$$q_{j2} = \overline{p_{j2} \vee s'_{j2} p_{j3} \vee s'_{j2} s'_{j3} p_{j4} \vee s'_{j2} s'_{j3} s'_{j4} q_{(j+1), 1}}; \quad (6.19)$$

$$q_{j1} = \overline{p_{j1} \vee s'_{j1} p_{j2} \vee s'_{j1} s'_{j2} p_{j3} \vee s'_{j1} s'_{j2} s'_{j3} p_{j4}}; \quad (6.20)$$

где $q_{j2} - q_{j4}$ — полные входящие переносы во 2—4-й разряды j -й группы сумматора;

$q_{(j+1), 1}$ — входящий перенос в j -ю группу из $(j+1)$ -й;

q_{j1} — перенос из j -й группы, формируемой в СГП без учета входящего переноса в группу;
 $j = 1, 2, \dots, 8.$

Группы 3—8 объединены схемой межгруппового переноса СМГП (рис. 6.12, а), представляющей собой совокупность цепей параллельных переносов в старшие группы. С помощью этой схемы входящий перенос, например, во 2-ю группу формируется аналогично зависимости

$$q_{\text{вх. 2}} = q'_8 \vee A_3 q'_{12} \vee A_3 A_4 q'_{16} \vee A_3 A_4 A_5 q'_{20} \vee A_3 A_4 A_5 A_6 q'_{24} \vee \\ \vee A_3 A_4 A_5 A_6 A_7 q'_{28} \vee A_3 A_4 A_5 A_6 A_7 A_8 q_{(j+1)}, \quad (6.21)$$

где $A_j = s'_{j1} s'_{j2} s'_{j3} s'_{j4}$;

$q'_8 - q'_{28}$ получаются так же, как и (q'_{j1}) в (6.20).

Две старшие группы (1 и 2) сумматора соединены между собой цепью последовательного переноса.

Коды, подлежащие сложению в сумматоре, заносятся в РА и РВ (рис. 6.4). Процесс суммирования начинается сразу же после ввода слагаемых в регистры и по длительности может достигать 330 нс. Однако, как было указано выше, в целом на сложение двух 32-разрядных кодов в процессоре ЕС-2030 отводится один машинный такт, равный 600 нс.

Логические операции в соответствии с их алгоритмами выполняются путем побайтной переработки информации. Действия над разрядами байт производятся в комбинационном узле обработки байт УОБ (рис. 6.13). С помощью УОБ может осуществляться:

- выполнение логических операций над разрядами байт: И, ИЛИ, равнозначности, отрицания равнозначности (исключающего ИЛИ);
- сравнение и сопоставление величин 8-разрядных двоичных кодов (двух байтов);
- выборка нужного байта из 32-разрядного слова;

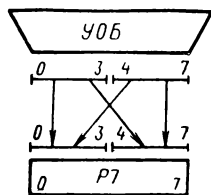


Рис. 6.13. Узел обработки байт

— изменение любого байта в слове;
 — замещение тетрады (старшей или младшей) одного байта тетрадой другого. При передаче результатов обработки байт в регистр результата P7 возможна перемена мест тетрад в байте благодаря наличию перекрестных связей, показанных на рис. 6.13.

6.5. АРИФМЕТИКО-ЛОГИЧЕСКИЙ БЛОК ПРОЦЕССОРА ЕС-2020

В арифметико-логическом блоке выполняются все арифметические и логические действия над операндами и адресными частями команд. Через АЛБ производится обмен информацией между регистрами процессора и обслуживание запросов мультимплексного канала.

В модели ЕС-1020 принят микропрограммный способ управления операциями. Микропрограммы, в соответствии с которыми производятся действия в АЛБ, основаны на реализации элементарных операций (микроопераций), перечисленных в табл. 6.2.

В этой таблице для модели ЕС-1020 через X и Y обозначены коды байтов соответственно 1-го и 2-го операндов. Перед началом очередной микрооперации эти коды заносятся соответственно в регистры А и В — входные регистры АЛБ (см. рис. 6.5).

Т а б л и ц а 6.2

Элементарные операции

Код операции	Операция	Название операции
0000	$X \vee Y$	Поразрядное логическое сложение
0001	Косвенная функция	Выполнение косвенной функции
0010	$X - Y$ (дес)	Вычитание (десятичное) 2-го операнда
0011	$X - Y$ (дв)	Вычитание (двоичное) 2-го операнда
0100	X ТРАНЗИТ	Транзитная пересылка 1-го операнда
0101	$X \wedge Y$	Поразрядное логическое умножение
0110	$Y - X$ (дес)	Вычитание (десятичное) 1-го операнда
0111	$Y - X$ (дв)	Вычитание (двоичное) 1-го операнда
1000	$X \vee \bar{Y}$	Поразрядное логическое сложение прямого кода 1-го операнда с обратным кодом 2-го
1001	Y ТРАНЗИТ	Транзитная пересылка 2-го операнда
1010	$X \oplus Y$	Поразрядная сумма по mod 2
1011	$\bar{X} \vee Y$	Поразрядное логическое сложение обратного кода 1-го операнда с прямым кодом 2-го
1100	Сдв. Y 1П	Сдвиг кода Y на один разряд вправо
1101	Сдв. Y 1Л	Сдвиг кода Y на один разряд влево
1110	$X + Y$ (дес)	Сложение (десятичное)
1111	$X + Y$ (дв)	Сложение (двоичное)

Микропрограммы всех операций, определяемых системой команд ЕС ЭВМ, действий над адресами, обмена и пр., составляются из микрокоманд перечисленных выше элементарных операций. Одна такая микрооперация выполняется в АЛБ за один машинный такт, равный 1 мкс. Исполнением микрооперации может управлять либо прямая, либо косвенная функция (см. гл. 7).

АЛБ, обеспечивающий реализацию перечисленных 15 элементарных операций, в своем составе содержит следующие узлы (рис. 6.14):

- входные регистры РА и РВ;
- первый и второй десятичные корректоры УДК1 и УДК2;
- узел перекосов УПК;
- логический узел УЛ;
- узел сдвига УСД;
- узел переноса УП;
- узел суммирования по mod 2, УСМ |2|;
- регистр байта состояния арифметико-логического блока РБС АЛБ.

Кроме регистров, все узлы, составляющие АЛБ,— комбинационные.

Входные регистры РА и РВ предназначены для хранения очередных байт операндов, обрабатываемых в АЛБ.

Первый десятичный корректор УДК1 при выполнении операций над десятичными данными используется для внесения в тетрады 1-го операнда избытков +6 (0110). В остальных арифметических и логических операциях над двоичными операндами информация при передаче через УДК1 не претерпевает изменений.

Узел перекосов УПК служит для передачи очередного байта из регистра РВ либо без изменений, либо меняя взаимное расположение и даже сочетание его тетрад (рис. 6.15). Тетрады либо могут меняться только местами, либо только одна из них может пере-

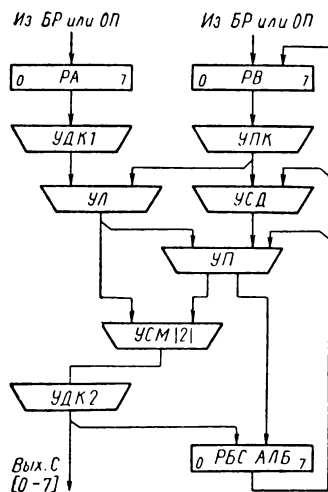


Рис. 6.14. Арифметико-логический блок

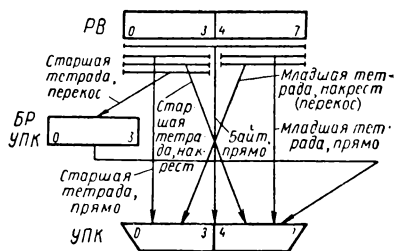


Рис. 6.15. Информационные связи регистра В с УПК

даваться прямо или накрест, а вместо другой при этом следуют нули. Кроме этого, может осуществляться пересылка тетрад со смещением, получившим название перекоса. При выполнении его младшая тетрада, выдаваемая из РВ, смещается влево на место старшей, а старшая — запоминается в буферном регистре БР УПК. Буферный регистр — двухступенчатый. Поступающая в него старшая тетрада запоминается в I ступени, а во II в это время еще хранится тетрада от предыдущего байта. Она в данном такте выдается на входы младших разрядов УПК. По окончании выдачи информации в УПК содержимое I ступени БР переписывается во II. Передача с перекосом используется для осуществления сдвигов влево сразу на четыре разряда.

Логический узел УЛ предназначен для выполнения поразрядных логических операций над кодами байт, прошедших через УДК1 и УПК. Режим работы УЛ, как и всех других узлов АЛБ, определяется исполняемой микрокомандой. Например, при арифметическом сложении кодов в нем образуется поразрядная сумма по mod 2 байт слагаемых, т. е. в каждом разряде получается $x_i \oplus y_i$. Совместно с кодом Y она затем используется в других узлах для формирования поразрядных переносов и вычисления окончательной суммы. При выполнении логических операций преобразование информации происходит только в УЛ. Остальные блоки АЛБ в этих случаях работают в режиме простой передачи.

Узел сдвига УСД служит для организации сдвигов влево или вправо на один разряд байт данных, поступающих в АЛБ через регистр В. Полностью эти микрооперации реализуются узлом сдвига совместно с узлом переносов УП. При арифметическом сложении байт второго слагаемого проходит через УСД без изменения, а при вычитании типа $X - Y$ он поразрядно инвертируется. В логических операциях узел сдвига для информации всегда закрыт (на выходах всех разрядов выдаются сигналы кода 0).

Узел переносов УП в первую очередь предназначается для выработки межразрядных и межбайтовых переносов. Эти переносы затем используются в узле суммирования по mod 2 для получения окончательных значений арифметической суммы или разности.

Межразрядные переносы q_i вырабатываются согласно логическим формулам:

$$q_i = \bar{s}'_{i+1} y_{i+1} \vee s'_{i+1} q_{i+1}, \quad (6.22)$$

где $s'_{i+1} = x_{i+1} \oplus y_{i+1}$ — сумма по mod 2 на выходе $(i + 1)$ -го разряда УЛ;

y_{i+1} — значение $(i + 1)$ -го разряда операнда, прошедшего через УСД;

q_{i+1} — перенос в $(i + 1)$ -й разряд;
 $i = 0, 1, 2, \dots, 7$.

Переносы между разрядами (между разрядными схемами формирования переносов СП на рис. 6.16) в младшей и старшей тетрадах

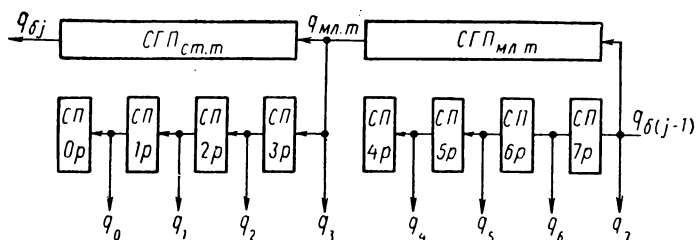


Рис. 6.16. Схема распространения переносов

распространяются последовательно. Кроме того, имеются схемы групповых переносов младшей и старшей тетрад (СГП_{мл.т} и СГП_{ст.т}). Межтетрадный ($q_{мл.т}$) и межбайтовый (q_6) переносы формируются этими схемами по принципу параллельного распространения переносов. Например, для младшей тетрады формула, согласно которой получается $q_{мл.т}$, имеет вид

$$q_{мл.т} = \bar{s}_4 y_4 \vee s_4 \bar{s}_5 y_5 \vee s'_4 s'_5 \bar{s}'_6 y_6 \vee s'_4 s'_5 s'_6 \bar{s}'_7 y_7 \vee s'_4 s'_5 s'_6 s'_7 q_7, \quad (6.23)$$

где q_7 — перенос в 7-й разряд, являющийся либо межбайтовым переносом q_6 ($j+1$) из предыдущего ($i+1$)-го байта, либо дополнительной единицей, подсуммируемой к младшему разряду при образовании дополнительного кода. Перенос q_{6j} , формируемый схемой групповых переносов старшей тетрады, является межбайтовым переносом из j -го байта в ($j-1$)-й. Он запоминается в триггере регистра байта состояния арифметико-логического блока РБС АЛБ и при сложении кодов следующих байтов передается в младший разряд как q_7 . Цепи формирования и распространения всех переносов — парафазные.

Кроме выработки переносов, узел УП совместно с УСД используется для сдвига кодов, поступающих на вход АЛБ через РВ, на один разряд вправо или влево. При выполнении логических операций на выходе УП устанавливается нулевой код.

Узел суммирования по mod 2 УСМ |2| получает информацию с выходов УЛ и УП. При выполнении микроопераций арифметического сложения и вычитания в этом узле происходит выработка окончательного результата. Он получается путем суммирования по mod 2 предварительной суммы по этому же модулю, полученной в УЛ, и кода переносов, сформированного в УП. Так как в логических операциях на выходе УП устанавливается нулевой код, то в этих случаях информация, сформированная в УЛ, проходит УСМ |2| без изменения.

Второй десятичный корректор УДК2 предназначен для коррекции результатов десятичного сложения и вычитания. К тем тетрадам, на выходах переносов которых при сложении не возник межтетрадный перенос, прибавляется по mod 16 код 1010 — дополнение 6 до 16. Таким образом, изымается избыток 6 из тех тетрад,

из которых он не удалится автоматически при действии над кодами в узлах УЛ, УП и УСМ | 2 |.

Во всех остальных микрооперациях, не относящихся к десятичным, УДК2 не меняет значение проходящей через него информации.

Регистр байта состояния арифметико-логического устройства РБС АЛУ предназначен для запоминания признаков, характеризующих результат действия в очередном машинном такте над одним или парой байтов данных. Эти признаки — межбайтовый перенос, знак байта результата, указатель переполнения, правильности десятичных данных и т. п. — используются как для обеспечения правильной обработки последующих байтов при непрерывной работе АЛУ, так и после прерывания. В случае прерывания исполняемой микропрограммы содержимое РБС заносится в РВ, а оттуда через АЛБ либо в какой-нибудь регистр блока БР процессора, либо в оперативную память. Если необходимо продолжить прерванную микропрограмму, то состояние АЛБ восстанавливается засылкой в РБС запомненного ранее байта состояния. Он вводится в АЛБ через регистр РА и, следуя транзитом, заносится в РБС.

Как следует из табл. 6.2, все элементарные операции делятся на три группы: логические и пересылочные, арифметические, сдвиговые.

Логические операции полностью выполняются в узле УЛ. Остальные узлы либо пропускают информацию, не изменяя ее (УДК1, УПК, УСМ | 2 |, УДК2), либо выдают на выходе нулевые коды (УСД, УП). При пересылочных операциях коды проходят через все узлы, в том числе и через УЛ, нигде не претерпевая изменений.

При выполнении арифметических действий количество узлов, участвующих в преобразовании информации, и характер их работы зависят от типа микрооперации. Например, сложение двух 8-разрядных двоичных чисел $X [0-7] = 0111 1101$ и $Y [0-7] = 1111 1011$ будет происходить так, как показано на рис. 6.17.

В отличие от этого, если над двоичными числами выполняется операция вычитания вида $X - Y$ или $Y - X$, то узлы, в особенности УЛ, работают несколько иначе. Обусловлено это тем, что при вычитании

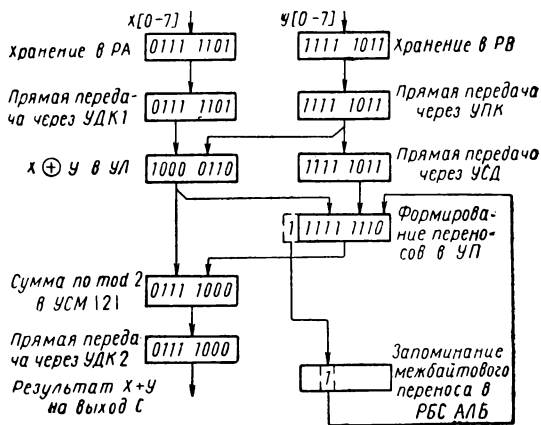


Рис. 6.17. Схема процесса сложения в АЛБ 8-разрядных двоичных кодов

необходимо брать дополнение для вычитаемого и уже в логическом узле УЛ оперировать с его поразрядной инверсией. Поразрядная же инверсия кода может быть произведена только в узле сдвигов УСД. Поэтому в узле УЛ поразрядная сумма по mod 2 $X \oplus \bar{Y}$ или $\bar{X} \oplus Y$ определяется следующим образом. Так как

$$x_i \oplus \bar{y}_i = \bar{x}_i \oplus y_i = x_i y_i \vee \bar{x}_i \bar{y}_i = \bar{s}_i, \quad (6.24)$$

то, вместо суммирования по mod 2 прямого кода уменьшаемого с обратным кодом вычитаемого, выполняется поразрядная операция равнозначности (эквиваленция). Результат ее следует в узел переносов УП, где совместно с кодом, поступающим с выхода УСД, участвует в образовании кода переносов.

Если выполняется операция $X - Y$ (дв.), то в УСД байт вычитаемого поразрядно инвертируется, и межразрядные переносы формируются согласно формуле

$$q_i = \bar{s}_{i+1} \bar{y}_{i+1} \vee \bar{s}_{i+1} q_{i+1}. \quad (6.25)$$

Пусть, например, требуется вычесть из двоичного числа $X [0-7] = 0111 \ 1111$ двоичное число $Y [0-7] = 0000 \ 0111$. В результате выполнения УЛ поразрядной операции равнозначности получится

$$\begin{array}{r} \infty \\ \hline S' [0-7] = \frac{0111 \ 1111}{0000 \ 0111} \end{array}$$

По этому результату совместно с инвертированным значением $[Y]_{\text{обр}} = \bar{Y} [0-7] = 1111 \ 1000$ согласно (6.25) сформируется код переносов:

$$\begin{array}{r} 1000 \ 011 \ 1 \text{ — код на выходе УЛ } (\bar{S}' [0-7]) \\ 1111 \ 100 \ 0 \text{ — код на выходе УСД } (Y [0-7]) \\ \hline \boxed{1} \boxed{1111} \boxed{111} \boxed{1} \text{ — код переносов на выходе УП} \end{array}$$

Выделенная рамкой младшая единица в коде переноса является дополнительной единицей, которую необходимо учитывать при образовании дополнительного кода Y . Старшая выделенная единица — межбайтовый перенос. В результате сложения в УСМ $|2|$ получится следующий результат:

$$\begin{array}{r} \oplus 1000 \ 0111 \text{ — код на выходе УЛ} \\ 1111 \ 1111 \text{ — код на выходе УП} \\ \hline 0111 \ 1000 \text{ — результат операции } X - Y \text{ (дв.)} \end{array}$$

При выполнении операции $Y - X$ (дв) код уменьшаемого (Y) проходит через УСД, не инвертируясь. Межразрядные переносы в УП формируются согласно

$$q_i = \bar{s}_{i+1} y_{i+1} \vee \bar{s}_{i+1} q_{i+1}. \quad (6.26)$$

Например, требуется вычесть из двоичного числа $Y [0-7] = 0000 0111$ двоичное число $X [0-7] = 0111 1111$. Выполнение операции в УЛ даст тот же результат, что и в предыдущем случае, т. е.

$$\overline{S' [0-7]} = X [0-7] \oslash Y [0-7] = 1000 0111.$$

Согласно (6.26) в УП сформируется код переносов:

$$\begin{array}{r} 1000 \ 011 \ 1 \text{ — код на выходе УЛ}(\overline{S' [0-7]}) \\ 0000 \ 011 \ 1 \text{ — код на выходе УСД} (Y[0-7]) \\ \hline \text{0} \text{0000} \ \text{111} \ \text{1} \text{ — код переносов на выходе УП.} \end{array}$$

Здесь также перенос в младший разряд представляет собой дополнительную единицу, необходимую для образования дополнительного кода X . Поразрядное сложение по mod 2 в УСМ | 2 | даст следующий результат:

$$\oplus \begin{array}{r} 1000 \ 0111 \text{ — код на выходе УЛ} \\ 0000 \ 1111 \text{ — код на выходе УП} \\ \hline 1000 \ 1000 \text{ — результат операции } Y - X \text{ (дв)} \end{array}$$

Выполнение операций над десятичными операндами во многом аналогично рассмотренному выше. Добавляются лишь дополнительные корректирующие действия в узлах десятичной коррекции УДК1 и УДК2. Характер этих действий рассмотрен в пп. 2.5 и 3.4. Там же имеются и иллюстрирующие их примеры.

Операции сдвига кодов, поступающих в АЛБ через регистр РВ, на разряд влево или вправо производятся узлами УСД и УП. Сдвиг влево фактически выполняется в УП. Для этого в УЛ формируется нулевой код, который подается в УП совместно с кодом Y , следующим через УСД. На выходах УП согласно (6.22) при $s'_i = 0$ будем иметь:

$$q_i = \overline{0}y_{i+1} = y_{i+1}.$$

Например, пусть сдвигается код $Y_2 = 1111 1101$. В результате действий в разрядах УП получим

$$\begin{array}{r} 0000 \ 000 \ 0 \text{ — код на выходе УЛ} \\ 1111 \ 110 \ 1 \text{ — код на выходе УСД} \\ \hline \text{1} \text{1111} \ 101 \ \text{0} \text{ — результат сдвига влево.} \\ q_{6j} \qquad \qquad q_6 (j+1) \end{array}$$

При сдвиге влево кодов, состоящих из нескольких байт, значение разряда, соответствующего межбайтовому переносу q_{6j} , запоминается в РБС и используется при сдвиге соседнего слева ($j - 1$)-го байта. Оно заносится в его младший разряд, освобождающийся при сдвиге.

Аналогично рассмотренному работает УП и при сдвиге кода на разряд вправо. Однако, чтобы получить такой сдвиг, сдвигаемый

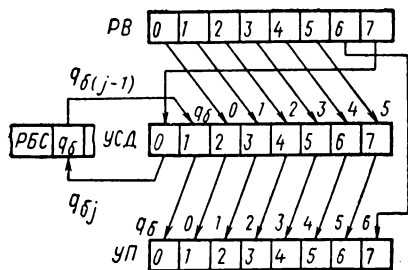


Рис. 6.18. Схема сдвига на разряд вправо

7-го разряда предыдущего ($j - 1$)-го байта как $q_{6(j-1)}$, подается из схемы памяти на вход 1-го разряда УСД. Для того чтобы можно было одновременно запоминать q_{6j} и выдавать $q_{6(j-1)}$, схема памяти выполнена двухступенчатой. По нулевому коду, поступающему с выходов УЛ, и коду с выходов УСД в УП формируется код, сдвинутый относительно исходного на один разряд вправо.

Исполнением каждой микрооперации в АЛБ управляют микрокоманды (см. гл. 7). В микрокоманде тип исполняемой микрооперации определяется четырехразрядным полем ФУНКЦИЯ. Соответствие возможных кодов в этом поле различным микрооперациям дано в табл. 6.2. При необходимости выполнить косвенную функцию, код ее заносится в специальный регистр косвенной функции в блоке регистров процессора. Задание этого кода производится одной из предшествующих микрокоманд.

В микрокоманде элементарной операции, исполняемой в АЛБ, указываются номера регистров, из которых должна приниматься информация в регистры РА и РВ и куда должен выдаваться результат с выхода С. Имеется в ней также поле ДЕФОРМАЦИЯ. Трехразрядный код в этом поле определяет способ пересылки очередного байта из РВ через УПК на входы УЛ и УСД. Например, код 000 устанавливает прямую передачу, код 001 — передачу тетрады байта накрест, код 110 — передачу с перекосом и т. п.

6.6. ОСОБЕННОСТИ ПЕРЕМЕЩЕНИЯ И ПРЕОБРАЗОВАНИЯ ОПЕРАНДОВ ОДИНАКОВЫХ ОПЕРАЦИЙ В АРИФМЕТИКО-ЛОГИЧЕСКИХ УСТРОЙСТВАХ РАЗЛИЧНЫХ МОДЕЛЕЙ

В п. 3.1 кратко рассмотрен общий порядок действий, производимых в процессоре любой модели ЭВМ ЕС при выполнении арифметической или логической операции. Там же перечислены некоторые основные причины, в первую очередь обусловившие разнообразие алгоритмов одних и тех же операций, реализованных в различных моделях. Принципы выполнения арифметических и логических операций, положенные в основу построения и организации работы АЛУ процессоров ЕС-2020, ЕС-2030 и ЕС-2050, изложены в последующих параграфах гл. 3.

Здесь же рассмотрим особенности взаимодействия узлов и блоков различных АЛУ в процессе выполнения ряда операций, а также особенности перемещения и преобразования при этом участвующих в них операндов. Изложение материала данного параграфа основывается на знании содержания гл. 3.

Арифметические операции над операндами с фиксированной запятой. Различия в выполнении этих операций в разных моделях машин обусловлены особенностями построения операционных частей АЛУ.

Сложение и вычитание. Целые отрицательные числа из памяти поступают в АЛУ в дополнительном коде.

В процессоре ЕС-2050 сложение и вычитание 32-разрядных операндов может происходить как в старших разрядах сумматора СМ БАС, так и в младших (рис. 6.3). Все зависит от того, из какого регистра РОН выбирается первый операнд. Если он берется из регистра с четным номером, то операнды из БЦД принимаются в РВ1 [0—31] (1-е слагаемое, уменьшаемое) и РВ2 [0—31] (2-е слагаемое, вычитаемое), а далее соответственно — в РБ1 [0—31] и РБ2 [0—31]. В разрядах с 32-го по 63-й всех регистров находятся нули. Сложение кодов происходит в старшей половине сумматора. При этом нулевой разряд является знаковым, относительно которого анализируется возможность переполнения и по которому определяется знак результата. Переполнение имеет место в том случае, если перенос в нулевой разряд и из нулевого разряда не согласуются. Результат, полученный в сумматоре и зафиксированный в регистре результата РР, передается в РВ1 [0—31], откуда записывается в РОН по адресу R1.

Если же 1-й операнд выбирается из РОН с нечетным номером, то оба операнда движутся в БАС через младшие [32—63] разряды регистров РВ1, РБ1 и РВ2, РБ2. Теперь старшие разряды всех регистров БЦД и БАР — с 0-го по 31-й — находятся в нулевом состоянии. Коды операндов суммируются в младшей половине СМ и результат фиксируется в РР [32—63]. В этом случае роль знакового выполняет 32-й разряд. Запись результата в РОН по адресу R1 производится через 32—63 разряды РВ1.

Для указания того, какие половины регистров и сумматора должны работать, в блоке управления АЛУ вырабатывается управляющий сигнал. Он выдается специальным триггером четности адреса 1-го операнда, принимающим единичное состояние, когда адрес R1 — четный, и нулевое, если R1 — нечетный.

В процессоре ЕС-2030 сумматор АЛУ — 32-разрядный (рис. 6.4). Для выполнения операций сложения и вычитания операнды из РОН или ООП через Р5 [0—31] засылаются в Р1 [0—31] и Р3 [0—31]. В Р3 помещается первый операнд (1-е слагаемое, уменьшаемое), а в Р1 — второй (2-е слагаемое, вычитаемое). Для сложения в сумматоре СМ коды операндов передаются в РА и РВ. Если производится вычитание, то при передаче вычитаемого из Р1 в РА, оно инвертируется. Полученный в сумматоре результат через Р5 отсылается в РОН по адресу R1. По значению содержимого нулевого разряда и по переносам в нулевой разряд и из нулевого вырабатывается признак результата.

В процессоре ЕС-2020 из РОН или из РОН и ООП для выполнения операций операнды вызываются по частям. Вначале в регистры Н и З (см. рис. 6.5) выбираются два младших байта второго операнда (2-е слагаемое, вычитаемое). Затем через АЛБ байт из регистра Н пересылается в регистр Д, а байт из регистра З — в регистр Л. После этого снова производится обращение к ОП и из РОН выбираются два младших байта первого операнда (1-е слагаемое, уменьшаемое).

Над выбранными двумя полусловами выполняются действия, соответствующие исполняемой команде. Сначала в АЛБ обрабатываются младшие байты операндов, находящиеся соответственно в регистрах З и Л, а затем старшие из регистров Н и Д. Байты — результаты с выхода С АЛБ заносятся в регистры З и Н, после чего полученная младшая половина результата записывается по адресу R1 в 16—31-й разряды РОН. Далее действия повторяются для старших половин слов операндов.

При выполнении микрооперации над старшими байтами операндов формируется признак результата. По значению знакового разряда и по переносам он вырабатывается так же, как и в других моделях.

Умножение. Оно может выполняться как путем перемножения 32-разрядных операндов, так и 32-разрядного на 16-разрядный. Если в первом случае форми-

руется 64-разрядное произведение, то во втором — 32-разрядное. И то, и другое в конце операции заносится в РОН.

В модели ЕС-1050 умножение производится одновременно на 4 разряда множителя, начиная с его младших разрядов (см. гл. III). Множимое из РЧ1 БЦД поступает в РВ1 [0—31] БАР, а множитель, если он 32-разрядный, — из РЧ2 либо в РВ2 [32—63], либо, если он 16-разрядный, в РВ2 [48—63] (рис. 6.3). Разряды РВ1 и РВ2, в которые не заносятся операнды, устанавливаются в нулевое состояние.

Перед началом операции умножения происходит предварительный модифицированный сдвиг множимого на четыре разряда вправо. Это необходимо для исключения потери значащих разрядов при образовании частных произведений, когда множимое приходится сдвигать влево. Сдвиг на четыре разряда вправо осуществляется путем передачи множимого через соответствующие цепи из регистра РВ1 и РБ1 и обратно. Так как выполняется модифицированный сдвиг, то значение знакового разряда при этом распространяется вправо.

Четыре младших разряда множителя (РВ2 [60—63]) дешифрируются специальным дешифратором. По результату дешифрирования формируются два частных произведения (см. п. 3.2). Первое частное произведение через соответствующие цепи заносится в РБ1, а второе — в РБ2. На первом шаге умножения в сумматоре СМ БАС суммируются только содержимые РБ1 и РБ2. На последующих шагах вместе с очередной парой частных произведений на третий вход сумматора из РР подается текущая сумма частных произведений. Эта текущая сумма, полученная на предыдущем шаге умножения, подается на вход сумматора со сдвигом на 4 разряда вправо.

По окончании умножения произведение из РР передается в РВ1, откуда записывается в РОН. Если по командам УМНОЖИТЬ формата *RR* или *RX* перемножаются 32-разрядные операнды, то 64-разрядное произведение помещается в два смежных регистра РОН, с четным и нечетным номерами соответственно. Если же по команде формата *RX* умножение осуществляется на 16-разрядный множитель, то 32-разрядное произведение из РР передается либо в старшие разряды РВ1, либо — в младшие. При *R1* четном результат передается в РВ1 [0—31], а при нечетном — в РВ1 [32—63]. После этого оно записывается соответственно либо в РОН с четным номером, либо — с нечетным.

В процессоре модели ЕС-1030 32-разрядное множимое перед умножением помещается в регистр Р1, а множитель — в Р4 (см. рис. 6.4). В двухразрядном сдвигающем регистре Р4—Р4' на каждом очередном шаге умножения производится сдвиг множителя вправо. Очередные разряды множителя выдаются в схему анализа из 30 и 31 разрядов Р4. Алгоритм умножения модели ЕС-1030 дан в п. 3.2. Текущая сумма частных произведений с выхода СМ заносится в Р5, а оттуда — в Р3. Сдвиг этой суммы без сложения с очередным частным произведением или после сложения производится путем соответствующей передачи из СМ в Р5 со смещением в нужную сторону и на необходимое количество разрядов. Младшие разряды произведения, при сдвиге вправо выходящие из разрядной сетки Р5, заносятся в старшие освобождающиеся разряды Р4. В этом регистре они сдвигаются вправо вместе с множителем, освобождая место для следующих разрядов произведения. Окончательное произведение из Р3 и Р4 через РВ, СМ и Р5 отсылается в РОН МОП.

В модели ЕС-1020 умножение ведется, начиная с младших разрядов множителя, с одновременной расшифровкой четырех его разрядов. При этом специфика АЛУ этой модели и последовательно-параллельный способ выполнения в нем операций обусловили необходимость реализации такого алгоритма, при котором все кратные множимого оказывается возможным формировать перед началом собственно процесса умножения (см. п. 3.2). В результате умножение сводится к суммированию комбинаций этих кратных величин в зависимости от расшифровываемых тетрад множителя.

Каждой тетраде множителя ставится в соответствие не более двух частных произведений, представляющих собой кратные множимого *A* (см. табл. 3.2).

В начале операции умножения из МОП по адресу *R1* полусловами выбирается множимое *A*. Для его абсолютного значения определяются кратные величины: *2A*, *3A* и *6A*. Они вместе с абсолютным значением $[A]$, представляющим собой

кратное 1А, надлежащим образом размещаются в ячейках МОП, образующих рабочую память процессора. Затем из МОП или ООП по второму адресу в процессор выбирается множитель и размещается в регистрах Д, Ф, Е и Л БР (см. рис. 6.5).

Анализ множителя начинается с анализа тетрад байта, расположенного в регистре Л. По результату этого анализа из рабочей памяти выбираются необходимые кратные множимого, которые подсуммируются к соответствующим образом сдвинутой текущей сумме частных произведений. Для самой младшей тетрады кратные множимого подсуммируются к нулевой сумме. Процесс суммирования протекает так же, как и при выполнении операции сложения чисел с фиксированной запятой.

Текущая сумма частных произведений также хранится в МОП в рабочей памяти процессора. После умножения на обе тетрады очередного байта множителя подготовка следующего байта для анализа производится сдвигом содержимого регистров Д, Ф, Е таким образом, что следующий байт смещается в регистр Л. Так как в модели ЕС-1020 на множитель в коде, соответствующем знаку, умножается абсолютное значение множимого, то знак последней суммы частных произведений соответствует знаку множителя. Код ее соответствует этому знаку. Для получения действительного знака произведения и соответствующего ему кода результата анализируется знак множимого. Если множимое отрицательно, то для кода последней суммы частных произведений берется дополнение. Таким образом, при положительном множимом последняя сумма частных произведений, а при отрицательном — ее дополнение, засылается в МОП по адресу первого операнда.

Деление. Независимо от формата команды в операции этого типа 64-разрядное делимое (1-й операнд) делится на 32-разрядный делитель (2-й операнд). Результат — частное с остатком — помещается по первому адресу на место делимого.

В модели ЕС-1050 в начале операции деления делитель выбирается из РОН или ООП и заносится в РВ2 БАР (см. рис. 6.3). Делимое из РОН поступает в РВ1 [0—63]. В РВ2 делитель вводится в старшие и младшие разряды, т. е. в РВ2 [0—31] и РВ2 [32—63]. Это необходимо для осуществления проверки корректности деления по формуле $2^{31}d + d - D$ при разных знаках операндов. При одинаковых знаках операндов проверка выполняется согласно формуле $2^{31}d - D$.

Величина d образуется путем передачи содержимого РВ2 [32—63] в РВ2, а затем — через сумматор в РР [32—63]. При пересылке из РВ2 [32—63] в РВ2 [32—63] знак делителя распространяется на старшие разряды РВ2, т. е. на разряды с 0 до 31. Величина $2^{31}d$ образуется путем передачи содержимого РВ2 [0—31] в РВ2 [0—63] со сдвигом на один разряд вправо. Делимое в нужном для вычитания коде пересылается из РВ1 [0—63] в РБ1 [0—63]. Из РБ1, РВ2 и РР слагаемые поступают на три входа сумматора СМ (согласно второй формуле при одинаковых знаках операндов содержимое РР на вход СМ не подается). Полученная в РР сумма анализируется. Если деление возможно, то выполнение операции продолжается.

Перед началом многошагового процесса собственно деления в модели ЕС-1050 производится нормализация делителя путем сдвига его влево. Для этого он вначале передается из РВ2 [0—31] в РВ2 [0—63], а оттуда в РВ1 [0—63]. Сдвиг влево на 8, 4 или на 1 разряд осуществляется путем соответствующих обменных передач между РВ1 и РБ1. Сдвиг на 2 разряда влево производится при пересылке из РВ1 в РВ2. Вначале анализируются 8 старших разрядов делителя. Если в этих разрядах оказываются незначительные цифры, то он сдвигается влево сразу на 8 разрядов. Затем снова производится анализ 8 старших разрядов, в результате чего может выясниться, что либо возможен повторный сдвиг на 8 разрядов, либо надо перейти к определению возможности сдвига на 4 разряда. Такой итерационный процесс сдвига на 8, 4, 2, 1 разрядов продолжается до тех пор, пока делитель не будет удовлетворять условию: $2^{30} \leq |d| \leq 2^{31}$. Количество разрядов, на которое сдвигается делитель, подсчитывается. Затем на это же количество разрядов сдвигается влево делимое. Нормализованный делитель через сумматор пересылается в РР [0—31], где и находится до конца деления.

Перед первым вычитанием делителя из делимого последнее сдвигается на 1 разряд влево. На вход сумматора код сдвинутого делимого поступает из РВ1 [0—63], а код делителя — из РР [0—31]. Далее выполняется процесс деления, алгоритм которого рассмотрен в п. 3.2. В ходе его двухрядные коды остатков располагаются в РС и РП (см. рис. 6.6). При очередном сложении или вычитании код S из РС подается через РВ1 на вход ПСМЗ со сдвигом влево на 1 разряд, а код Q из РП через РВ2 — со сдвигом в ту же сторону на 2 разряда.

На каждом шаге деления полностью приводятся переносы только в пяти старших разрядах двухрядного кода остатка. Приведение осуществляется в специальном трехходовом 5-разрядном сумматоре схемы управления делением СУД, расположенной в блоке управления АЛУ. По результату этого приведения определяется требуемый характер действий на следующем шаге деления (прибавление или вычитание кода делителя или просто сдвиг влево двухрядного кода остатка).

Определяемые на каждом шаге разряды положительной и отрицательной составляющих частного накапливаются в специальных 8-разрядных регистрах РД1 и РД2 блока АР. Эти регистры на схеме рис. 6.3 не показаны. После выполнения очередных восьми шагов деления и, следовательно, заполнения РД1 и РД2 их содержимые заносятся в соответствующие 8 разрядов соответственно РВ1 и РВ2. Затем регистры РД1 и РД2 очищаются и, таким образом, подготавливаются к накоплению следующих цифр составляющих частного.

По окончании 32-го шага деления или раньше, при обнаружении нулевого остатка, по положительной и отрицательной составляющим частного, соответственно расположенным в РВ1 и РВ2, вычисляется действительный код частного. Для этого содержимое РВ2 вычитается из РВ1. Одновременно, если необходимо, корректируется последний остаток. Он должен иметь тот же знак, что и делимое. Кроме того, его необходимо уменьшить, если перед делением производилась нормализация делителя. В этом случае остаток сдвигается вправо на столько разрядов, на сколько сдвигался влево делитель при нормализации. При компоновке результата перед отсылкой в РОН откорректированный остаток помещается в РВ1 [0—31], частное — в РВ1 [32—63].

В модели ЕС-1030 в начале деления делимое (первый операнд) в АЛУ располагается в Р3 и Р4, а делитель (второй операнд) — в Р1. Если после анализа делителя и делимого на нуль выясняется, что деление должно быть выполнено, то производится проверка корректности деления. Для этого анализируется на значимость 1-й разряд делимого. Если он — незначащий, то делимое сдвигается на разряд влево и из него алгебраически вычитается делитель. Путем сравнения знаков первого остатка и делителя вырабатывается значение нулевого разряда частного. Одновременно определяется корректность деления. Если устанавливается, что переполнения результата не будет, то процесс деления продолжается.

Для получения 32-разрядного частного выполняется 32 шага деления (см. алгоритм в п. 3.2). Деление производится по методу без восстановления остатка. На каждом шаге содержимое Р3 и Р4 сдвигается на разряд влево. Сдвиг осуществляется путем соответствующих передач их содержимых через СМ и Р5.

Очередная цифра частного заносится в специальный 4-разрядный сдвигающий регистр Р6 (на рис. 6.4 не показан). После накопления в Р6 тетрады частного она через Р5 заносится в освобождающиеся разряды Р4. Таким образом, в конце деления остаток оказывается расположенным в Р3, а частное — в Р4. Перед записью в РОН при необходимости производится коррекция результата.

В модели ЕС-1020 также принят метод деления без восстановления остатка. Делятся абсолютные значения делимого и делителя. Для вычитания делителя из делимого и положительных остатков используется его дополнение.

В начале операции делитель из РОН или ООП, начиная со старшего полуслова, выбирается в регистры Р, И, Т и Л БР (рис. 6.5). В процессе выборки анализируется его знак. Отрицательный делитель заносится в регистры непосредственно в дополнительном коде, а для положительного делителя предварительно вычисляется дополнение. Таким образом, в указанных регистрах фактически оказывается расположенным дополнение абсолютного значения делителя. Это оказалось удобно для реализации операции.

Делимое из РОН МОП засылается в регистры Ф, Е, У, Д, Н и З. Два младших байта делимого (6-й и 7-й) записываются и временно хранятся в рабочей области МОП. В процессе исходного размещения делимого определяется код его абсолютного значения. Для этого отрицательное число вычитается из нуля, а положительное складывается с нулем. Если делимое — минимальное отрицательное число, то это обнаруживается по отрицательному результату после взятия дополнения.

Из четырех старших байт делимого, расположенных в регистрах Ф, Е, У и Д, вычитается делитель. Первое пробное вычитание производится, как и все последующие, последовательно, побайтно, начиная с младших байтов (с содержимых регистров Д и Л). При пробном вычитании проверяется корректность деления и определяется первая цифра частного, которая должна быть равна нулю. Если деление возможно, то первый остаток сдвигается на разряд влево. При этом на освободившееся место в регистре Д вдвигается (сносится) разряд делимого. Для этого вместе со сдвигами старших четырех байт остатка сдвигаются влево и два младших байта делимого, расположенные в регистрах Н и З.

Сдвинутый первый остаток складывается с делителем. Определяемая в результате этого следующая цифра частного заносится в освободившийся младший разряд регистра З. Затем следует снова сдвиг остатка влево, сносится очередная цифра делимого (вдвигается в регистр Д цифра из старшего разряда регистра Н), выполняются действия по определению следующей цифры частного и т. д. Таким образом, в регистрах Н и З вначале формируется старшее полуслово частного, которое временно запоминается в рабочих ячейках МОП. На место него из рабочих ячеек МОП выбираются два младших байта делимого. Деление продолжается до тех пор, пока в регистрах Н и З не сформируются два младших байта частного.

Остаток деления должен иметь тот же знак, что и делимое. Поэтому, если последний остаток получается отрицательным, то к нему добавляется делитель. Остатку присваивается знак делимого, а частному — знак, определенный согласно алгебраическому правилу по знакам делимого и делителя. При отсылке результата в РОН остаток и частное представляются в кодах, соответствующих их знакам.

Арифметические операции над операндами с плавающей запятой. Во всех моделях для выполнения этих операций используется в основном то же оборудование процессоров, что и для операций над операндами с фиксированной запятой.

Сложение и вычитание. В отличие от целых двоичных чисел отрицательные числа с плавающей запятой хранятся в памяти в прямом коде.

В модели ЕС-1050 в исходном положении для выполнения операции операнды располагаются в регистрах РЧ1 (1 операнд) и РЧ2 (2 операнд) (см. рис. 6.3). Короткие операнды занимают разряды 0—31, а длинные — 0—63. Затем мантиссы 1-го и 2-го операндов передаются без смещения в соответствующие разряды РВ1 и РВ2, а характеристики — в регистры порядков РП1 и РП2 блока арифметического управления БАУ процессора. В блоке АУ характеристики сравниваются, для чего вторая вычитается из первой. Для этой цели в нем имеется сумматор порядков.

Вывравнивание порядков операндов осуществляется сдвигом вправо мантиссы с меньшей характеристикой. Он может быть выполнен только путем соответствующих обменных передач между регистрами РВ1 и РВ1. Поэтому, если меньшую характеристику имеет 1-й операнд, то сдвигается вправо его мантисса. Если же, наоборот, в результате сравнения характеристик выясняется, что меньшая — у 2-го операнда, то его мантисса из РВ2 через РВ2 пересылается в РВ1. В то же время мантисса 1-го операнда из РВ1 через РВ1 помещается в РВ2, т. е. происходит взаимоперекрестный обмен мантиссами между входными регистрами.

Для выравнивающего порядка сдвига мантисс разность характеристик отсылается в счетчик сдвига БАУ. По содержимому его младших разрядов устанавливается, на сколько разрядов одновременно можно осуществлять сдвиги (на 8 или на 4). Если производится сдвиг на 8 разрядов (на две шестнадцатичные позиции), то содержимое счетчика уменьшается на 2 единицы, если на 4 (на одну шестнадцатичную позицию), то — на одну единицу. Сдвиги прекращаются, когда содержимое счетчика становится равным нулю.

Сложение и вычитание кодов мантисс, имеющих одинаковые порядки, происходит согласно алгоритму, рассмотренному в п. 3.3. Мантисса суммы (разности) в прямой коде получается в регистре результата РР, откуда она отправляется в РВ1. При необходимости эта мантисса нормализуется. Одновременно корректируется результирующая характеристика. После анализа ее и мантиссы вырабатывается признак результата РР. Затем характеристика и знак результата заносятся в разряды 0—7 РВ1, откуда вместе с мантиссой отправляются по первому адресу в РПЗ БЦР.

Для определения знака суммы кодов мантисс и выявления случаев переполнения в мантиссе результата, применен модифицированный дополнительный код. В качестве знаковых разрядов используются 6-й и 7-й разряды регистров и сумматора. Для мантисс отведены разряды с 8-го по 35-й (при коротких операндах) или по 67-й (при длинных). В обоих случаях разряды 32—35 и 64—67 используются для целей округления.

Знак конечного результата определяется путем совместного анализа знака суммы кодов выравненных мантисс (по значению 6 разряда РР), а также кода операции, результата сравнения знаков исходных операндов и признака, характеризующего наличие или отсутствие перемены мест мантисс. Учет кода операции необходим потому, что при вычитании знак второго операнда искусственно меняется на обратный. Код суммы мантисс, поступающей из сумматора в РР (прямой или дополнительный), устанавливается по значению содержимого 6-го разряда РР. Если $(РР [6]) = 1$, то требуется перевод мантиссы результата в прямой код. Для этого берется дополнение мантиссы суммы путем вычитания ее из кода нуля: на один вход сумматора поступает обратный код содержимого РР, на два других — коды нуля из РВ1 и РВ2 и на вход переноса младшего разряда сумматора СМ — код 1.

В модели ЕС-1030 основное отличие в выполнении сложения и вычитания операндов с плавающей запятой заключается в методике сравнения и выравнивания порядков. Сравнение характеристик производится в узле обработки байт УОБ (рис. 6.4).

При выполнении операций над короткими операндами 1-й операнд находится в Р3, а 2-й — в Р1. В случае длинных операндов их старшие и младшие половины соответственно располагаются: в Р3 и Р4 (1-го операнда) и в Р1 и Р2 (2-го операнда). Из разрядов 1—7 регистров Р3 и Р1 характеристики 1-го и 2-го операндов поступают в регистры РД и РС. Одновременно характеристика 1-го операнда посылается в регистр-счетчик РС1 (на рис. 6.4 не показан).

Сопоставлением в УОБ величин кодов, находящихся в РД и РС, определяется меньшая характеристика; соответствующая ей мантисса сдвигается вправо на одну шестнадцатиричную позицию. Если сдвигается мантисса 1-го операнда, то к содержимому РС1 прибавляется единица, если 2-го, то единица вычитается. После этого код из РС1 посылается в РД и сопоставляется по величине с содержимым РС. Такой порядок действий и сдвиги продолжаются до тех пор, пока не станут равными коды в РС и РС1. Предварительной характеристикой результата является большая, которая теперь запоминается в РС1.

Сдвиги операндов при выравнивании порядков осуществляются путем их пересылки через сумматор СМ в регистр Р5 и затем обратно в исходный регистр. В выходном коммутаторе сумматора имеются цепи передачи в Р5 со сдвигом на 4 разряда вправо (П4).

Далее операция сложения или вычитания выполняется согласно алгоритму, рассмотренному в п. 3.3. При сложении мантисс в сумматоре их значащие части находятся правее 7-го разряда. Запятая считается фиксированной между 7-м и 8-м разрядами сумматора. По значению содержимого 7-го разряда, по переносам в этот разряд и из него производится определение знака разности мантисс, выявляются случаи получения результата в дополнительном коде, нарушения нормализации влево (переполнение в мантиссе). Для обнаружения случаев нарушения нормализации вправо анализируются на ноль с 8-го по 11-й разряды сумматора.

В модели ЕС-1020 сравнение порядков осуществляется путем вычитания характеристики второго операнда из характеристики первого. При выравнивании порядков сдвигается вправо мантисса операнда с меньшей характеристикой.

Особенность действий по выравниванию состоит в том, что перед сложением кодов перемещение сдвигаемой мантиссы не производится. Вместо этого просто выбирается новый адрес сдвигаемого операнда. Адрес его байта, который должен стать младшим после сдвига, определяется путем вычитания целой части от деления на два разности характеристик из адреса младшего байта операнда, подлежащего сдвигу. Если разность характеристик — нечетное число, то окончательный сдвиг на четыре разряда, который не учитывается при формировании нового адреса, производится с помощью соответствующего размещения в регистрах БР (рис. 6.5), выбираемых из памяти мантисс и подачи одной из них на сложение в АЛБ «с перекосом».

В целом в этой модели алгоритм операций сложения и вычитания операндов с плавающей запятой соответствует рассмотренному в п. 3.3.

Умножение и деление. Во всех моделях перед выполнением этих операций мантиссы операндов нормализуются. Исключением является операция умножения коротких операндов в модели ЕС-1030, в которой предварительная нормализация сомножителей короткого формата не производится. Абсолютные значения нормализованных мантисс перемножаются или делятся согласно алгоритмам, рассмотренным в п. 3.3.

В модели ЕС-1050 множимое и множитель любой длины в процессе умножения целиком находятся в РВ1 и РВ2 (рис. 6.3), а делимое и делитель при делении — соответственно в РВ1 и РР. Действия над порядками производятся в сумматоре порядков блока арифметического управления, в регистры РП1 и РП2 которого они передаются в начале операции.

В модели ЕС-1030 (рис. 6.4) 1-й операнд (множимое, делимое) в начале операции размещается в Р3 (короткий операнд) или в Р3 и Р4 (длинный операнд), а 2-й операнд (множитель, делитель) — в Р1 (короткий операнд) или в Р1 и Р2 (длинный операнд). Короткие операнды перемножаются и делятся, целиком находясь в АЛУ. Мантиссы длинных операндов перемножаются и делятся по частям. При длинных мантиссах вначале содержимые Р1 и Р2 умножаются на содержимое Р4, а затем — на содержимое Р3. Промежуточные произведения, соответствующим образом сдвинутые друг относительно друга, складываются так, что получается 56-разрядная мантисса произведения. Для того чтобы можно было реализовать такой процесс умножения, перед его началом содержимые регистров Р1, Р3 и Р4 запоминаются в рабочих регистрах РР местной оперативной памяти МОП. По мере необходимости эти части мантисс считаются из МОП в АЛУ процессора. Мантисса произведения в итоге получается в регистрах Р3 и Р4.

При делении длинных операндов в рабочие регистры МОП перед началом операции засылаются содержимые регистров Р1 и Р2, в которые был выбран делитель. В процессе деления он по частям вводится в Р1. Старшая часть делимого и остатков все время находится в Р3, а его младшая часть из Р4 помещается в Р2. В регистре Р4 в процессе деления постепенно формируется частное. Оно сюда поступает тетрадами из специального 4-разрядного сдвигающего регистра Р6, где вначале накапливаются его разряды, получаемые в процессе деления. При действии с длинными операндами формирование частного происходит в два этапа. Сначала в регистре Р4 накапливаются 24 старших разряда мантиссы частного. Эта часть отсылается по адресу 1-го операнда в РПЗ с четным номером. После этого в Р4 формируется младшая часть мантиссы, которая в конце процесса деления запоминается в РПЗ, имеющем следующий, больший на единицу нечетный адрес.

В начале операции анализируются мантиссы сомножителей или делимого и делителя на нуль. Если умножение и деление возможно, то выполнение операции начинается соответственно со сложения или вычитания характеристик. Коды их суммируются в старших разрядах 0—7 сумматора. Полученная предварительная характеристика результата через Р7 посылается на хранение в регистр-счетчик АЛБ РС1. При умножении окончательная характеристика результата получается после нормализации мантиссы произведения. При делении окончательная характеристика частного образуется в начале операции. Для этого мантиссы нормализованных делимого и делителя сравниваются. Если разность окажется положительной, то мантисса делителя сдвигается влево на 4 разряда, а к характеристике,

находящейся в РС, прибавляется единица. После этого выполняется процесс деления, в результате которого мантисса частного всегда получается нормализованной.

В модели ЕС-1020 после нормализации исходных операндов и вычисления предварительных характеристик результатов заготавливаются кратные множители: А, 2А, и 5А (табл. 3.3), — при умножении или кратные делителя: d, 2d, 4d и 8d, — при делении. Эти кратные величины запоминаются в рабочей области МОП. Умножение и деление в итоге сводятся к многоцикловым процессам их сложений и вычитаний в зависимости либо от расшифровываемых тетрад множителя при умножении, либо — от знаков остатков при делении. Такие действия выполняются аналогично рассмотренным для операндов с фиксированной запятой.

Операции над десятичными числами. Такие операции относятся к типу память—память, так как десятичные операнды выбираются из основной оперативной памяти и результаты также заносятся туда же по адресу 1-го операнда. Поля, в которых располагаются операнды, могут начинаться с любого байта и иметь длину от одного до 16 байт. Это обстоятельство обуславливает то, что ни в одной модели операнды максимальной длины не могут быть полностью размещены в регистрах процессора. Поэтому в общем случае процессы обработки десятичных чисел строятся таким образом, что операнды подаются на обработку в АЛУ по частям. Рассмотрим особенности движения операндов в различных моделях на примере операции типа сложения.

При выполнении операций над десятичными числами операнды выбираются в порядке справа налево. Непосредственно в операции числа участвуют в упакованном формате выравненными по правым границам (по младшим десятичным цифрам). Точного совмещения младших цифр при выборе из основной оперативной памяти операндов можно было добиться при побайтном выборе. Однако даже в модели ЕС-1020 за одно обращение считывается два байта, причем обязательно левый из ячейки с четным, а правый — из ячейки с нечетным адресом. При этом может оказаться, что младшие байты обоих операндов будут иметь адреса либо одинаковой четности, либо различной.

В модели ЕС-1020 в связи с этим возможны два варианта циклических процессов сложения: один для случая, когда младшие цифры слагаемых расположены в байтах, имеющих оба либо четные, либо нечетные адреса, а второй, когда младший байт одного операнда имеет четный адрес, а другой — нечетный.

Оба цикла обеспечивают совмещенную подачу на обработку в арифметикологический блок одноименных байтов операндов, начиная с младших. Если пронумеровать байты в операндах справа налево, то можно сказать, что в результате совмещения вначале будут, допустим, складываться первые байты операндов, затем — вторые, потом — третьи и т. д. В отличие от сложения двоичных чисел с фиксированной запятой, при сложении двоично-десятичных кодов в АЛБ работают десятичные корректоры (см. п. 3.4). Десятичные числа одного знака складываются в прямом коде, а при разных знаках — один из операндов представляется в дополнительном коде.

В модели ЕС-1030 ширина выборки из основной оперативной памяти равна четырем байтам. Для сложения кодов в сумматоре в АЛУ подаются одновременно от каждого десятичного операнда также по четыре байта. Однако эти байты не поступают на выполнение операции непосредственно из ООП. Вначале, выбираясь по частям, операнды размещаются в рабочих регистрах местной оперативной памяти. Оттуда пары четверок их байтов друг за другом посылаются в АЛУ для производства над ними арифметических действий.

Из ООП в МОП операнды выбираются, начиная с байтов, содержащих младшие цифры десятичных чисел (справа налево). Каждая очередная выбираемая из ООП четверка байтов, проходя через АЛУ, может сдвигаться вправо. Эти сдвиги производятся для того, чтобы младшие байты операндов оказывались примыкающими к границам слов. Таким образом, в этой модели осуществляется выравнивание операндов по правым границам целых десятичных чисел.

Части операндов, по длине равные словам, складываются в сумматоре СМ (рис. 6.4). Для представления прямого кода 2-го операнда в коде с избытком 6 и для коррекции результата на входе регистра РА предусмотрены специальные

логические цепи. Собственно сложение кодов осуществляется также согласно алгоритму, рассмотренному в п. 3.4.

Результат операции слово за словом заносится в МОП. По окончании сложения двух операндов и анализа результата производится его запись в ООП. Если в начале операции операнды выравнивались, то перед отсылкой результата производится его необходимый сдвиг влево. Это делается для того, чтобы правильно поместить результат по адресу первого операнда. Знак к результирующему числу приформировывается сразу после сдвига последнего в АЛУ влево перед отсылкой слова, содержащего младший байт, в ООП.

В модели ЕС-1050 ширина выборки из ООП равна 8 байтам (2-м словам). Двойные слова 1-го и 2-го операндов из РЧ1 и РЧ2 БЦД принимаются в РВД1 и РВД2 БАЦ (см. рис. 6.3). В первых, выбранных в начале операции, двойных словах правые границы операндов могут оказаться в любых байтах. Поэтому адреса младших байт операндов, как и длины самих операндов, указываются специальными числами. Эти числа вместе с первыми частями операндов поступают в специальные счетчики блока БАЦ (на рис. 6.3 счетчики не показаны). Наличие такой информации и однобайтового десятичного сумматора исключает необходимость предварительного сдвига операндов для целей их выравнивания по правым границам. Требуемые очередные байты операндов выбираются в РСМ1 и РСМ2 из двойных слов, находящихся в РВД1 и РВД2, с помощью счетчиков адресов байтов. Результат сложения байтов в десятичном сумматоре из РРБ заносится в РВД1 по адресу, по которому был выбран в РСМ1 байт 1-го операнда.

Действия выборки пар байтов, их обработки и записи результирующего байта в нужное место РВД1 периодически повторяются до тех пор, пока либо не потребуется вызвать из ООП продолжение одного из операндов или обоих вместе, либо не будет установлено, что обработаны все байты 1-го и 2-го операндов. Если требуется принять в БАЦ продолжение 1-го операнда, то уже накопленные в РВД1 байты результата не отсылаются в ООП, а запоминаются в одном из регистров процессора. Первый операнд, а следовательно, и результат, в предельном случае длиной в 16 байт, могут располагаться в трех двойных словах. Это — в том случае, если правая граница операнда не совпадает с целочисленной границей двойного слова. Поэтому может возникнуть необходимость дважды вновь обращаться в ООП за продолжением 1-го операнда. При вызове третьей его части вторая часть результата также запоминается в одном из регистров процессора.

Запись результата в основную оперативную память производится по окончании сложения всех байтов операндов и определения знака результата. Так как при сложении кодов с разными знаками 2-й операнд преобразуется в дополнительный код, то в зависимости от величин слагаемых сумма может получиться как в прямом, так и в дополнительном коде. Признаком получения суммы в дополнительном коде является отсутствие переноса из ее старшего байта при разных знаках слагаемых. Перед отсылкой результата в ООП он из дополнительного кода должен быть преобразован в прямой. Это преобразование осуществляется вычитанием полученной суммы из нуля.

Знак суммы заносится в правую половину младшего байта непосредственно перед записью результата в ООП. После этого прямой код результата отсылается в основную оперативную память по адресу 1-го операнда. Если результат расположен в трех двойных словах, занимающих в процессоре три регистра, то запись в ООП происходит в три приема. Вначале из РВД1 посылаются в ООП младшая часть результата, а на ее место в РВД1 принимается средняя часть. Затем в ООП записывается средняя часть, а в РВД1 передается старшая часть и т. д.

Логические операции над полями данных. В таких операциях операнды рассматриваются как группы байтов. По полям операндов части этих групп выбираются из основной оперативной памяти и обрабатываются в процессоре слева направо. Размеры выбираемых частей зависят от ширины выборки ООП.

При большой длине операндов (от одного до 256 байт) проблемы выбора частей операндов и подачи их на обработку в АЛУ в основном те же, что в операциях десятичной арифметики.

Например, в модели ЕС-1050 вместе с двойными словами, в которых могут содержаться только начала операндов, в блок АЦ поступают адреса начальных байтов операндов и код их длины. В командах формата SS код длины — общий

для обоих операндов. В предельном случае 256-байтовый операнд может находиться в 33 ячейках, по длине равных двойным словам.

Побайтовая обработка операндов в модели ЕС-1050 происходит в логическом коммутаторе ЛК блока АЦ, в ЕС-1030 — в узле обработки байтов УОБ АЛУ, а в ЕС-1020 — в АЛБ. Если результат логических преобразований операндов должен быть помещен на место 1-го операнда, то в модели ЕС-1050, например, каждый результирующий байт из РРБ заносится в РВД1. При этом замещает он тот байт, который был послан для получения данного байта результата. По окончании обработки всех байтов 1-го операнда содержимое РВД1 отсылается в ООП, а в РВД1 вызывается его продолжение. Если окончилась обработка всех байтов части 2-го операнда, то в РВД2 также выбирается его продолжение из ООП. При выполнении операций над полями данных, в том числе и арифметических операций над десятичными числами, содержимое регистров общего назначения и регистров для операндов с плавающей запятой не меняется.

ЦЕНТРАЛЬНОЕ УСТРОЙСТВО УПРАВЛЕНИЯ

Под центральным устройством управления любой модели ЭВМ Единой системы понимается совокупность блоков и узлов процессора, обеспечивающих координирование работы всех устройств ЭВМ и управление ими для всех принятых режимов. Центральное устройство управления, реализуя программы математического обеспечения и рабочие программы, организует все необходимые действия по оценке и преобразованию исходной информации, по получению результирующих данных.

7.1 ОБЩИЕ ПРИНЦИПЫ УПРАВЛЕНИЯ
РАБОТОЙ МАШИН ЕДИНОЙ СИСТЕМЫ

Все модели ЭВМ Единой системы представляют собой цифровые вычислительные машины с хранимыми программами и прямой реализацией основных принципов программного управления. Решение любой задачи на них в конечном итоге сводится к последовательной выборке и выполнению команд соответствующей программы, что организуется центральным устройством управления (ЦУУ). В связи с этим ЦУУ может быть представлено как преобразователь первичной командной информации, представляемой командами программы, во вторичную командную информацию, представляемую формируемыми ЦУУ исполнительными адресами и управляющими сигналами, воздействие которых на соответствующие узлы и блоки устройств машин приводит к выполнению заданных операций. Помимо команд к первичной командной информации относятся также коды и сигналы, характеризующие состояния отдельных блоков и устройств.

Выполнение, или реализация, команды обычно проводится в следующей последовательности:

- выборка команды из оперативной памяти;
- формирование адресов операндов по информации, содержащейся в коде команды;
- выборка операндов из оперативной или местной памяти;
- выполнение действий в арифметических блоках;
- отсылка результата выполнения операции в оперативную или местную память.

Каждый из указанных этапов действий несколько отличается для команд различных форматов и различных моделей ЭВМ; отличия в выполнении поэтапных действий для различных моделей ЭВМ вызываются принимаемыми принципами построения процессоров, форматами слов на входе и выходе оперативной памяти, способами формирования управляющих сигналов. Однако во всех случаях можно выделить специальные блоки процессора или группы узлов, обеспечивающие выполнение действий по реализации команд и относящиеся к центральному устройству управления.

В машинах Единой системы используется в основном двух-адресная система команд, причем только для формата RR принята прямая адресация. Компоненты исполнительных адресов для случаев относительной адресации (форматы RX, RS, SI, SS), как и операнды для формата RR, размещаются в местной памяти, имеющей 16 ячеек для хранения 32-разрядных слов с фиксированной запятой и 4 ячейки для хранения 64-разрядных слов с плавающей запятой. Местная память функционально может быть отнесена к центральному устройству управления, хотя в моделях младших номеров (например, в ЕС-1020) она представляет собой часть оперативной памяти с начальными адресами; в моделях старших номеров местная память выполняется в виде набора регистров общего назначения.

В общем случае, когда все действия, связанные с преобразованием кодов команд в наборы управляющих сигналов и исполнительные адреса, выполняются непосредственно в центральном устройстве управления (ЦУУ), это устройство должно включать в свой состав (рис. 7.1) следующие блоки: блок выборки команд (БВК), блок центрального управления (БЦУ), блок адреса результата (БАР), блок местной памяти (БМП), блок сумматора адреса (БСА), блок выборки данных (БВД). Кроме этих блоков, обеспе-

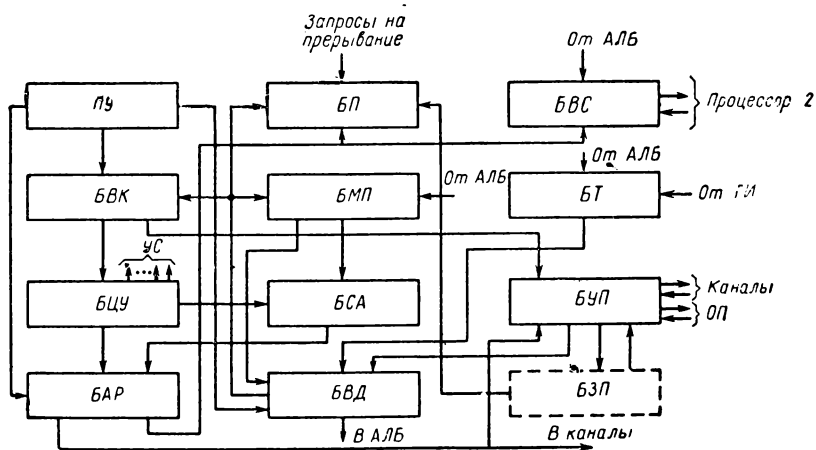


Рис. 7.1. Общая структура центрального устройства управления

чивающих преобразование и реализацию кодов команд, в состав ЦУУ обычно включаются: пульт управления (ПУ), блок прерываний (БП), блок управления памятью (БУП), блок таймеров (БТ) и блок внешних связей (БВС); иногда к центральному устройству управления относят и блок защиты памяти (БЗП), показанный на рис. 7.1 штриховой линией.

БВК служит для формирования адреса очередной команды (или очередного участка программы) с помощью обычного суммирующего счетчика и размещения кодов команд в своих регистрах. Известно, что команды могут быть двухбайтными (формат RR), четырехбайтными (форматы RX, RS и SI) и шестибайтными (формат SS). В то же время выборка информации из оперативной памяти в ЕС-1020 производится по полусловам (2 байта), в ЕС-1030 — по словам (4 байта), в ЕС-1050 — по двойным словам (8 байт). Таким образом, только в случае команд формата RR в ЕС-1020 и форматов RX, RS, SI в ЕС-1030 выбираемые из оперативной памяти коды представляют собой коды соответствующих команд. Во всех остальных случаях код, выбранный из оперативной памяти, представляет либо часть команды, либо несколько команд — от двух до четырех, т. е. участок программы.

Для подготовки кодов команд к преобразованию в БВК всегда производится анализ их первых двух разрядов, определяющих длину команд в байтах. Если реальная длина команды больше длины выбираемых из оперативной памяти кодов, то организуются дополнительные циклы обращения или один дополнительный цикл обращения к ОП для получения остальных частей команды. Так, в ЕС-1020 возможно трехкратное обращение к ОП для выборки полного кода команды формата SS. Если же реальная длина команды не превосходит длину выбираемых из ОП кодов, то производится простое выделение кода команды из многобайтного кода и размещение его в соответствующем регистре. Возможный остаток многобайтного кода дает начало очередной команды.

Отдельный блок выборки команд имеют центральные устройства управления моделей старших номеров. В моделях младших номеров, например, в ЕС-1020, функции БВК выполняют общие регистры процессора совместно с арифметическо-логическим блоком, обеспечивающим формирование адресов очередных команд. Это связано с тем, что в моделях младших номеров осуществлено совмещение функций узлов процессора с целью уменьшения общего аппаратурного состава.

Блок выборки данных служит для размещения кодов, принимаемых из оперативной памяти, местной памяти, блока таймеров, а также от пульта управления. Он включает в свой состав один или несколько регистров, информация из которых выдается в арифметическо-логический блок (АЛБ), блок выборки команд, а также в блок прерываний и блок местной памяти. Этот блок выполняется в различных моделях ЭВМ Единой системы аналогично блоку выборки команд.

В блоке адреса результата размещается адрес, по которому производится запись в память результата выполнения операции. Основу его составляет регистр адреса, выделяемый специально для этих целей или относящийся к общим регистрам процессора.

Блок сумматора адреса служит для формирования исполнительных адресов операндов по информации, содержащейся в коде команды. В предельном случае исполнительный адрес формируется путем сложения трех компонент: базового адреса, выбираемого из местной памяти, 12-разрядного кода смещения, содержащегося в коде команды, и индекса, или модификатора, выбираемого из местной памяти. Поэтому основу рассматриваемого блока при его выполнении в виде отдельной функциональной части составляет суммирующая схема, обеспечивающая сложение кодов трех чисел. Исполнительные адреса, формируемые в БСА, направляются в блок управления памятью либо непосредственно, либо через блок адреса результата.

Преобразование кода операции в набор основных управляющих сигналов (УС) производится в блоке центрального управления. В зависимости от принимаемых принципов формирования управляющих сигналов этот блок строится либо в виде микропрограммника, либо в виде дешифратора кода операций и схем формирования управляющих сигналов. Для моделей младших номеров по ЕС-1040 включительно принят микропрограммный принцип формирования управляющих сигналов, а для ЕС-1050 и ЕС-1060 — схемный принцип формирования УС. Более подробно эти принципы формирования управляющих сигналов и их реализация в различных моделях ЭВМ Единой системы рассматриваются ниже в пп. 7.2 и 7.3.

Пульт управления содержит переключатели, кнопки и средства индикации, обеспечивающие управление работой машины со стороны оператора, а также визуальный контроль состояния отдельных устройств и проведение профилактических работ.

Блок прерываний, функционально выделяемый практически во всех моделях ЭВМ Единой системы, обеспечивает реализацию запросов на прерывания в соответствии с их приоритетами. Отметим, что включение в состав центрального устройства управления блока прерываний характеризует машины Единой системы как машины третьего поколения.

Особое место в составе центрального устройства управления занимает блок управления памятью. Дело в том, что в машинах Единой системы непосредственный обмен информацией с оперативной памятью осуществляется не только процессором, но и всеми каналами. Для обработки заявок на обслуживание со стороны ОП и установления очередности их выполнения в соответствии с принятыми приоритетами в состав ЦУУ и включается блок управления памятью. Все данные, записываемые в ОП или выбираемые из нее, а также их адреса, поступающие из каналов или процессора, всегда проходят через блок управления памятью.

Блок внешних связей и блок таймеров относятся к так называемым системным средствам управления. Первый из них обеспечивает обмен управляющей информацией с другим процессором при построении и работе многопроцессорной вычислительной системы. Блок таймеров служит для подсчета временных интервалов и управления работой процессора в системе в соответствии с принимаемыми временными соотношениями.

Блок защиты памяти обеспечивает выявление случаев обращения к запрещенным для данной рабочей программы участкам, или страницам, оперативной памяти и организацию для них прерываний в реализации программ.

Рассмотренная общая структура центрального устройства управления является наиболее полной, отвечающей принципам организации управления работой машин Единой системы. Реализация же этой структуры различна для различных моделей ЭВМ вследствие отличия принимаемых уровней совмещения функциональных узлов и принципов формирования управляющих сигналов. При этом действует общий для всех цифровых вычислительных машин закон: чем выше уровень совмещения функциональных узлов, т. е. чем меньше аппаратурный состав, тем ниже при прочих равных условиях уровень совмещения отдельных действий при выполнении команд, т. е. тем меньше, в конечном итоге, быстродействие машины. Действие этого закона наглядно иллюстрируется возможностями моделей ЕС-1020 и ЕС-1030, имеющими общую элементную базу и устройства управления, реализующие принципы микропрограммного управления, но в 5 раз отличающимися по быстродействию.

Синхронизация. Отличительной особенностью всех моделей ЭВМ Единой системы является использование принципов жесткой синхронизации работы их процессорных частей. Для этого применяются либо отдельные блоки синхронизации, либо сложные системы, включающие в свой состав узлы генерирования, распределения и ретрансляции синхронизирующих сигналов.

В машине ЕС-1020, например, в состав центрального устройства управления включен отдельный блок синхронизации, в котором формируются пять серий синхроимпульсов, обеспечивающих совместную работу блоков процессора: исходные синхроимпульсы серии С, главные синхроимпульсы серии ГИ, рабочие синхроимпульсы серий ТИ, ХИ, СИ.

Каждая серия состоит из четырех импульсов; период следования этих импульсов составляет один машинный такт и равен 1 мкс. В каждой серии импульсы нумеруются с 1-го по 4-й, например, С1, С2, С3, С4.

Синхроимпульсы серии С вырабатываются непрерывно и используются для формирования главных и рабочих синхроимпульсов. Из импульсов серии С одновременно формируются главные импульсы ГИ и одна из трех рабочих серий ТИ, ХИ или СИ; тактовые импульсы ТИ формируются для выполнения микро-

команд, холостые импульсы ХИ — для изменения последовательности выборки микрокоманд, селекторные импульсы СИ формируются для организации обращения селекторных каналов к оперативной памяти. При одновременном появлении запросов на формирование импульсов рабочих серий всегда выдерживается следующая приоритетность: 1) СИ, 2) ХИ, 3) ТИ. Длительность, период следования и последовательность синхроимпульсов в пределах каждой серии одинаковы для всех серий и определяются только параметрами исходных синхроимпульсов С.

Формирование импульсов серии С происходит в схемах генератора исходных синхроимпульсов (ГС), включающего в свой состав задающий генератор (ЗГ), представляющий собой автогенератор с линией задержки (ЛЗ) в цепи обратной связи, одноразрядный двоичный счетчик (ТСЧ), триггер формирования длительности такта, или триггер функционального перехода (ТФП), и схему выработки синхроимпульсов серии С, как это показано на рис. 7.2. Генератор исходных синхроимпульсов несинхронизируется; он начинает формировать синхроимпульсы С1—С4 сразу же после подачи на него питающего напряжения; начало работы ГС не зависит от состояния триггера функционального перехода.

Существенную роль в формировании импульсов серии С по длительности и в организации их временного сдвига играет линия задержки с тремя выходами и общим входом. Задержка сигналов по тракту 1 обеспечивается на время порядка 200 нс, по тракту 2 — порядка 250 нс и по тракту 3 — порядка 300—350 нс. В обычном режиме, когда триггер ТФП находится в нулевом состоянии, работу задающего генератора определяет характер задержки сигналов по тракту 2 ЛЗ. Действительно, в этом случае на выходе элемента И—ИЛИ—НЕ задающего генератора смена уровней сигнала будет происходить через 250 нс. Поэтому можно сказать,

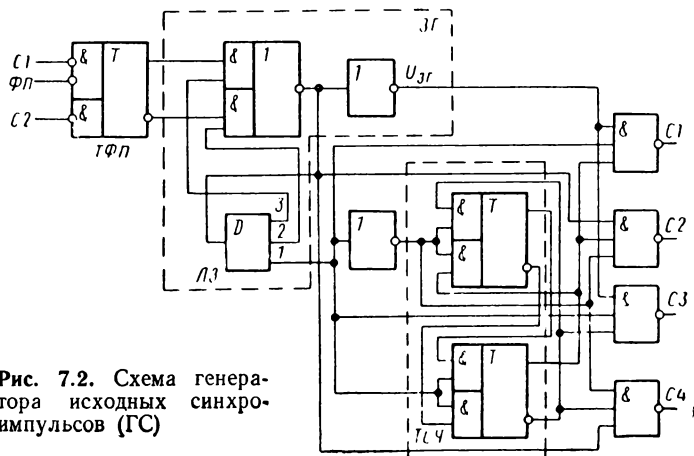


Рис. 7.2. Схема генератора исходных синхроимпульсов (ГС)

что в обычном режиме ЗГ формирует импульсы с длительностью $\tau_{зг} = 250$ нс и периодом $T_{зг} = 500$ нс.

Длительность синхроимпульсов С1—С4 равна 200 нс, а временной интервал между задним фронтом данного импульса серии и передним фронтом следующего импульса серии равен 50 нс; все импульсы серии С имеют основной период $T_c = 1$ мкс; этот период определяет в обычных условиях длительность одного машинного такта, как это показано на временной диаграмме рис. 7.3. Для формирования импульсов С1—С4 по длительности и их взаимного сдвига используются сигналы, образующиеся на выходе 1 линии задержки, а также сигналы, формируемые на выходах ТСЧ. Поскольку ТСЧ является делителем частоты на два, то на его выходах длительность сигналов равна 500 нс.

Выделение сигналов С1—С4 обеспечивается четырьмя элементами И—НЕ на три входа каждый. Сигналы с единичного выхода ТСЧ управляют формированием С1 и С2, а сигналы с нулевого выхода ТСЧ — формированием С3 и С4. Для того чтобы на выходе любого из четырех элементов И—НЕ начал действие один из импульсов серии С отрицательной полярности, необходимо совпадение по времени действия на входах элемента сигналов высокого уровня. Так, сигнал С1 начинает действовать, когда на входах первого элемента И—НЕ (см. рис. 7.2) одновременно действуют сигналы высокого уровня, снимаемые с выхода 1 линии задержки и единичного выхода ТСЧ.

С выхода генератора исходных синхроимпульсов сигналы С1—С4 подаются на специальные формирователи, работающие на кабель. Этим обеспечивается разводка синхроимпульсов серии С по всем панелям стойки процессора.

В некоторых случаях, например, при формировании адреса для постоянной памяти в случае функционального перехода, требуется расширить основной машинный такт до величины порядка 1,1 мкс. Для этого по сигналу С1 сигналом ФП триггер функционального перехода перебрасывается в единичное состояние; в этом состоянии ТФП находится до прихода очередного сигнала С2. Во время нахождения ТФП в единичном состоянии в кольцевую цепочку задающего генератора включается тракт 3 линии задержки. Вследствие этого очередной импульс на выходе ЗГ имеет длительность не 250 нс, а 300—350 нс. Временной интервал между очередными С1 и С2

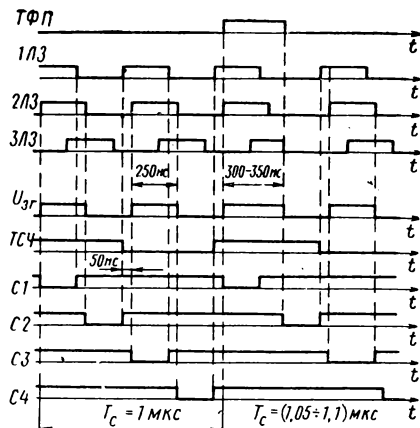


Рис. 7.3. Временная диаграмма работы генератора синхроимпульсов

возрастает до 100—150 нс, что и обеспечивает увеличение основного машинного такта до 1,05—1,1 мкс. При этом длительность всех импульсов серии С остается равной 200 нс, как это показано на временной диаграмме рис. 7.3.

Синхроимпульсы серии ГИ, ТИ, ХИ, СИ практически повторяют основные синхроимпульсы серии С. Схемы их выработки располагаются в каждой панели стойки процессора и управляются сигналами, формируемыми в соответствии с условиями выполнения команд. Все эти схемы практически одинаковы, что позволяет рассмотреть их на примере схемы для синхроимпульсов серии ГИ.

Схема, обеспечивающая непосредственное формирование синхроимпульсов серии ГИ, приведена на рис. 7.4. Она включает в свой состав четыре входных инвертора, так как сигналы С1—С4 имеют низкий уровень, четыре элемента И на два входа каждый и четыре выходных инвертора, поскольку сигналы ГИ1—ГИ4 в общем случае имеют тот же уровень, что и сигналы С1—С4. Формирование сигналов ГИ1 и ГИ2 происходит по разрешающему сигналу высокого уровня РГИ1, а формирование ГИ3 и ГИ4 — по разрешающему сигналу высокого уровня РГИ2.

В машине ЕС-1050 синхроимпульсы используются для жесткого тактирования работы запоминающих элементов типа триггеров и синхронизации работы логических схем. При этом для логических схем длительность информационных и управляющих сигналов принята кратной длительности основного такта, т. е. периоду синхроимпульсов. Для повышения надежности работы запоминающих элементов, включаемых последовательно, передача информации между ними осуществляется по синхроимпульсам, разнесенным по времени на величину, кратную длительности полутакта.

Синхронизация работы элементов в ЕС-1050 осуществляется централизованно, от единого задающего генератора. Для ретрансляции сигналов синхронизации в каждой стойке и каждой панели процессора имеется специальный ТЭЗ (типовой элемент замены).

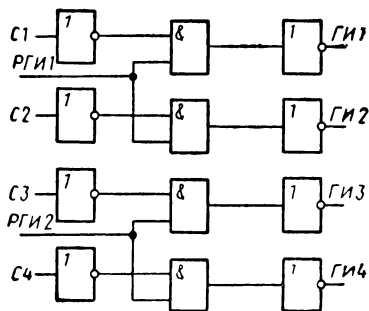


Рис. 7.4. Схема формирования импульсов серии ГИ

Задающий генератор непрерывно формирует в процессе нормальной работы машины серию С начальных синхроимпульсов с $\tau_{зг} = 40$ нс и $T_{зг} = 80$ нс, т. е. $F_{зг} = 12,5$ МГц. Одновременно он формирует вспомогательные синхросигналы СО, имеющие ту же длительность, но период, равный 320 нс. При проведении профилактических работ частота задающего генератора с помощью кнопок, расположенных на пульте управления, может изменяться на $\pm 10\%$. Для наладочных режимов задаю-

ший генератор переключается на работу с частотой 6,25 МГц; возможно изменение этой частоты на $\pm 20\%$.

Основными сигналами синхронизации являются сигналы серий С1 и С2, формируемые панельными ТЭЗами синхронизации по сигналам серии С, получаемым из задающего генератора. Эти сигналы имеют частоту, равную 6,25 МГц, при длительности $\tau_{C1} = \tau_{C2} = 40$ нс; длительность такта по основным синхроимпульсам $T_{СИ} = 160$ нс (см. временную диаграмму на рис. 7.5). Сигналы С0 используются для запуска счетчиков в панельных ТЭЗх синхронизации при работе в шаговом режиме, т. е. при работе в режиме выполнения действий по отдельным тактам.

Сигналы серий С1 и С2 сдвинуты друг относительно друга на полутакт, т. е. на половину периода их следования. Поэтому они непосредственно могут использоваться для синхронизации последовательно включенных триггеров. Однако при организации передачи информации по длинным цепочкам таких сигналов оказывается недостаточно. Необходимо использовать сигналы, имеющие ту же фазу и длительность, что и сигналы серий С1 и С2, но разнесенные по времени на четыре полутакта.

С этой целью в панельных схемах формирования синхросигналов по сигналам С1 и С2 формируются с помощью линий задержки и логических элементов управляющие сигналы У1, У2, У3, У4. Далее, с помощью четырех элементов И, на входы которых подаются пары С1 и У1, С2 и У2, С1 и У3, С2 и У4, формируются синхроимпульсы серий С11, С21, С12, С22, имеющие увеличенный по сравнению с сигналами серий С1 и С2 период следования.

Организация и совмещение циклов. Под циклом обычно понимается временной интервал, в течение которого выполняются действия по выборке и выполнению одной команды. Поскольку формат кодов, выбираемых из оперативной памяти моделей ЭВМ Единой системы младших номеров, меньше формата некоторых команд, то в течение цикла возможно неоднократное обращение к ОП только по выборке команды. Последовательность действий, выполняемых в течение цикла, зависит еще от уровня совмещения функциональных узлов процессора. Таким образом организация циклов зависит от характеристик процессоров и оперативных запомина-

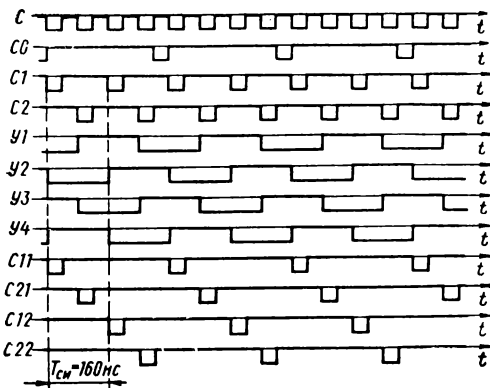


Рис. 7.5. Временная диаграмма сигналов синхронизации

ющих устройств моделей ЭВМ, а также от вида реализуемых команд.

При высоком уровне совмещения функциональных узлов процессора и представлении местной памяти в виде участка общей оперативной памяти, как это сделано в ЕС-1020, цикл выполнения команды оказывается растянутым и включает многократные обращения к оперативной памяти. Пусть выполняется команда формата RR, определяющая некоторую арифметическую операцию. Тогда общая последовательность действий внутри цикла оказывается такой:

1) формирование адреса текущей команды за счет прибавления к адресу предыдущей команды числа два (обмен информацией с ОП осуществляется двухбайтными кодами) в арифметическо-логическом блоке (АЛБ);

2) выборка команды;

3) анализ и выделение адреса первого операнда;

4) выборка первого операнда;

5) выделение адреса второго операнда;

6) выборка второго операнда;

7) выполнение в АЛБ заданной операции;

8) запись результата.

Очевидно, что даже при выполнении весьма простой команды требуется не менее 4 раз в течение цикла производить обращение к оперативной памяти; если операнды и результат выполнения операции представляются четырехбайтными словами, то в течение цикла необходимо 7 раз обращаться к оперативной памяти. При выполнении команд формата RX количество обращений к оперативной памяти в течение цикла может возрасти до 12 и более, а в АЛБ за это же время не менее 5 раз выполняется операция сложения. Все действия должны выполняться последовательно в одних и тех же блоках и устройствах, поэтому совмещение циклов невозможно.

Совмещение действий внутри циклов и самих циклов возможно тогда, когда выполнение каждого основного этапа реализации команды обеспечивается отдельным функциональным узлом или блоком, а местная память выполнена в виде набора регистров общего назначения. Такая возможность имеется, например, в машине ЕС-1050, центральное устройство управления которой выполнено по схеме, близкой к схеме рис. 7.1, при выделении в каждом блоке ЦУУ достаточного количества функционально независимых узлов; практически обеспечивается совмещение действий по выполнению трех последовательных команд.

Применительно к случаю совмещения трех циклов действия по выполнению одной команды (инструкции процессора) разделяются на три уровня:

выборка участка программы из оперативной памяти и выборка инструкции из этого участка с размещением ее в регистре команд (распаковка участка);

формирование адресов операндов и выборка операндов из местной или оперативной памяти с размещением их в специальных регистрах ЦУУ;

выполнение действий над операндами в АЛБ и запись результата операции в местную или оперативную память.

Каждому уровню отвечает функционально независимые узлы и блоки ЦУУ. Так, первому уровню, или уровню К, отвечает в ЦУУ блок выборки команд, второму уровню, или уровню Ч, — блок сумматора адреса, блок центрального управления и блок выборки данных, а третьему уровню, или уровню Т, — блок адреса результата совместно с АЛБ.

Реализация инструкций на всех указанных уровнях производится по «плавающим» подциклам, т. е. без жесткой временной регламентации работ на каждом уровне. Основные блоки уровней, т. е. БВК, БЦУ и БАР, имеют отдельные счетчики тактов, работающие независимо и определяющие продолжительность действий на каждом уровне в соответствии с особенностями выполняемой инструкции. Минимальный подцикл обработки инструкции на одном уровне равен 6 полутактам или 0,48 мкс. Минимальные подциклы обработки инструкций одновременно на всех уровнях возможны только при реализации команд формата RR.

Основные блоки уровней К, Ч, Т имеют в своем составе триггеры занятости и готовности, регламентирующие последовательность обработки инструкций с переходом от предыдущего уровня к последующему. Работа по реализации инструкции на данном уровне может начинаться только тогда, когда триггер занятости этого уровня находится в нулевом состоянии, что свидетельствует о незанятости уровня работой или хранением информации, а триггер готовности предыдущего уровня находится в единичном состоянии, что свидетельствует о завершении обработки информации, необходимой для рассматриваемого уровня, на предыдущем уровне.

Триггеры занятости и готовности уровней устанавливаются в соответствующие состояния сигналами от счетчиков тактов, а также внешними по отношению к основным блокам уровней сигналами. К внешним относятся: сигналы сопровождения информации из блока управления памятью, сигналы окончания операций в АЛБ, сигнал отключения каналов и сигнал окончания записи в оперативную память. Так, например, при обработке инструкций формата RX триггер готовности уровня Ч устанавливается в единичное состояние сигналом сопровождения, поступающим из блока управления памятью одновременно со вторым операндом. Это объясняется тем, что к этому моменту времени первый операнд уже выбран из местной регистровой памяти и размещен в соответствующем регистре блока выборки данных.

Принципы совмещения циклов при обработке инструкций по трем уровням иллюстрируются диаграммой, приведенной на рис. 7.6. Эта диаграмма построена применительно к случаю после-

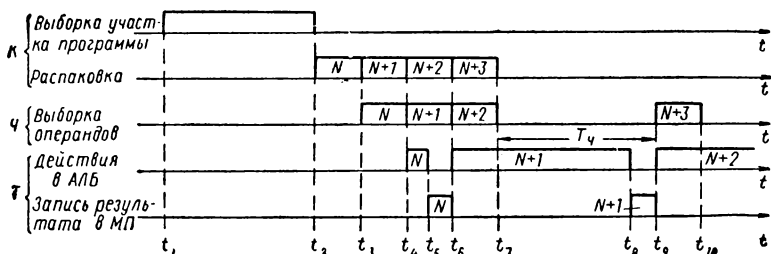


Рис. 7.6. Диаграмма совмещения циклов

довательной обработки инструкций формата RR. На временном интервале от t_1 до t_2 производится выборка участка программы; очевидно, что восьмибайтный код, принимаемый из ОП, содержит четыре инструкции формата RR. Распаковка, т. е. выделение инструкций, начинается с момента времени t_2 ; инструкции на рис. 7.6 обозначены номерами N , $N + 1$, $N + 2$, $N + 3$. В рассматриваемом случае уровень Ч не сдерживает работу уровня К, где распаковка четырех инструкций производится без приостановок.

В момент времени t_3 начинается и к t_4 заканчивается выборка операндов по инструкции N . Поэтому с t_4 работы производятся на всех трех уровнях: на уровне К производится распаковка инструкции $N + 2$, на уровне Ч производится выборка операндов по инструкции $N + 1$, на уровне Т — выполнение операции и запись результата в местную память МП. С момента времени t_7 работа производится только на уровне Т. В момент времени t_8 заканчивается работа на уровне Т по инструкции $N + 1$; на этот уровень передаются операнды по инструкции $N + 2$, а на уровне Ч после простоя, равного $T_ч$, начинается выборка операндов по инструкции $N + 3$.

7.2. МИКРОПРОГРАММНОЕ УПРАВЛЕНИЕ

Формирование управляющих сигналов в центральном устройстве управления осуществляется в соответствии с принципами микропрограммного управления тогда, когда основные УС вырабатываются за счет последовательного считывания и расшифровки информационных слов, располагающихся в ячейках отдельного постоянного ЗУ, или постоянной памяти (ПП). Каждое такое слово называется *микрокомандой* и содержит информацию о микрооперациях, выполняемых в течение одного машинного такта, а также информацию, обеспечивающую формирование адреса очередной микрокоманды.

Микрооперация — это элементарная операция, которая выполняется при подаче одного управляющего сигнала в управляемую цепь; например, сброс регистра в нулевое состояние, передача информации из одного регистра в другой и т. п.

Последовательность микрокоманд, обеспечивающих выполнение одной машинной команды или отдельной процедуры, называется *микропрограммой*. Для реализации микропрограмм в каждом такте производится выборка одной микрокоманды и ее преобразование в набор управляющих сигналов, действующих, как правило, одновременно. Каждая микрокоманда обычно разделяется на ряд частей, или полей, преобразуемых в управляющие сигналы автономно; собственно преобразование обеспечивается с помощью дешифраторов.

Принципы построения и работы микропрограммного устройства управления рассмотрим на примере машины ЕС-1020, в которой оно обеспечивает работу арифметическо-логического блока, выборку и запись информации в оперативную память, анализирует запросы на прерывания и выполняет процедуру прерывания. Кроме того, это устройство осуществляет связь процессора с каналами и обеспечивает реализацию пультовых процедур.

Структура машины. Основные функциональные связи. Микрокоманды разделяются на поля в соответствии с принятым делением машины на функциональные узлы и блоки, связями между ними, принципами выполнения действий по реализации инструкций процессора. Именно с этих позиций и рассматривается общая структура ЕС-1020, а также функциональные связи между ее узлами и блоками, иллюстрируемые схемой, приведенной на рис. 7.7.

Центральное устройство управления в качестве своих основных составных частей содержит блок управления (БУ), постоянную память (ПП) микропрограмм, пульт управления (ПУ) и блок защиты памяти (БЗП). Основной его функцией является формирование управляющих сигналов (УС) в соответствии с ха-

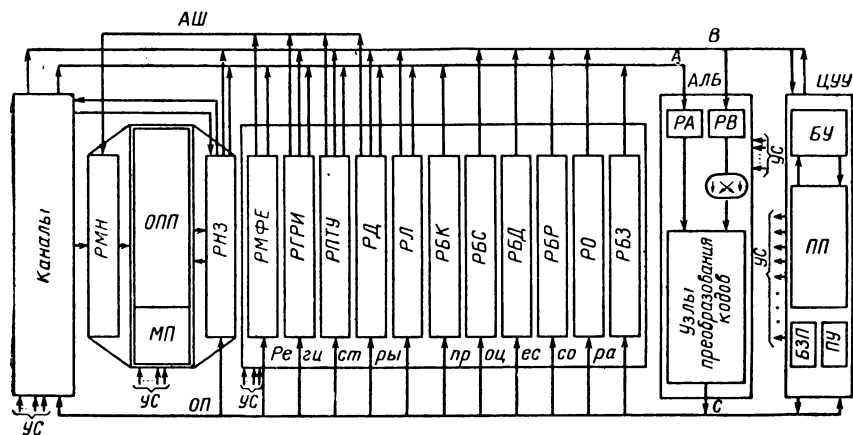


Рис. 7.7. Основные функциональные связи в машине с микропрограммным управлением (ЕС-1020)

рактором выполняемых операций и процедур. Коды операций поступают в ЦУУ по общим информационным шинам В. Посредством этих шин, а также общих информационных шин С центральное устройство управления имеет возможность обмениваться информацией с другими узлами и блоками машины.

Арифметическо-логический блок (АЛБ) оперирует только с однобайтными операндами, поэтому практически все операции в нем выполняются за несколько последовательных этапов. Входными для АЛБ являются регистры РА и РВ, связанные с информационными шинами А и В соответственно. Полубайты из РВ в узлы преобразования кодов могут передаваться обычным порядком или со сменой мест, что на рисунке условно обозначено набором стрелок. Образующиеся на выходе АЛБ байты результатов выполнения операций выдаются в информационные шины С.

В оперативной памяти машины выделены собственно оперативная память, или основная оперативная память (ООП), и местная память (МП), представляющая собой группу ячеек с начальными адресами. Регистром адреса оперативной памяти является 18-разрядный регистр РМН. Код адреса вводится в этот регистр по адресным шинам АШ; при обращении к местной памяти используется только младший байт адреса; при работе ОП с каналами адреса в РМН поступают из соответствующих регистров каналов. В качестве регистра числа ОП используется двухбайтный регистр РНЗ, связанный с информационными шинами А, В и С, а также с каналами. Отметим, что при всех передачах информации каждый информационный байт сопровождается контрольным разрядом.

Центральное устройство управления не содержит в своем составе узлов и блоков для преобразования управляющих слов (типа счетчика команд, сумматора адреса и т. п.). Любые преобразования информации осуществляются только в АЛБ. Для хранения и выдачи информации, необходимой в процессе функционирования машины, используются регистры процессора, объединяемые в отдельный блок. Сюда относятся адресные регистры (РМФЕ, РГРИ, РПТУ), общие регистры (РЛ и РД), а также служебные регистры, выделенные на рисунке. Все служебные и общие регистры — однобайтные.

Ввод информации в любой из указанных регистров процессора производится с информационных шин С, связанных с выходом арифметическо-логического блока. Выдача информации из регистров производится в соответствии с ее назначением по отдельным управляющим сигналам. Адресные регистры практически представляют собой объединение трех регистров: двух однобайтных (8 информационных разрядов + 1 контрольный разряд) и одного трехразрядного, хранящего значения так называемых разрядов расширения. Например, РМФЕ состоит из регистра РМ (разряды расширения), регистра РФ (старший

байт) и регистра РЕ (младший байт); иначе говоря, можно рассматривать РМФЕ как совокупность РМ, РФ и РЕ.

Адресные регистры предназначены для хранения адресов команд и операндов; в отдельных случаях они могут использоваться для размещения другой информации, например, для размещения операндов. Изменение информации в адресных регистрах осуществляется по микрооперации соответствующих полей микрокоманд.

Регистр РМФЕ служит главным образом для хранения адреса текущей команды; вместе с АЛБ он выполняет функции своеобразного счетчика команд. Если этот регистр используется для размещения операндов, то адрес текущей команды записывается в фиксированные ячейки местной памяти: (РМ) по адресу 8Д, (РФ) по адресу 8Е и (РЕ) по адресу 8F; в этом случае триггер адреса команды (ТАК) устанавливается в состояние 1. Если адрес текущей команды находится в регистре РМФЕ, то этот триггер находится в состоянии 0. Ввод информации в части РМФЕ производится через автономные группы вентиляй. Выдача информации через шины В производится по частям, т. е. отдельными байтами; выдача адреса в РМН через адресные шины АШ производится всегда полноразрядным кодом.

Адресные регистры РГРИ и РПТУ служат для хранения адресов операндов. Они выполнены по одинаковым схемам и имеют идентичные связи. Ввод информации в эти регистры осуществляется так же, как и в адресный регистр РМФЕ. Выдача информации из регистров РГРИ и РПТУ производится подобно случаю регистра РМФЕ. Но в отличие от регистра РМФЕ байты из частей РТ и РУ регистра РПТУ или из частей РР и РИ регистра РГРИ могут выдаваться не только в регистр РВ арифметическо-логического блока, но и в регистр РА. Кроме того, для задания адресов местной памяти содержимое части РР (или части РТ) может непосредственно передаваться в младшие разряды регистра РМН; при этом содержимое старших разрядов регистра РМН не изменяется.

Общие регистры предназначены для хранения и выдачи информации, участвующей в процессе обработки микропрограмм. Регистры РД и РЛ, относящиеся к общим регистрам, являются однобайтными и выполнены по одинаковым схемам. Прием информации в эти регистры осуществляется с выхода С арифметическо-логического блока; выдача информации из них через автономные группы вентиляй может производиться как на информационные шины А, так и на информационные шины В. Для организации адреса местной памяти содержимое регистра РД, кроме того, через отдельную группу вентиляй может выдаваться в младшие разряды регистра РМН.

Служебные регистры представляют собой восьмиразрядные объединения триггеров, в которых фиксируется информация о состояниях различных блоков машины, состоянии выполняемой

программы, а также о возникновении ошибок в процессе обработки данных. Эта служебная информация на определенных этапах работы машины анализируется с целью принятия решения о последующих действиях; анализ служебной информации производится автоматически при реализации соответствующих микрокоманд в микропрограммах выполнения операций. Объединение триггеров в регистры произведено в основном с целью обеспечения удобства пересылки служебной информации по байтам в другие блоки машины, а также ввода необходимой информации в триггеры, предназначенные для хранения служебных данных.

Все служебные регистры имеют полноразрядные группы входных и выходных вентилях, через которые может приниматься информация с выхода С арифметическо-логического блока и выдаваться информация в регистры РА и РВ этого блока. Кроме того, отдельные триггеры служебных регистров имеют цепи установки их в состояние, соответствующие ситуациям, возникающим в процессе работы машины. Правильность передачи информации из служебных регистров в РА (РВ) всегда контролируется; формирование контрольного разряда производится путем дополнения основных восьми разрядов до нечетного числа единиц.

Регистр РБК служит для хранения запросов на внешние прерывания. Триггеры разрядов со 2-го по 7-й хранят запросы на прерывания, поступающие от внешних объектов по шести отдельным линиям. Сигналы от таймера и кнопки прерывания на пульте управления заносятся соответственно в 0-й и 1-й разряды. Наличие запроса на прерывание отражается единственным состоянием соответствующего триггера. Установка запроса на прерывание от таймера осуществляется микропрограммно путем пересылки в 1-й разряд РБК единицы из АЛБ. Запросы на остальные внешние прерывания устанавливаются аппаратно, т. е. путем передачи сигналов непосредственно на входы соответствующих триггеров.

Регистр РБР служит для хранения кода маски прерываний и запросов на ввод-выводные прерывания. Разряды 0, 1, 2 и 7-й регистра хранят соответственно маски мультиплексного, первого селекторного, второго селекторного каналов и маску внешних прерываний; эти одноразрядные маски составляют маску системы. Единичное состояние соответствующего триггера указывает на то, что прерывание разрешено, а нулевое — запрещено (замаскировано). Разряды 3, 4 и 6-й хранят соответственно запросы на прерывания от мультиплексного, первого селекторного и второго селекторного каналов; единичное состояние соответствующего триггера указывает на наличие запроса на прерывание. Пятый разряд регистра хранит маску контроля машины;

единица в этом разряде указывает на то, что прерывание по сбою разрешено, а нуль — на то, что это прерывание замаскировано. Все триггеры рассматриваемого регистра управляются

микропрограммно, т. е. их состояния изменяются путем засылки в них информации из АЛБ.

Регистр РБС служит для хранения признаков сбоев по адресации и защите, универсальных указателей и кода условия, отражающего результат выполнения действий в арифметическо-логическом блоке. Разряд РБС [0], т. е. 0-й разряд этого регистра, устанавливается в единичное состояние аппаратно при обнаружении ошибки по адресации при обращении к оперативной памяти. Разряд РБС [1] устанавливается в единичное состояние также аппаратно при нарушении защиты памяти. Разряды РБС [2—5] являются универсальными указателями (индикаторами), используемыми для организации ветвлений в микропрограммах; установка триггеров этих разрядов в единичное и нулевое состояния осуществляется микропрограммно. Разряды РБС [6—7] хранят код условия, который устанавливается в соответствии с результатами выполнения действий в АЛБ.

Регистр РБД служит для хранения информации, отражающей состояние различных блоков машины. Так, единичное состояние триггера нулевого разряда РБД указывает на наличие запроса на микропрограммную приостановку от одного из каналов, нулевое — на работу процессора. Единичное состояние триггера 1-го разряда РБД указывает на выполнение процессором последовательности действий по загрузке начальной программы. Единичное состояние триггера 2-го разряда РБД указывает на то, что процессор находится в ждущем состоянии. Управляется этот триггер микропрограммно, а его состояние определяется 14-м разрядом текущего слова состояния программы.

Триггер 3-го разряда РБД находится в единичном состоянии во время выполнения микропрограммы гашения системы. Сброс и установка этого триггера производятся микропрограммно. Установка триггера 4-го разряда РБД в единичное состояние влечет за собой переход процессора в состояние останова перед выборкой очередной команды. Это, например, производится при нажатии на пульт управления клавиши «ОСТАНОВ».

Триггер 5-го разряда РБД находится в единичном состоянии во время выполнения пультовых микропрограмм чтения, записи и занесения адреса команды. Триггер 6-го разряда РБД обеспечивает занесение в регистр адреса постоянной памяти микропрограмм фиксированного адреса первой микрокоманды микропрограммы обработки прерывания по сбою машины. Единичное состояние триггера 7-го разряда РБД свидетельствует о том, что схемами контроля выявлен сбой в работе машины.

Фиксация сигналов о сбоях производится в регистре ошибок (РО); регистр РБЗ является информационным для блока защиты памяти; эти регистры также относятся к числу служебных регистров.

Состав и работа блока управления. Блок управления является частью микропрограммного ЦУУ, в состав которого еще обяза-

тельно включается постоянное запоминающее устройство, или постоянная память. Он служит для формирования управляющих сигналов и формирования адреса следующей микрокоманды; начало каждой микропрограммы выполнения машинной операции связывается с кодом операции, выделяемым из кода команды. Все микропрограммы хранятся в постоянной памяти, с которой в процессе работы непрерывно взаимодействует блок управления.

Собственно блок управления включает в свой состав дешифраторы полей микрокоманд (ДШ), узел занесения кодов в регистр адреса постоянной памяти (УЗАН РАПП), узел формирования адреса микрокоманды (УФАМ), регистр возврата селекторный (РВС) и регистр возврата мультиплексный (РВМ). Структура блока показана на рис. 7.8, отражающем общую структуру микропрограммного устройства управления. На этом рисунке в составе постоянной памяти выделены только накопитель, регистр адреса (РАПП) и регистр числа, или регистр микрокоманды (РМК).

Дешифраторы полей микрокоманд служат для непосредственного преобразования соответствующих частей микрокоманд в управляющие сигналы. В узле занесения кодов в регистр адреса ПП формируется адрес следующей микрокоманды, а управление формированием осуществляется узлом формирования адреса микрокоманды.

В РВС и РВМ запоминается адрес микрокоманды, выполнение которой задерживается из-за выполнения микропрограмм обслуживания селекторного или мультиплексного каналов. После завершения этих микропрограмм содержимое РВС или РВМ передается в регистр адреса постоянной памяти.

Работа блока управления протекает в следующей последовательности. Сформированный адрес микрокоманды передается в РАПП. Выбранная по этому адресу микрокоманда заносится в РМК. Поля микрокоманды, находящейся в РМК, дешифрируются с помощью ДШ блока управления; образуется необходимый набор управляющих сигналов, вызывающий соответствующие действия в течение одного такта работы машины. Общая

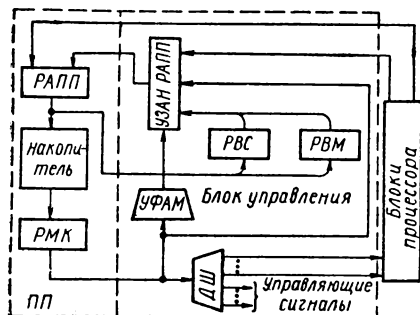


Рис. 7.8. Структура микропрограммного устройства управления

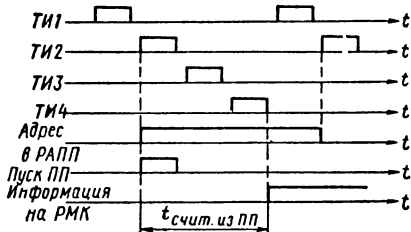


Рис. 7.9. Временная диаграмма работы устройства управления

последовательность действий по выборке и преобразованию одной микрокоманды отражена на временной диаграмме (рис. 7.9).

Одновременно с дешифрированием полей текущей микрокоманды происходит формирование адреса следующей микрокоманды. В формировании адреса при различных условиях участвуют: определенные поля текущей микрокоманды, содержимое РВМ и РВС, код адреса микрокоманды, набранный на пульте управления, содержимое четырех старших разрядов РВ арифметическо-логического блока. Все основные действия определяются кодами полей выполняемых микрокоманд.

Структура микрокоманды. Микрооперации. Микрокоманда представляет собой 64-разрядное двоичное слово, разделяемое на ряд полей. Каждое поле определяет до 2^m микроопераций, если m — разрядность поля; за один машинный такт выполняется только одна микрооперация для данного поля. Поля, по которым формируются управляющие сигналы, имеют не больше пяти разрядов, так что каждое поле определяет не более 32 различных микроопераций.

Поля микрокоманд объединяются в следующие группы: управление арифметическо-логическим блоком, управление оперативной памятью, занесение в триггеры и регистры, организация адреса микрокоманды, контроль, резерв.

Все эти группы и отдельные поля сведены в таблицу, представленную на рис. 7.10. Для всех возможных кодов каждого поля в таблице даны сокращенные обозначения соответствующих микроопераций. Отметим, что при пользовании таблицей необходимо, если $m < 5$, в кодах полей, представленных в первом столбце, отбрасывать $(5 - m)$ старших разрядов. Так, если $m = 4$, то в кодах полей читаются (используются) четыре младших разряда, и т. д.

Управление АЛБ. К этой группе относятся поля А, ФУНКЦИЯ, В, ДЕФОРМАЦИЯ В, управляющие работой арифметическо-логического блока. Четырехразрядное поле А определяет источник информации, вводимой во входной регистр А арифметическо-логического блока. В качестве источников информации используются, как правило, регистры процессора, которые на рис. 7.10 отмечены своими сокращенными обозначениями; используются также и другие регистры. По коду 1001 поля А в РА вводится константа К выполняемой микрокоманды, т. е. содержимое РМК [44—52]. При коде 0000 содержимое РА не изменяется. При коде 1111 в РА заносятся нули.

Четырехразрядное поле ФУНКЦИЯ определяет микрооперацию, выполняемую арифметическо-логическим блоком. При всех значениях кодов этого поля, кроме 0001, в АЛБ выполняется одна из 15 микроопераций, заданная составлением этого поля; считается, что в этих случаях выполняется прямая функция. При коде 0001 выполняется микрооперация по так называемой косвенной функции; эта микрооперация задается заранее путем

Код	Поля																																			
	С	Управление АЛБ										Память							Установ	УСЛ1	УСЛО	М	КР1	КСТ	АЛСТ	КМЛ	КМЛ1	КМЛ2	КР2							
		А	Функция	В	Деформация	Адрес	Режим	Тит	28	32	33	36	37	41	42	43																				
00000	0	4	5	8	9	12	13	17	18	20	21	23	24	25	26	27	28	32	33	36	37	41	42	43	44	45	46	52	58	59	60	67	68			
00000				А ∨ В		Прямо																														
00001	РН	РН	Вып КФ	РН	Накрест	РМФЕ	ЧТ	0																												
00010	РЗ	РЗ	А-В (дес)	РЗ	Ил.прямо	РГМИ	ЭП	РГ																												
00011	РЛ	РЛ	А-В (об)	РЛ	Ст.прямо	ВЛПУ	СТ	М																												
00100	РД	РД	А транзит	РД	Ил.накрест	РР																														
00101	РТ	РТ	А ∨ В	РТ	Ил.накрест	РГ																														
00110	РЧ	РЧ	В-А (дес)	РЧ	Перекоп	РД																														
00111	РР	РР	В-А (об)	РР	Перекоп	КФ																														
01000	РИ	РИ	А ∨ В	РИ																																
01001	РМ	К	В транзит	К																																
01010	РФ	РМ	А ⊕ В	РФ																																
01011	РГ	РГ	А ∨ В	РГ																																
01100	РЧ	РП	Сдвиг РВ10 вл	Р0																																
01101	РГ	РБК	Сдвиг РВ10 вл	РБД																																
01110	РП	РБЗ	А+В (дес)	РБС																																
01111	РБД	РА: = 0	А+В (об)	РБР																																
10000	РБС			РРЗ																																
10001	РБР			РР4																																
10010	РБК			РР5																																
10011	РБЗ			РР6																																
10100	РР1			РР9																																
10101	РР2			РР6																																
10110	РР5			РРВ																																
10111	РР6			РРГ																																
11000	РР7			РРД																																
11001	РР8			РРЛ																																
11010	РР9			РР: = 0																																
11011	РА			РБА																																
11100				РРБ																																
11101				РРП																																
11110																																				
11111																																				

Рис. 7.10. Структура микрокоманд

занесения необходимого кода в отдельный регистр косвенной функции (РКФ). Для занесения кода в РКФ в поле УСТАНОВ одной из предшествующих микрокоманд должен быть код, определяющий микрооперацию ЗКФ, т. е. ЗАНЕСТИ КОСВЕННУЮ ФУНКЦИЮ. По этой микрооперации в РКФ записывается содержимое поля КМЛ микрокоманды с микрооперацией ЗКФ. Таким образом, при коде 0001 рассматриваемого поля в АЛБ выполняется микрооперация, заданная заранее, т. е. КОСВЕННАЯ ФУНКЦИЯ.

Пятиразрядное поле В микрокоманды определяет источник информации, вводимой в РВ арифметическо-логического блока. В качестве источников информации используются регистры процессора; кроме того, в РВ может вводиться константа К, т. е. содержимое РМК [44—52], а также байт состояния арифметическо-логического блока БА. При коде 00000 содержимое регистра РВ не изменяется. При коде 11010 в РВ заносятся нули.

Трехразрядное поле ДЕФОРМАЦИЯ В определяет способы передачи информации со входного регистра РВ в узлы обработки данных, или узлы преобразования кодов, АЛБ; содержимое РВ при всех передачах информации не изменяется. Восемьразрядный байт информации в РВ разбивается на две тетрады: старшую и младшую. Микрооперация ПРЯМО вызывает передачу тетрад обычным порядком. При микрооперации НАКРЕСТ старшая тетрада передается на место младшей, а младшая — на место

старшей. Если выполняется микрооперация МЛАДШИЕ ПРЯМО, то на место старшей тетрады подаются нули, а младшая тетрада передается без изменений. В случае СТАРШИЕ ПРЯМО на место младшей тетрады подаются нули, а старшая тетрада передается без изменений.

По микрооперации МЛАДШИЕ НАКРЕСТ младшая тетрада передается на место старшей, а на место младшей подаются нули; по микрооперации СТАРШИЕ НАКРЕСТ старшая тетрада подается на место младшей, а на место старшей подаются нули. При выполнении микрооперации ПЕРЕКОС младшая тетрада передается на место старшей, а старшая запоминается в четырехразрядном буферном регистре; на место младшей передается прежнее содержимое буферного регистра.

При коде 111 рассматриваемого поля осуществляется предварительное задание режима «Перекося косвенной функции». После этого соответствующие действия выполняются всякий раз, когда в поле ФУНКЦИЯ задается микрооперация ВЫПОЛНИТЬ КФ.

Управление оперативной памятью. Поля этой группы АДРЕС, РЕЖИМ, ТИП определяют работу оперативной памяти, которая в ЕС-1020 функционально разделена на основную, местную, или локальную, и мультиплексную.

Трехразрядное поле АДРЕС определяет регистр, содержимое которого заносится в адресный регистр оперативной памяти. При коде 000 адрес в этот регистр не подается, а при коде 111 в него вводится константа К. Содержимое регистров РР, РТ, РД и константа передаются в младшие разряды адресного регистра, т. е. используются для адресации ячеек локальной памяти.

Двухразрядное поле РЕЖИМ определяет вид обращения к оперативной памяти: ЧТ — чтение (считывание); ЗП, РГ — запись, регенерация; СТ — стирание. При коде 00 обращение к оперативной памяти не производится.

Двухразрядное поле ТИП определяет тип памяти; З — память блока защиты, О — основная память, Л — локальная память, М — мультиплексная память.

Занесение в триггеры и регистры. Эта группа включает в свой состав поле УСТАНОВ и поле С.

Пятиразрядное поле УСТАНОВ определяет микрооперации установки различных служебных регистров в состояния 0 и 1, а также некоторые специальные микрооперации. В таблице, поясняющей структуру микрокоманд, обозначение ТПКФ: = 1, например, указывает, что при данном коде поля УСТАНОВ триггер признака косвенной функции устанавливается в состояние 1. При коде 00000 изменение состояний служебных триггеров не производится. При коде 00001 гасятся, т. е. устанавливаются в состояние 0, триггеры, хранящие байт состояния арифметическо-логического блока.

Микрооперация ИГН используется в случае неучета межбайтного переноса, т. е. тогда, когда этот перенос игнорируется.

Микрооперация ЗКФ (занесение косвенной функции) обеспечивает занесение кода микрооперации, которая будет в последующем выполняться в АЛБ, из поля КМЛ в регистр косвенной функции и сброс предварительно заданного режима перекоса. По микрооперации СБП осуществляется сброс буферного регистра перекоса; микрооперация ПСО используется для организации состояния останова. КУ1 и КУ2 — коды условий, при которых по определенному алгоритму формируются значения РБС [6] и РБС [7].

Пятиразрядное поле С определяет приемник информации с выхода С арифметическо-логического блока, т. е. регистр, в который передается код из АЛБ. При коде 00000 информация с выхода С этого блока никуда не передается.

Организация адреса микрокоманды. Поля этой группы УСЛ1, УСЛО, М, КСТ/АПСТ, КМЛ и АПМЛ используются для формирования адреса следующей микрокоманды.

Четырехразрядное поле УСЛ1 определяет состояние, в которое устанавливается триггер 1-го разряда регистра адреса постоянной памяти, т. е. РАПП [1]. При коде 0000 в РАПП [1] заносится код 0, при коде 0001 — код 1. При остальных кодах занесение в РАПП производится по следующему правилу: если соответствующий данному коду поля служебный триггер находится в заданном (определенном) состоянии, то в РАПП [1] заносится код 1; в противном случае — код 0. Так, например, если при коде 1000 триггер результата косвенной функции находится в состоянии 0, что в таблице отмечено условными обозначениями как $ТРКФ = 0$, то в РАПП [1] заносится код 1, и т. д.

Пятиразрядное поле УСЛО определяет состояние, в которое устанавливается РАПП [0]. При коде 00000 в РАПП [0] заносится код 0, при коде 00001 — код 1. Для остальных кодов занесение в РАПП [0] осуществляется по такому же правилу, как и для РАПП [1].

Двухразрядное поле М определяет способ формирования адреса следующей микрокоманды: АД — адрес перехода длинный; АК — адрес перехода короткий; АД/В — адрес перехода длинный или сброс РАПП; Ф, ВС, ВМ, ГРИ — функциональный переход, передача в РАПП содержимого РВС, РВМ и РГРИ.

Содержимое полей КСТ/АПСТ, КМЛ и АПМЛ непосредственно передается в РАПП (как составляющая адреса следующей микрокоманды) и в регистры А, В, МН, КФ (как константа). Использование этих полей определяется кодом поля М:

— при АД формируются длинный 12-разрядный адрес перехода (поля КСТ/АПСТ, АПМЛ, УСЛ 1, УСЛ 0) и короткая 4-разрядная константа (поле КМЛ);

— при АК формируются короткий 8-разрядный адрес (поля АПМЛ, УСЛ 1, УСЛ 0) и длинная 8-разрядная константа (поля КСТ/АПСТ и КМЛ); при этом поля УСЛ1 и УСЛО участвуют

в формировании адреса не непосредственно своими кодами, а микрооперациями установки РАПП [1] и РАПП [0] в соответствующее состояние.

Контроль. Поля этой группы представляют собой контрольные разряды: КР1 — контрольный разряд для константы РМК [45—52], КР2 — контрольный разряд РАПП, КР3 — контрольный разряд РМК.

Резерв. Резервные разряды 59—61-й микрокоманды не используются.

Триггеры состояний (служебные триггеры). Поля микрокоманд УСТАНОВ, УСЛ1, УСЛО определяют в основном микрооперации по установке и анализу состояний триггеров, отражающих внутреннее состояние процессора. Триггеры регистра РБС, т. е. РБС [0], РБС [1] и т. д., используются для запоминания сигналов сбоя по адресации и защите памяти, а также как универсальные индикаторы для организации разветвлений в микропрограммах.

Триггер результата прямой функции ТРПФ устанавливается в состояние 1 при каждой микрооперации, выполняемой в АЛБ по прямой функции, если результирующий байт не равен нулю. Аналогичны функции и триггера результата косвенной функции ТРКФ.

Триггер знака (ТЗН) хранит информацию о состоянии старшего, т. е. нулевого, разряда байта результата выполнения микрооперации в АЛБ. Триггер четности байта (ТЧЕТ) хранит информацию о состоянии (значении) младшего разряда байта результата. Триггер переполнения (ТПЕР) хранит информацию о наличии или отсутствии переполнения восьмиразрядной сетки АЛБ.

Триггер неверных десятичных данных (ТНДД) хранит информацию о правильности задания десятичных данных; он устанавливается в состояние 1 при неверной десятичной цифре на любом входе АЛБ. Триггер вводо-выводных и внешних запросов (ТВВВ) устанавливается в состояние 1 при появлении запроса на внешние или вводо-выводные прерывания, а также на корректировку таймера. Триггер блокировки прерываний (ТБП) используется для блокировки прерываний по защите, адресации и машинному контролю во время выполнения микропрограммы гашения системы.

Если адрес команды находится в локальной (местной) памяти, то триггер адреса команды (ТАК) находится в состоянии 1; в состоянии 0 он находится тогда, когда этот адрес располагается в регистре МФЕ. Состояние 1 триггера центрального процессора (ТЦП) отражает работу каналов, а состояние 0 — работу процессора. Триггер признака блока защиты (ТБЗ) находится в состоянии 1, если в процессоре не установлен блок защиты. Состояние 1 триггера разрешения пакета (ТРП) указывает на то, что внешнее устройство запрашивает монопольный режим работы.

Отдельный триггер постоянной памяти (ТПП) представляет фактически схему занесения в РАПП [12]. Он устанавливается в состояние 1 при обращении к микрокоманде, адрес которой $A_{МК} \geq 4096$. Единичное состояние триггера тяжелого останова (ТТО) приводит к физическому останова машины за счет прекращения формирования синхроимпульсов ТИ, ХИ, СИ. Такое состояние этого триггера отражает режим тяжелого останова, который создается при возникновении сбоя в процессе реализации диагностических программ, включаемых в работу после выявления машинной ошибки (сбоя) любого характера.

Организация адреса микрокоманды. По способу организации адрес микрокоманды может быть нефиксированным и фиксированным. Нефиксированные адреса — это те, которые образуются с помощью информации, содержащейся в полях М, УСЛ1, УСЛ0, КСТ/АПСТ, АМПЛ выполняемых микрокоманд. Нефиксированный адрес может образовываться четырьмя способами, отвечающими кодам поля М. При этом код 10 может вызвать сброс регистра адреса постоянной памяти, что обеспечивает переход к микропрограмме выборки очередной команды. При функциональном переходе Ф с помощью УСЛ0 и содержимого четырех разрядов, пересылаемых из регистра РВ арифметическо-логического блока в РАПП [1—4], возможно осуществление разветвления по 32 различным направлениям.

В 13-разрядных регистрах РВС и РВМ запоминается адрес микрокоманды, перед которой произошли прерывание рабочей микропрограммы и переход на выполнение микропрограммы обслуживания селекторного или мультиплексного канала. Так как микропрограмма обслуживания селекторного канала имеет более высокий приоритет, то она может прервать микропрограмму обслуживания мультиплексного канала. В этом случае в РВМ запоминается адрес микрокоманды рабочей микропрограммы, к которой необходимо перейти после выполнения микропрограммы обслуживания мультиплексного канала, а в РВС запоминается адрес микрокоманды микропрограммы обслуживания мультиплексного канала, к которой необходимо перейти после обслуживания селекторного канала.

Фиксированные адреса — это адреса, формирование которых производится без участия полей микрокоманд. Эти адреса заносятся в РАПП по синхроимпульсу ХИ2 и дают начало микропрограммам гашения системы обработки прерываний по контролю машины, обслуживанию селекторного канала, обслуживанию мультиплексного канала, обработки сбоев по защите и адресации.

7.3. СХЕМНОЕ УПРАВЛЕНИЕ

Управление считается схемным тогда, когда формирование управляющих сигналов осуществляется специальными схемами, узлами центрального ядра устройства управления. Основные управляющие сигналы формируются, как правило, схемами блока

центрального управления, производящего расшифровку кодов операций. Остальные блоки могут формировать управляющие сигналы для «внутреннего» потребления, а также для обеспечения взаимодействия с другими блоками, особенно при совмещении циклов, как это осуществляется в ЕС-1050.

Структура центрального ядра «схемного» устройства управления приведена на рис. 7.11; она дана применительно к случаю совмещения трех циклов, когда производится так называемая трехуровневая обработка инструкций процессора. Уровню К отвечает блок выборки команд (БВК); уровню Ч отвечают блок центрального управления (БЦУ), блок сумматора адреса (БСА) и блок выборки данных (БВД); уровню Т отвечает блок адреса результата (БАР). Кроме того, к центральному ядру устройства управления относится блок местной памяти (БМП), содержащий группу быстродействующих регистров. Основные связи блоков и их состав рассматриваются ниже с позиций обеспечения всех необходимых действий при реализации инструкций процессора.

Блок выборки команд. Выходные коды оперативной памяти машины ЕС-1050 имеют формат двойного слова, что превышает формат любой из возможных команд. Поэтому считается, что блок выборки команд обеспечивает выборку из ОП участков программ, а второй его главной функцией является выделение команд (инструкций) из участков программ, т. е. их распаковка. Структурная схема блока приведена на рис. 7.12, где выделены его основные узлы и их внутренние и внешние связи. К таким узлам относятся: счетчик тактов начального цикла (СЧНЦ), счетчик тактов уровня К (СЧТК), счетчик адреса программ, или счетчик адреса участков программ (СЧАП), регистр адреса программ (РАП), шифратор кода длины (ШКД), младшие разряды регистра слова состояния программы (РССП), регистры команд РК1 и РК2, счетчик команд (СЧК) с выходным регистром команд (РК), вспомогательный регистр команд (РКВ), дешифратор регистра команд (ДШРК) и регистр распакованной команды (РКР).

Узлы рассматриваемого блока по функциональным признакам составляют три группы, обеспечивающие формирование адресов участков программ, выбираемых из оперативной памяти, распаковку команд (инструкций) и управление всеми действиями на уровне К. Составной частью первой группы являются младшие разряды РССП [32—63], содержащие код длины команды — РССП [32—33], признак результата — РССП [34—35], маску программы — РССП [36—39] и адрес инструкции (адрес участка программы) — РССП [40—63]. Информация в РССП заносится

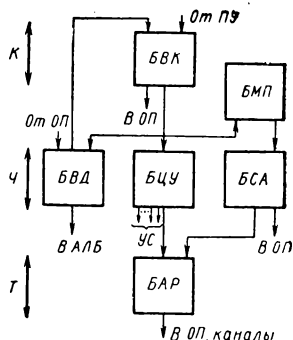


Рис. 7.11. Центральное ядро «схемного» устройства управления

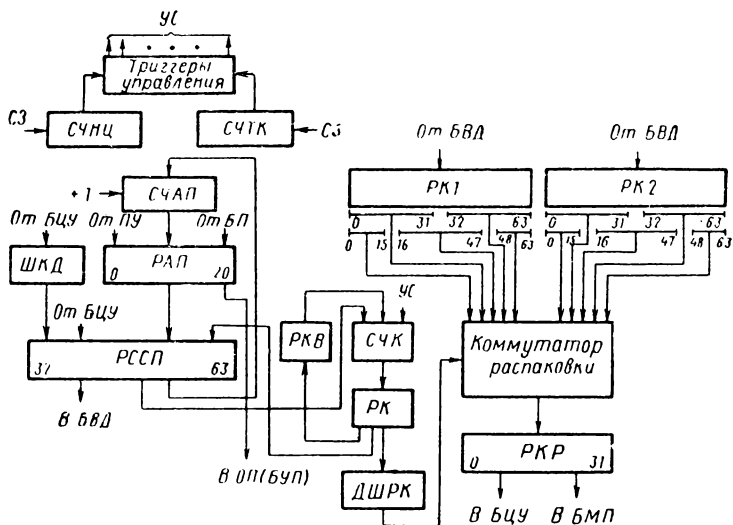


Рис. 7.12. Блок выборки команд

из шифратора кода длины, схем пульта управления, узлов блока центрального управления. В процессе формирования адресов участков программ главную роль играет кольцевая связь РССП → СЧАП → РАП → РССП.

СЧАП обеспечивает формирование адреса очередного участка путем прибавления единицы к содержимому РССП [40—63]. Вновь полученный адрес фиксируется в РАП, откуда выдается через блок управления памятью в ОП, а также в РССП [40—63], замещая прежний адрес участка программы. Адрес в РАП может вводиться из блока прерываний, а также от наборного регистра адреса пульта управления.

Управление работой узлов первой группы осуществляется посредством СЧНЦ, который по сигналам запуска (СЗ) формирует четыре тактовых импульса, приводящие в необходимое состояние триггеры управления. Триггеры управления обеспечивают непосредственное воздействие на группы вентилей РАП и РССП, а также управляемые цепи этих регистров и СЧАП при выполнении действий по преобразованиям и передачам информации.

Двойное слово, выбираемое из оперативной памяти по содержимому РАП, через блок выборки данных вводится в один из регистров РК1 или РК2, что определяется состояниями триггеров занятости этих регистров (если находящаяся в регистре информация не использована полностью, то соответствующий триггер занятости находится в состоянии 1). Каждый из регистров команд имеет пять групп выходных вентилей, обеспечивающих выделение двух полных слов [0—31] и [32—63], полуслов [0—15]

и [48—63], а также «срединного» слова [16—47]. Все это отвечает выделению команд различных форматов и необходимо при их распаковке.

Управление процессом распаковки команд производится с помощью СЧК, связанного с триггерами управления подобно счетчику начального цикла. На выходах триггеров формируются необходимые управляющие сигналы.

Передача содержимого РК1 и РК2 на РКР может производиться восемью способами, отвечающими всем возможным форматам команд и условиям их реализации. Передача осуществляется через коммутатор распаковки, управляемый сигналами с ДШРК, имеющего восемь выходов. Дешифрируемый трехразрядный код регистра команд формируется счетчиком команд. Формирование кода производится с учетом действий на предыдущем шаге распаковки. Базовой величиной для СЧК является содержимое РССП [60—62]. Если в РКР не содержится никакой код, то содержимое РССП [60—62] в счетчике команд не изменяется и непосредственно определяет способ передачи распаковываемой команды в РКР. Если в РКР находится команда формата RX, RS, SI или четырехбайтная часть команды формата SS, то на СЧК к содержимому РССП [60—62] прибавляются две единицы. Если же в РКР находится команда формата RR или двухбайтная часть команды формата SS, то на СЧК к содержимому РССП [60—62] прибавляется единица.

Указанный порядок трансформирования содержимого РССП [60—62] отвечает порядку расположения команд (инструкций) в восьмибайтных кодах, находящихся в регистрах РК1 и РК2. Поскольку распаковка команды формата SS производится в два приема, то в этом случае код из РК через вспомогательный регистр РКВ вновь подается на счетчик команд для организации кода, по которому из РК1 (РК2) в РКР передается вторая часть команды.

Размещение в РКР кода очередной команды, или инструкции процессора, свидетельствует об окончании ее обработки на уровне К. Поэтому при завершении действий по распаковке команды триггер готовности уровня К устанавливается в состояние 1. Если к этому моменту времени на уровне Ч завершена работа по реализации предыдущей команды, то начинается выдача кода операции в блок центрального управления, а адресов — в блок местной памяти.

Блок местной памяти. Местная память — это регистровое сверхбыстродействующее ОЗУ, предназначенное для хранения операндов и результатов выполнения некоторых операций, а также кодов базы и индекса, используемых при формировании исполнительных адресов. Как и любое запоминающее устройство, блок местной памяти включает в свой состав накопитель, регистры, дешифраторы, группы логических элементов и схему управления.

Накопитель выполнен в виде группы триггерных регистров, разделенной на две части. Первую часть составляют регистры для

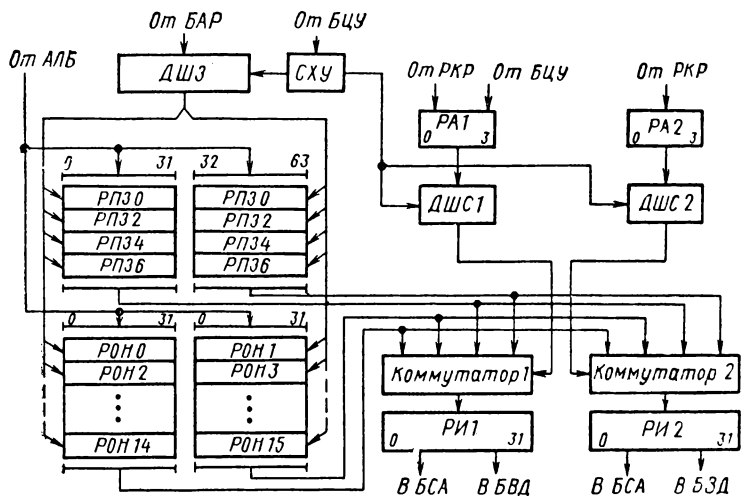


Рис. 7.13. Структура блока местной памяти

размещения чисел, представленных в форме с плавающей запятой; эти регистры являются 64-разрядными и обозначены на общей схеме блока местной памяти (рис. 7.13) как РПЗ 0, РПЗ 2, РПЗ 4, РПЗ 6. Вторую часть составляют регистры для размещения чисел, представленных в форме с фиксированной запятой. Таких регистров шестнадцать; они обычно называются регистрами общего назначения и являются 32-разрядными. На рис. 7.13 они обозначены как РОН 0—РОН 15.

Ввод информации в регистры производится через группы входных вентилях, автономные для каждого регистра. Записываемые коды подаются из АЛБ, как результаты выполнения операций или как информация, следующая транзитом из оперативной памяти. Управление записью осуществляется дешифратором записи (ДШЗ) на 4 входа и 16 выходов, схемой управления (СХУ), а также триггерами переключения управляющих сигналов, не показанными на схеме. Переключение управляющих сигналов необходимо вследствие того, что адреса РПЗ и восьми первых регистров общего назначения совпадают. Регистр адреса записи в состав рассматриваемого блока не входит; код адреса на ДШЗ подается с одного из регистров блока адреса результата.

Вывод информации из регистрового накопителя производится через группы выходных вентилях, автономные для каждого регистра. Распределение считываемых кодов по выходным информационным регистрам РИ1 и РИ2 осуществляется с помощью двух коммутаторов, управляемых дешифраторами считывания ДШС1 и ДШС2. Эти дешифраторы обеспечивают расшифровку кодов адресов операндов, базы или индекса, поступающих из регистра распакованной команды блока выборки команд или

из блока центрального управления в регистры адреса РА1 или РА2. Из выходных информационных регистров РИ1 и РИ2 коды выдаются в блок сумматора адреса и блок выборки данных.

Запись кодов в регистры РОН или РПЗ местной памяти осуществляется за три полутакта; считывание кодов из этих регистров в регистры РИ1 и РИ2 осуществляется также за три полутакта. На первом полутакте устанавливается значение адреса в РА1 или РА2, на втором — производится дешифрирование адреса с помощью ДШС1, ДШС2 или ДШЗ. На третьем полутакте при записи информация вводится в один из регистров накопителя, а при считывании код из накопителя вводится в РИ1 или РИ2.

Блок центрального управления. Формирование основных управляющих сигналов в рассматриваемом устройстве управления осуществляется за счет непосредственной расшифровки кода операции. Это производится в блоке центрального управления, который одновременно является основным управляющим органом по обработке инструкций (команд) на уровне Ч. Основные узлы блока и их связи показаны на рис. 7.14. Для формирования управляющих сигналов используются дешифратор кода операций (ДШКО), счетчик тактов уровня Ч (СЧЧ), дополнительный счетчик тактов М (СЧМ), коммутатор и триггеры управления. Входной и выходной регистры команд, т. е. РКМ и РКЧ, а также регистр адреса РА выполняют вспомогательные функции по временному хранению кодов, используемых в процессе реализации команд. Кроме этих узлов в состав блока центрального управления входят малоразрядные счетчики и регистры, используемые для управления выполнением команд групповой обработки информации и некоторых дополнительных операций.

СЧЧ построен как сдвигающий регистр и формирует шесть тактовых импульсов, используемых во всех блоках устройства управления, отнесенных к уровню Ч. Начало работы этого счетчика по сигналу запуска СЗ определяется моментом установки в состояние 1 триггера готовности уровня К при условии незаня-

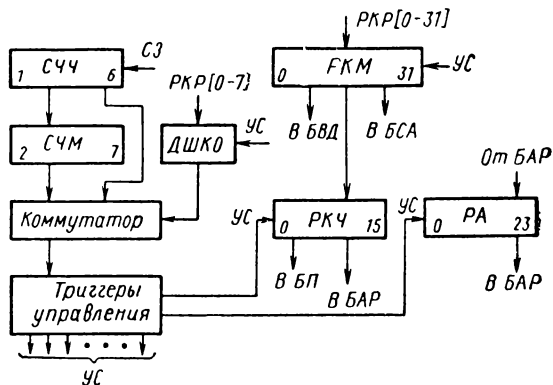


Рис. 7.14. Общая схема блока центрального управления

тости блоков уровня Ч. Сформированный первый тактовый импульс ТИ1Ч используется для приема на входной регистр блока кода команды из блока выборки команд, возбуждения элементов первой ступени дешифратора кода операции и выполнения ряда других действий. Дальнейшее формирование тактовых импульсов производится либо счетчиком СЧЧ, либо счетчиком СЧМ, причем СЧМ формирует серию ТИ2М—ТИ7М (шесть тактовых импульсов) в промежутке между ТИ1Ч и ТИ2Ч, когда работа счетчика СЧЧ приостанавливается.

Формирование тактовых импульсов, начиная со второго, определяется форматом команды, поступающей для обработки на уровень Ч. Если на уровне К была подготовлена команда формата RR, то продолжает работать счетчик СЧЧ, формирующий ТИ2Ч—ТИ6Ч, используемые для выборки операндов из местной регистровой памяти и передачи их в АЛБ для выполнения операции. Если же на уровне К была подготовлена команда другого формата, т. е. требующая формирования 24-разрядного исполнительного адреса, то после ТИ1Ч начинается работать счетчик СЧМ, формирующий ТИ2М—ТИ7М. Эти тактовые импульсы используются для выборки из местной памяти кодов базы и индекса, управления работой БСА в процессе формирования исполнительного адреса, а также формирования запроса для выборки из ОП операнда по этому адресу. После окончания работы СЧМ вновь начинает работать СЧЧ, формирующий сигналы, по которым завершается реализация команды на уровне Ч.

На первом такте работы уровня Ч, т. е. по тактовому импульсу ТИ1Ч, код команды принимается из РКР блока выборки команд в РКМ блока центрального управления. На третьем такте два старших байта команды передаются из РКМ в РКЧ, называемый выходным регистром команд блока центрального управления. Наличие двух регистров команд на уровне Ч позволяет начинать работу на уровнях Ч и Т одновременно, по одному и тому же сигналу синхронизации. Из регистра РКМ информация может выдаваться также в блок выборки данных и блок сумматора адреса. Из РКЧ информация передается, главным образом, в блок адреса результата.

Регистр РА служит для промежуточного хранения адреса первого операнда при выполнении операций по инструкциям форматов SI и SS. Никаких других функций этот регистр не выполняет.

Дешифратор кода операций получает информацию непосредственно из старших разрядов регистра распакованной команды блока выборки команд, т. е. содержимое РКР [0—7]. С каждым выходом дешифратора связан один или несколько триггеров управления, формирующих управляющие сигналы при расшифровке данного кода операции. Собственно дешифратор построен по двухступенчатой схеме с расшифровкой на первой ступени четверок разрядов кода операции.

Блок сумматора адреса.

Основной функцией блока сумматора адреса является формирование исполнительного адреса, по которому операнд для реализуемой на уровне Ч операции выбирается из оперативной памяти. В предельном случае при обработке инструкции формата RX исполнительный адрес образуется как сумма трех чисел: базы, или базового адреса, индекса и смещения. Базовый адрес и индекс представляют собой 24-разрядные двоичные числа, размещающиеся в РОН блока местной памяти. Смещение представляет собой 12-разрядное двоичное число, содержащееся в коде команды. Поэтому основными источниками информации для рассматриваемого блока являются РИ1 и РИ2 блока местной памяти и РКМ блока центрального управления.

Принципы формирования исполнительного адреса поясняются с помощью схемы, приведенной на рис. 7.15 и отражающей основной состав блока сумматора адреса. Основными узлами рассматриваемого блока являются: входной сумматор адреса (СМА1) с регистрами переноса (РП) и суммы (РС), основной сумматор адреса (СМА2) с выходным регистром исполнительного адреса (РИА), группы вентилях. Схема, управляющая работой сумматоров, а также некоторые вспомогательные узлы на рисунке не показаны.

Сумматоры, используемые в составе БСА, представляют собой комбинационные схемы суммирования с групповыми переносами. Так как на входы СМА1 подаются коды трех чисел, то он может обеспечить формирование на своих выходах только кода поразрядной суммы и кода переносов. Эти коды фиксируются соответственно в регистрах РС и РП. Последующее суммирование содержимого РС и РП на основном сумматоре адреса СМА2 дает код исполнительного адреса операнда, фиксируемый в выходном регистре блока сумматора адреса РИА. Отметим, что блок сумматора адреса обеспечивает формирование адресов операндов и при реализации инструкций формата SS . Для этого он имеет дополнительные входы, позволяющие получать разность адресов и организовывать счет байт при взаимодействии со схемами блока центрального управления.

В процессе формирования исполнительного адреса осуществляется контроль правильности выполнения действий в сумматорах рассматриваемого блока. С этой целью в состав сумматоров включены специальные цепи образования значений контроль-

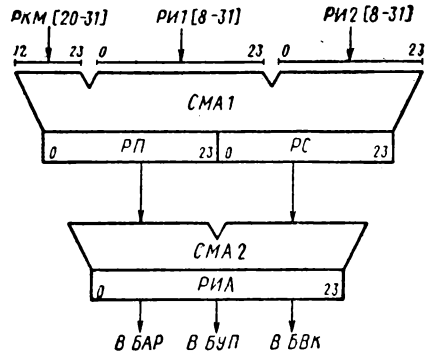


Рис. 7.15. Схема формирования исполнительного адреса

ных функций; при этом достигается глубина контроля, характеризующаяся контролем на уровне групповых переносов. Для каждого байта исполнительного адреса в процессе суммирования кодов обязательно формируется контрольный разряд, значение которого фиксируется отдельным триггером. После записи кода исполнительного адреса в РИА производится свертка его байт по модулю два для формирования значений контрольных разрядов. Таким образом, для исполнительного адреса имеются две группы контрольных разрядов: одна формируется по результатам действий в сумматорах, а вторая — непосредственно по коду исполнительного адреса. При совпадении всех пар контрольных разрядов по этим группам считается, что исполнительный адрес сформирован правильно.

Блок выборки данных. Центральное устройство управления получает информацию из оперативной памяти только через блок выборки данных; арифметическо-логические блоки процессора получают операнды из ОП также только через этот блок. Можно сказать, что блок выборки данных служит для выборки и хранения команд и операндов перед началом их обработки соответственно на уровне К устройства управления и в АЛБ. Рассматриваемый блок включает в свой состав три восьмибайтных регистра: РБЦ — буферный регистр центрального управления, РЧ1 — регистр первого числа, РЧ2 — регистр второго числа, а также однобайтный регистр ключей памяти (РКП), как это показано на рис. 7.16. Кроме того, в состав блока входят управляющие триггеры и группы логических элементов.

РБЦ принимает двойные слова из блока управления памятью, т. е. из соответствующих ячеек ОП; кроме того, в него возможен ввод информации с переключателей пульта управления, называемых наборным регистром данных. Двойные слова передаются из РБЦ в блок выборки команд (БВК), в РЧ1 или РЧ2 рассматриваемого блока; содержимое РБЦ [32—63] может передаваться также в регистр слова состояния программы блока прерываний.

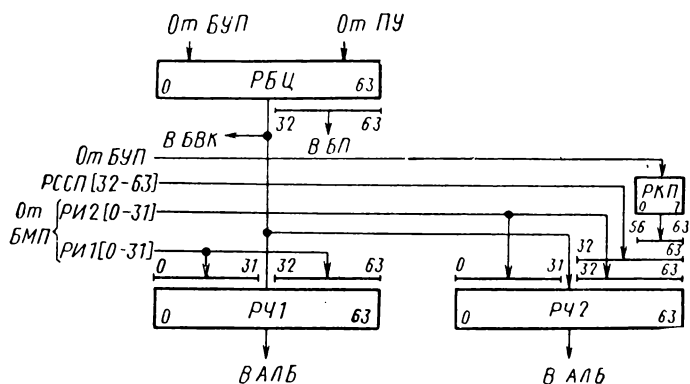


Рис. 7.16. Структура блока выборки данных

Отметим, что РБЦ, как и все остальные регистры процессора, имеет контрольные разряды — по одному на каждый байт. Содержимое контрольных разрядов дополняет количество единиц информационных байтов до нечетного числа и используется при контроле пересылок информации.

Регистры РЧ1 и РЧ2 построены по стандартным схемам, полностью совпадающим для всех разрядов. Прием двойных слов в эти регистры производится только из РБЦ. Пословно в РЧ1 информация вводится из выходного регистра РИ1 блока местной памяти. Аналогично из РИ2 блока местной памяти информация вводится в РЧ2. В младшие разряды РЧ2 может вводиться содержимое РССП [32—63], а также содержимое регистра ключей памяти. Использование РКП обусловлено необходимостью обеспечить выполнение специальных команд, задающих операции с ключами памяти.

Информация, хранящаяся в РЧ1 и РЧ2, представляет собой операнды, передаваемые в АЛБ. При этом выдача данных из РЧ1 производится, как правило, пословно, а из РЧ2 — по два байта, причем полуслова могут передаваться как прямо, т. е. без сдвига, так и со сдвигом на один или два байта. При выполнении инструкций формата SS оба операнда поступают в РЧ1 и РЧ2 через буферный регистр РБЦ из оперативной памяти, а при выполнении инструкций формата RR — из блока местной памяти через его выходные регистры РИ1 и РИ2. Если выполняется инструкция формата RX или RS, то первый операнд в РЧ1 вводится из РИ1 блока местной памяти, а второй операнд в РЧ2 — из оперативной памяти. При выполнении инструкций формата SI первый операнд в РЧ1 поступает из оперативной памяти, а второй операнд в РЧ2 — из РКМ блока центрального управления как часть кода реализуемой инструкции процессора.

Блок адреса результата. Обработка команд (инструкций процессора) на уровне Т обеспечивается только одним блоком ЦУУ — блоком адреса результата, который предназначен для управления записью результата выполнения операции, полученного из АЛБ, в местную память, формирования запроса в блок управления памятью, если результат должен записываться в ОП, а также для выдачи информации в каналы при выполнении операций ввода-вывода. Управление работой узлов рассматриваемого блока производится с помощью шестиразрядного счетчика тактов (СЧТ) уровня Т, формирующего по сигналам запуска СЗ шесть тактовых импульсов ТИ1Т—ТИ6Т. В формировании тактовых импульсов участвует коммутатор, как это показано на рис. 7.17. Кроме счетчика в состав блока входит регистр команд уровня Т (РКТ), дешифратор кода операций уровня Т (ДШК) с выходным регистром РДШК, регистр адреса результата (РАТ), дешифратор номера канала (ДШНК) и регистр номера канала (РНК).

В процессе реализации инструкции процессора на РКТ принимаются два старших байта кода команды из регистра команд

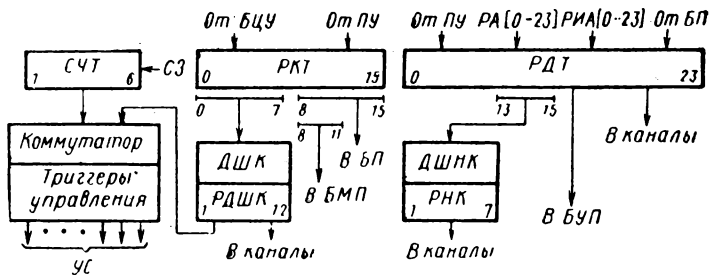


Рис. 7.17. Общая схема блока адреса результата

блока центрального управления. Возможен также ввод информации в РКТ из наборного регистра адреса пульта управления. Поскольку на уровне Т выполняются некоторые операции, то содержимое РКТ [0—7] подается на дешифратор кодов операций, имеющий 12 выходов. Сигналы, образующиеся на выходах ДШК, устанавливают в состояние 1 триггеры регистра РДШК. Четыре триггера этого регистра отвечают признакам четырех возможных операций ввода-вывода и выдают сигналы непосредственно в каналы. Остальные восемь триггеров отвечают операциям, которые выполняются в блоках устройства управления. Сюда относятся операции со словом состояния программы или с его частями, операции прямого управления и операция трансляции. Содержимое РКТ [8—15] может выдаваться в блок прерываний.

Регистр адреса результата уровня Т (РАТ) является своеобразным собирателем кодов адресов, направляемых в блок управления памятью для организации записи в ОП; при работе с каналами адрес записи из РАТ выдается на шины адреса устройств ввода-вывода. При выполнении инструкций форматов RX и RS адрес в РАТ вводится из регистра исполнительного адреса (РИА) блока сумматора адреса, а при выполнении инструкций формата SS — из регистра РА блока центрального управления. Если запись в оперативную память производится с пульта управления, то адрес в РАТ вводится из наборного регистра адреса ПУ. При прерываниях на РАТ [17—20] передается информация из блока прерываний; эта информация при сбросе в нулевое состояние РАТ [0—16] используется для формирования адреса записи старого ССП.

Содержимое РАТ [13—15] при работе с каналами передается на дешифратор номера канала (ДШНК), имеющий семь выходов. С каждым выходом дешифратора связан триггер регистра номера канала (РНК), отвечающий одному из каналов (мультиплексному или одному из шести селекторных). Возбуждение, т. е. установка в единичное состояние, триггера РНК обеспечивает формирование сигнала начала работы соответствующего канала; сигналы начала работы передаются в каналы по радиальным, т. е. автономным, линиям.

Обработка команд на уровне T производится как бы в два приема. Сначала по первым трем тактовым импульсам счетчика тактов (СЧТ) осуществляется прием информации в РКТ и РАТ, а также подготовка соответствующих схем для записи результата операции. Затем наступает пауза в работе узлов блока адреса результата, во время которой производятся действия в АЛБ. После паузы на четвертом, пятом и шестом тактах работы СЧТ производится запись результата по адресу, находящемуся в РАТ.

Блок адреса результата, как единственный блок центрального устройства управления, обеспечивающий обработку команд на уровне T, включает в свой состав схемы, формирующие запросы на прерывания по уровню T. Запросы на прерывания формируются при нарушении защиты памяти (совместно с БУП), при неправильной адресации, а также при поступлении соответствующих сигналов от арифметическо-логических блоков.

7.4. СИСТЕМА ПЕРЕРЫВАНИЙ И ПРИОРИТЕТОВ

Во всех машинах Единой системы процесс прерываний организуется как аппаратными, так и программными средствами. Центральные устройства управления моделей старших номеров включают в свой состав отдельные блоки прерываний со схемной реализацией процедуры опроса триггеров специального регистра, хранящего запросы на прерывания, и учета приоритетности прерываний. В моделях младших номеров, например в ЕС-1020, запросы на прерывания, а также коды масок отдельных прерываний размещаются в отдельных триггерах служебных регистров процессора (см. п. 7.2). Анализ содержимого триггеров производится по микрооперациям полей микрокоманд; при выявлении незамаскированного запроса на прерывание осуществляется передача управления первой команде программы, управляющей обработкой запроса на прерывание; состояние процессора на момент прерывания его работы запоминается, чтобы после завершения прерывающей программы можно было бы возобновить работу по прерванной программе.

Основная информация, отражающая состояние процессора и необходимая для возобновления выполнения прерванной программы, отражается в ССП — слове состояния программы, имеющем формат двойного слова. Структура ССП приведена на рис. 7.18. Восьмиразрядная маска системы включает коды следующих

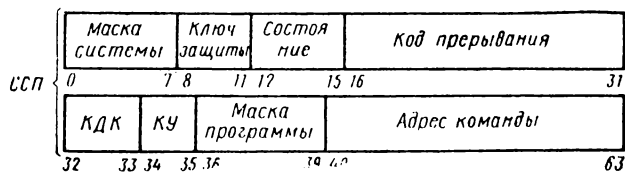


Рис. 7.18. Слово состояния программы

масок: нулевой разряд — маска мультиплексного канала, 1, 2, 3, 4, 5, 6-й разряды — маски селекторных каналов с такими же номерами (в ЕС-1030 не используются 5-й и 6-й разряды, а в ЕС-1020 — 3, 4, 5 и 6-й разряды), 7-й разряд — маска внешних прерываний. В разрядах «Состояние» отражается следующее: 12-й разряд — режим работы с КОИ-8, 13-й — маска прерываний от схем контроля машины, 14-й — состояние ОЖИДАНИЕ, 15-й — состояние ЗАДАЧА. Код длины команды КДК отражает количество полуслов для команды, к которой необходимо вернуться после обработки прерывания. КУ — код условия (признак результата для арифметической операции).

В четырехразрядной маске программы содержатся коды масок для четырех из пятнадцати возможных программных прерываний: 36-й разряд — маска переполнения с фиксированной запятой, 37-й — маска десятичного переполнения, 38-й — маска потери порядка, 39-й — маска потери значимости. Адрес команды в ССП позволяет возобновить выполнение прерванной программы в месте ее прерывания.

Состояния процессора. Процессоры машин Единой системы могут находиться в состояниях, определяемых четырьмя парами так называемых программных состояний, причем каждые две пары взаимно исключают друг друга. Три пары состояний отражаются в ССП, которое содержит всю информацию, необходимую для полного определения состояния процессора. Программные состояния составляют следующие пары:

- 1) ОСТАНОВ — РАБОТА;
- 2) ЗАДАЧА — СУПЕРВИЗОР;
- 3) ОЖИДАНИЕ — ВЫПОЛНЕНИЕ ПРОГРАММЫ;
- 4) МАСКА — ПРЕРЫВАНИЕ.

Первая пара состояний в ССП не отражается, так как нет никаких программных средств для перехода из состояния останова в рабочее состояние (состояние функционирования) и наоборот. Переходы обеспечиваются при нажатии соответствующих клавиш на пульте управления. Так, переход в состояние РАБОТА обеспечивается только при нажатии клавиши «ПУСК», а также при первоначальной загрузке программы. Состояние ОСТАНОВ характеризуется тем, что команды не выполняются, а запросы на прерывания игнорируются; к нему осуществляется переход при нажатии клавиш «ОСТАНОВ», «ПИТАНИЕ ВКЛ», «ГАШЕНИЕ» на пульте управления, а также при реализации с пульта останова по адресу или работы в режиме «Команда» (после завершения выполнения одной команды).

Остальные три пары состояний могут задаваться только программным путем за счет изменения значений соответствующих разрядов ССП. При состоянии ЗАДАЧА все команды, относящиеся к вводу-выводу, защите памяти и прямому управлению, а также команды ЗАГРУЗИТЬ ССП, УСТАНОВИТЬ МАСКУ СИСТЕМЫ, ДИАГНОСТИКА являются недопустимыми. Они называются

привилегированными и выполняются только при состоянии СУПЕРВИЗОР, когда допустимы все команды системы. Состоянию ЗАДАЧА отвечает код 1 в 15-м разряде ССП, а состоянию СУПЕРВИЗОР — код 0 в этом разряде. Переключение указанных состояний возможно при помощи команды ЗАГРУЗИТЬ ССП, а также при прерываниях и первоначальной (начальной) загрузке программы.

Если процессор находится в состоянии ОЖИДАНИЕ, то команды не выполняются, но запросы на внешние и вводи-выводные прерывания воспринимаются и обрабатываются. В состоянии ВЫПОЛНЕНИЕ ПРОГРАММЫ происходит нормальная выборка и выполнение команд. Состоянию ОЖИДАНИЕ отвечает код 1 в 14-м разряде ССП, состоянию ВЫПОЛНЕНИЕ ПРОГРАММЫ — код 0 в этом разряде. Переключение указанных состояний, т. е. изменение значения 14-го разряда ССП, производится только при занесении нового ССП в качестве текущего, т. е. реализуемого. В любом из этих состояний счет времени таймером не прекращается.

Четвертая пара состояний как бы отражает отношение процессора к прерываниям. Процессор реализует прерывание, т. е. находится в состоянии ПРЕРЫВАНИЕ, только тогда, когда это прерывание не маскируется. Запрос на прерывание игнорируется, т. е. процессор находится в состоянии МАСКА, если данное прерывание замаскировано. Прерывание считается замаскированным (запрещенным), если в соответствующем разряде ССП содержится код 0. При коде 1 прерывание разрешается. Переключение состояния процессора по отношению к прерываниям производится путем изменения значений разрядов масок ССП. Четырехразрядную маску программы можно изменить при помощи команды УСТАНОВИТЬ МАСКУ ПРОГРАММЫ, маску системы — при помощи команды УСТАНОВИТЬ МАСКУ СИСТЕМЫ. Маска контроля машины изменяется только при засылке нового ССП. Значения всех масок могут быть изменены при выполнении команды ЗАГРУЗИТЬ ССП, при прерывании по обращению к супервизору и при первоначальной загрузке программы. Смена масок не влияет на счет времени в таймере.

Прерывания. Все возможные прерывания (источники прерываний) в машинах Единой системы разделены на пять групп:

- прерывания по контролю машины, или прерывания от схем контроля машины;
- прерывания по ошибкам программы, или программные прерывания;
- прерывания при обращении к супервизору (вызов супервизора);
- прерывания по сигналам от внешних источников, или внешние прерывания;
- прерывания от устройств ввода-вывода, или вводи-выводные прерывания.

Все ситуации, вызывающие прерывание данной группы, анализируются (обрабатываются) отдельной программой, являющейся частью супервизора. Таким образом пяти группам прерываний отвечают пять супервизорных программ.

Управление выполнением любой программы производится в соответствии с информацией, содержащейся в ССП. Слово состояния программы, которой управляет выполняемой в данный момент времени программой, называется текущим. При прерывании это ССП должно запоминаться в оперативной памяти, а на его место должно вводиться ССП, отвечающее прерывающей программе. Поэтому каждой группе прерываний отвечают два ССП, называемые соответственно старым и новым и размещаемые в определенных (фиксированных) ячейках оперативной памяти.

В самом общем плане для всех пяти групп прерываний собственно прерывание заключается в запоминании текущего ССП в ячейке, предназначенной для старого ССП, и выборке из соответствующей ячейки нового ССП, становящегося текущим. Новое ССП, став текущим, управляет выполнением программы обработки запроса на прерывание. Конечной командой этой программы всегда бывает команда ЗАГРУЗИТЬ ССП, по которой текущее ССП заменяется на старое, т. е. система возвращается в состояние, предшествовавшее прерыванию, к выполнению прерванной программы. Общая процедура прерывания схематично отображена на рис. 7.19. Для этого рисунка: 1 — выполнение рабочей программы; 2 — первый этап прерывания с отсылкой текущего ССП в ячейку старого ССП; 3 — второй этап прерывания, связанный с началом обработки запроса на прерывание; 4 — обработка запроса; 5 — восстановление прерванного состояния.

Обычно прерывания производятся после завершения выполняемой процессором команды. В этом случае результаты заносятся в память, а код условия устанавливается в обычном порядке. В некоторых случаях выполнение команды может быть прекращено с занесением в память результата или его части; занесение

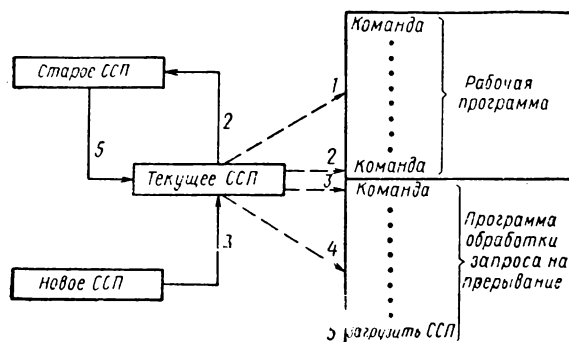


Рис. 7.19. Общая процедура прерывания

в память может и не производиться. Отдельным случаям прерывания отвечает подавление выполняемой команды, что равносильно ее пропуску.

Для обеспечения возможности обработки запроса на прерывание в старом ССП всегда фиксируется причина прерывания (код прерывания), а также информация, позволяющая установить расположение в памяти последней выполнявшейся до прерывания команды (адрес команды и код ее длины). Основная информация по прерываниям, в том числе все возможные источники прерываний по соответствующим группам отражены в табл. 7.1. Буквами «а» в этой таблице отмечены разряды адреса устройства ввода-вывода, «в» — разряды полей R1 и R2 команды ОБРАЩЕНИЕ К СУПЕРВИЗОРУ, «с» — другие условия прерывания.

Как уже указывалось, выполнение любой программы производится под управлением текущего ССП. Если в составе ЦУУ выделяется специальный регистр слова состояния программы (РССП), как это сделано, например, в машинах ЕС-1050 и ЕС-1030, то информация, содержащаяся в текущем ССП, хранится в этом регистре. При этом РССП обычно разделяется на две части, хранящие соответственно старшие и младшие разряды ССП. Части РССП функционально могут относиться к различным блокам ЦУУ: РССП [0—31], например, к блоку прерываний, а РССП [32—63] — к блоку выборки команд.

В ЕС-1020 в составе ЦУУ нет специального регистра для хранения слова состояния программы. Информация, содержащаяся в текущем ССП, хранится как в регистрах процессора, так и в ячейках локальной (местной) памяти. Места размещения отдельных частей ССП отвечают данным табл. 7.2, в которой номера ячеек ЛП представлены в шестнадцатиричной системе счисления.

Код прерывания в текущем ССП не имеет смысла, так как он формируется во время прерывания и запоминается в ячейках, отведенных для старого ССП. Код длины команды получается путем модификации двух первых разрядов кода операции выполняемой команды. Фиксация кода прерывания в слове состояния программы производится в соответствии с данными табл. 7.1 и только при его запоминании в ячейках ОП машины.

Приоритеты прерываний. Приоритетность прерываний распределяется по группам, причем программные прерывания и прерывания при обращении к супервизору имеют одинаковый приоритет, так как они взаимно исключают друг друга, т. е. запросы на программные прерывания и запрос на вызов супервизора не могут формироваться одновременно. Глубина прерываний отвечает четырем принятым приоритетам для групп источников прерываний. Высший приоритет имеет прерывание по контролю машины (приоритет 1). Далее в порядке приоритетности идут вводо-выводные, внешние и программные прерывания, включая прерывания при обращении к супервизору; поскольку при одновременном поступлении запросов на эти прерывания

Прерывания

Источник прерывания	Код прерывания	Раз- ряд маски	КДК	Прерываемая команда
1. Контроль машины				
Сбой в машине	00000000 00000000	13	—	Прекращается
2. Программные прерывания				
1) Некорректность кода операции	00000000 00000001	—	1, 2, 3	Подавляется
2) Привилегированная операция	00000000 00000010	—	1, 2	»
3) Некорректность команды ИСПОЛНИТЬ	00000000 00000011	—	2	»
4) Нарушение защиты памяти	00000000 00000100	—	0, 2, 3	Подавляется или прекращается
5) Неправильная адресация	00000000 00000101	—	0, 1, 2, 3	То же
6) Неправильная спецификация	00000000 00000110	—	1, 2, 3	Подавляется
7) Неправильные десятичные данные	00000000 00000111	—	2, 3	Прекращается
8) Переполнение с фиксированной запятой	00000000 00001000	36	1, 2	Завершается
9) Некорректность деления с фиксированной запятой	00000000 00001001	—	1, 2	Подавляется или завершается
10) Десятичное переполнение	00000000 00001010	37	3	Завершается
11) Некорректность десятичного деления	00000000 00001011	—	3	Подавляется
12) Переполнение порядка	00000000 00001100	—	1, 2	Прекращается
13) Потеря порядка	00000000 00001101	38	1, 2	Завершается
14) Потеря значимости	00000000 00001110	39	1, 2	»
15) Некорректность деления с плавающей запятой	00000000 00001111	—	1, 2	Подавляется
3. Вызов супервизора				
Код операции в команде	00000000 вввввввв	—	1	Завершается
4. Внешние прерывания				
1) Программа ВРЕМЯ, или таймер	00000000 1ccccccc	7	—	Завершается
2) Клавиша ПЕРЕРЫВАНИЕ на ПУ	00000000 c1cccccc	7	—	То же
3) Внешний сигнал 2	00000000 cc1cccc	7	—	»
4) Внешний сигнал 3	00000000 ccc1cccc	7	—	»
5) Внешний сигнал 4	00000000 cccc1ccc	7	—	»

Источник прерывания	Код прерывания	Разряд маски	КДК	Прерываемая команда
6) Внешний сигнал 5	00000000 ccccc1cc	7	—	Завершается
7) Внешний сигнал 6	00000000 ccccc1c	7	—	То же
8) Внешний сигнал 7	00000000 ccccc1	7	—	»
5. Вводо-выводные прерывания				
1) Канал 0 (МК)	00000000 aaaaaaaa	0	—	Завершается
2) Канал 1 (СК-1)	00000001 aaaaaaaa	1	—	То же
3) Канал 2 (СК-2)	00000010 aaaaaaaa	2	—	»
4) Канал 3 (СК-3)	00000011 aaaaaaaa	3	—	»
5) Канал 4 (СК-4)	00000100 aaaaaaaa	4	—	»
6) Канал 5 (СК-5)	00000101 aaaaaaaa	5	—	»
7) Канал 6 (СК-6)	00000110 aaaaaaaa	6	—	»

обработка ССП производится в порядке, обратном степени важности прерываний, то их приоритеты нумеруются, начиная с четвертого, т. е. 4, 3, 2. Если выполняется некоторое прерывание, то процессор маскируется для дальнейших прерываний той же группы.

При выполнении прерывания по контролю машины действия по текущей команде прекращаются. Программное прерывание или прерывание для вызова супервизора, которое могло произойти в результате выполнения текущей команды, игнорируется. В то же время сброс машины обычно не влияют на вводо-выводные, внеш-

Т а б л и ц а 7.2

Размещение частей ССП

Части текущего ССП	Место расположения	
	Локальная память	Регистры процессора
Маска системы	Ячейка 88	РБР [0—2] и РБР [7]
Ключ защиты	Ячейка 89	РБЗ [4—7]
Режим работы с КОИ-8	Ячейка 89	—
Маска прерываний от схем контроля машины	Ячейка 89	РБР [5]
Состояние ОЖИДАНИЕ	Ячейка 89	РБД [2]
Состояние ЗАДАЧА	Ячейка 89	—
Код прерывания	—	—
Код длины команды	—	—
Код условия	—	РБС [6—7]
Маска программы	Ячейка 8С	—
Адрес команды	Ячейки 8D, 8E, 8F	Регистр МФЕ

ние прерывания, работу таймера и передачу данных при вводе-выводе информации по селекторному каналу.

Если одновременно имеют место запросы на программные (вызов супервизора), внешние и вводи-выводные прерывания, то в машине производится следующее. Текущее ССП, т. е. ССП для прерываемой рабочей программы запоминается в ячейке 40 (для ЕС-1020 и ЕС-1030) основной памяти, а новое ССП, загружается на место текущего из ячейки 104. Но это ССП не реализуется, так как есть запросы на внешние и вводи-выводные прерывания. Текущее ССП, ставшее новым ССП программных прерываний, запоминается в ячейке 24, а вместо него загружается новое ССП из ячейки 88—ССП внешних прерываний. Затем это слово состояния программы запоминается в ячейке 56, а в качестве текущего становится выбираемое из ячейки 120 новое ССП—ССП прерываний по вводу-выводу. Таким образом, загрузка ССП производится в порядке номеров приоритетов этих прерываний, обратном их степени важности. Отметим, что при реализации прерывания по обращению к супервизору вместо программного прерывания используются не ячейки 40 и 104, а 32 и 96 соответственно.

Реализация прерываний производится в порядке их истинной приоритетности или важности. Так как текущим стало ССП вводи-выводных прерываний, то в рассматриваемом случае оно реализуется первым. В конце программы обработки прерывания от ввода-вывода выполняется загрузка старого ССП для этого прерывания, т. е. ССП из ячейки 56. Теперь текущим становится ССП внешних прерываний, и реализуется программа обработки запроса на такой вид прерываний. В конце этой программы выполняется загрузка ССП из ячейки старого ССП для внешних прерываний, т. е. из ячейки 24. Текущим становится ССП программных прерываний, которые и обрабатываются с помощью соответствующей программы. В конце программы обработки загружается ССП из ячейки 40, т. е. текущим становится ССП, которое обеспечивало управление выполнением рабочей программы до ее прерывания; процессор продолжает реализацию прерванной рабочей программы.

Если одновременно поступают запросы на вводи-выводные прерывания от нескольких источников, то удовлетворение этих запросов производится по отдельной приоритетной шкале. Высшим приоритетом обладают запросы мультиплексного канала; для селекторных каналов устанавливаются приоритеты в соответствии с их порядковыми номерами.

Блок прерываний. При схемной реализации большинства действий по осуществлению прерываний блок прерываний имеет в своем составе узлы, обеспечивающие хранение и формирование ССП, определение адресов ССП, прием и хранение запросов на прерывания, маскирование прерываний и учет их приоритетности. Структура такого блока прерываний приведена на рис. 7.20,

где приняты следующие обозначения: РЗП — регистр запросов на прерывания, СЗЗ — схема записи запросов, СП — сбой процессора, ЗПП — запросы на программные прерывания, ЗОС — запрос на обращение к супервизору, ЗВП — запросы на внешние прерывания, ЗВВП — запросы на вводо-выводные прерывания, СПП — схема приоритетов прерываний, РПП — регистр приоритетов прерываний, РАС — регистр адреса ССП, ШПП — шифратор программных прерываний, СПВВП — схема приоритетов вводо-выводных прерываний, РУК — регистр управления каналами, ШНК — шифратор номера канала, РССП — регистр слова состояния программы.

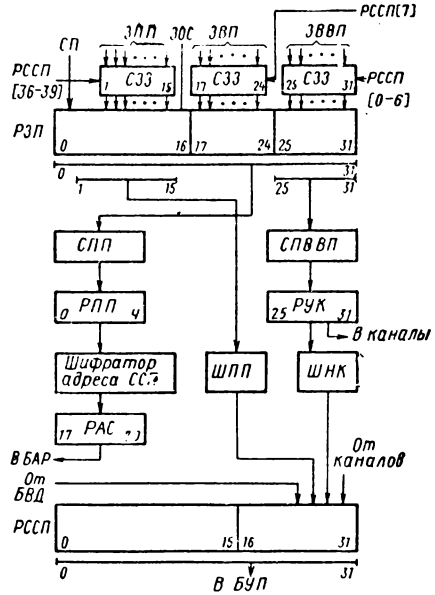


Рис. 7.20. Структура блока прерываний

Все запросы на прерывания фиксируются в 32-разрядном регистре РЗП. Входные для этого регистра схемы записи запросов обеспечивают учет масок прерываний в соответствии с данными табл. 7.1 Практически СЗЗ представляют собой набор вентилях, которые управляются сигналами с выходов соответствующих триггеров регистра слова состояния программы. Если, например, РССП [7], содержащий код маски внешних прерываний, находится в состоянии 1, то любой запрос на внешние прерывания фиксируется в РЗП; в противном случае запросы не пропускаются на этот регистр, что отвечает случаю запрета внешних прерываний.

Схема приоритета прерываний совместно с РПП обеспечивает учет приоритетов по всем группам источников прерываний. В первую очередь выявляется наличие запроса на прерывание по контролю машины. Далее просмотр содержимого РЗП производится в порядке условных номеров приоритетов, начиная с программных прерываний. Это производится для того, чтобы при наличии нескольких запросов на различные прерывания в конечном итоге первой производилась обработка наиболее важных запросов, начиная с вводо-выводных прерываний, как это уже отмечалось выше. В регистре приоритетов прерываний имеется 5 разрядов — по одному на каждую группу запросов; это сделано вследствие того, что при обращении к супервизору для нового и старого ССП отводятся не те ячейки, что в случае программных прерываний.

Собственно схема приоритета прерываний представляет собой группу логических элементов, реализующих следующие зависимости:

$$\begin{aligned} \text{РПП [0]} &: = 1 \text{ при } \text{РЗП [0]} = 1, \\ \text{РПП [1]} &: = 1 \text{ при } \overline{\text{РЗП [0]}} \wedge \text{РЗП [1—15]} = 1, \\ \text{РПП [2]} &: = 1 \text{ при } \overline{\text{РЗП [0—15]}} \wedge \text{РЗП [16]} = 1, \\ \text{РПП [3]} &: = 1 \text{ при } \overline{\text{РЗП [0—16]}} \wedge \text{РЗП [17—24]} = 1, \\ \text{РПП [4]} &: = 1 \text{ при } \overline{\text{РЗП [0—24]}} \wedge \text{РЗП [25—31]} = 1; \end{aligned}$$

обозначение вида РЗП [1—15] указывает, что хотя бы в одном из разрядов от 1-го до 15-го регистра запросов на прерывания содержится код 1; обозначение вида $\overline{\text{РЗП [0—16]}}$ эквивалентно единице только тогда, когда во всех разрядах РЗП с нулевого по 16-й содержатся нули; учитывается, что после учета запросов соответствующие триггеры РЗП устанавливаются в состояние 0.

После выявления каждого запроса, т. е. после фиксации кода 1 в любом разряде РПП, шифратор адреса формирует и записывает в РАС адрес, по которому производится запись на место старого текущего ССП, находящегося в РССП. Адрес нового ССП формируется в РАС автоматически за счет установки в состояние 1 триггера его старшего разряда. Адреса ССП передаются в блок адреса результата (БАР); отсюда они направляются в блок управления памятью для организации обращения к ОП.

Схема приоритета вводу-выводных прерываний и регистр управления каналами работают аналогично паре СПП—РПП. Но начинают они работать только при обработке незамаскированного прерывания по вводу-выводу. Шифратор программных прерываний и шифратор номера канала обеспечивают формирование кода прерывания, записываемого в РССП при удовлетворении запроса на прерывание; при обработке запроса на прерывание по вводу-выводу в РССП [24—31] записывается номер устройства ввода-вывода, поступающий из соответствующего канала. Запись информации в РССП [16—31] производится в соответствии с данными табл. 7.1.

Работа схем блока прерываний по обработке одного запроса на прерывание происходит в два этапа. На первом этапе производятся все действия, связанные с записью текущего ССП на место старого, а на втором — действия, связанные с вводом нового ССП в регистр слова состояния программы. Все действия производятся по тактовым импульсам, связанным с общими для всего процессора синхронизирующими сигналами. Сначала осуществляется последовательный просмотр разрядов РЗП посредством схемы приоритетов прерываний. При выявлении незамаскированного прерывания возбуждается соответствующий разряд РПП, а в РАС устанавливается значение адреса старого ССП.

До передачи содержимого РССП в оперативную память в этот регистр должен быть записан код прерывания, включая и адрес устройства ввода-вывода для случая ввода-выводного прерывания. Далее адрес ССП и содержимое РССП передаются в БУП (через другие блоки ЦУУ), организующий запись старого ССП в ОП. После записи старого ССП содержимому РАС [17] присваивается значение единицы; тем самым в РАС устанавливается адрес нового ССП, передаваемый в БУП. Новое ССП, выбранное из оперативной памяти, вводится в РССП.

Ввод нового ССП в регистр слова состояния программы еще не означает передачу управления блоку центрального управления УУ. Такая передача осуществляется только после обработки всех запросов на прерывания.

7.5. УПРАВЛЕНИЕ ОПЕРАТИВНОЙ ПАМЯТЬЮ

Выделение в составе машин Единой системы каналов, обслуживающих внешние запоминающие устройства и устройства ввода-вывода, привело к тому, что непосредственный доступ к оперативной памяти стали иметь многие устройства и блоки. Оперативная память стала своеобразной системой массового обслуживания; при этом запросы на обслуживание, приходящие в любые моменты времени, не должны теряться, а удовлетворение запросов должно производиться в соответствии с принятыми приоритетами. Для реализации значительно увеличившихся и усложнившихся функций по управлению оперативной памятью используются специальные блоки управления памятью, сложность которых возрастает с увеличением количества каналов, количества блоков процессоров, имеющих доступ к ОП, и количества блоков ОП, включаемых в состав машины, или данной модели ЕС ЭВМ.

На рис. 7.21 показана структура внешних связей блока управления памятью (БУП) машины ЕС-1050 при ее максимальной комплектации, т. е. при использовании мультиплексного канала (МК), шести селекторных каналов (СК1—СК6), четырех блоков (устройств) основной оперативной памяти (ООП) и четырех блоков большей оперативной памяти (БОП). Процессор связан с рассматриваемым блоком посредством линий передачи управляющих сигналов (УС), шин передачи информации в виде двойных слов и линий передачи 24-разрядного адреса. Каналы имеют аналогичные связи с блоком управления памятью, но, если УС между каждым каналом и БУП передаются по отдельным, так называемым радиальным линиям, то адреса и информационные слова передаются по адресным и информационным шинам (АШ и ИШ соответственно), общим для всех каналов.

Связи между БУП и блоками ООП, а также БОП, построены как по радиальному, так и по шинному принципу. Управляющие сигналы и информация от блоков памяти в БУП передаются по

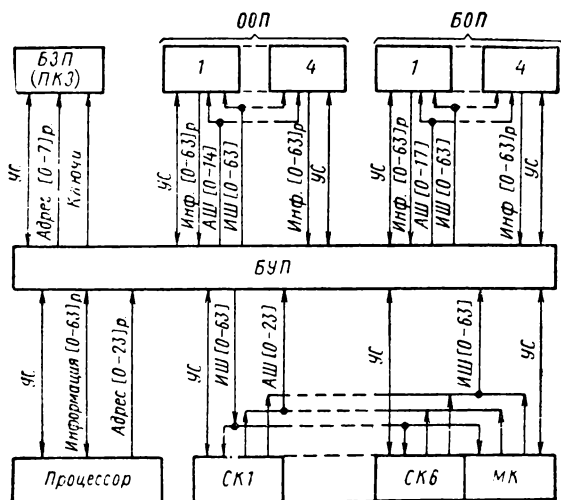


Рис. 7.21. Основные внешние связи блока управления памятью

радиальным линиям, а адреса и информация из БУП в блоки памяти — по адресным и информационным шинам. Такое построение связей определяется принятыми способами подключения блоков памяти.

Блок управления памятью в процессе работы взаимодействует с блоком защиты памяти (БЗП), одной из основных частей которого является память ключей защиты (ПКЗ). Обращение к ПКЗ производится по восьми старшим разрядам адреса, поступающего в блок управления от абонентов, т. е. процессора и каналов.

Общая структура блока управления памятью приведена на рис. 7.22. Она отражает только обобщенный состав этого блока и его основные внутренние связи. Для рассматриваемой схемы приняты следующие обозначения: РОЗ — регистр обобщенных запросов, РНА — регистр номера абонента, УФУС — узел формирования управляющих сигналов, РЗС1 — входной регистр запросного слова, РЗС2 — выходной регистр запросного слова, УУ ООП — узел управления обращением к основной оперативной памяти, УУ БОП — узел управления обращением к большой оперативной памяти, ПРИ — приемный регистр информации, УС — управляющие сигналы. Для блока защиты памяти (БЗП) на схеме показаны только отдельные связи, так как работа этого блока описывается ниже в гл. 9.

Основная часть узлов блока управления памятью работает как при обращении к СОП, так и при обращении к БОП. Автономные для этих видов оперативной памяти узлы управления

начинают работать только после расшифровки старших разрядов адреса, дающих вид памяти и номер блока, к которому производится обращение.

Обращение к оперативной памяти со стороны процессора может производиться четырьмя блоками, причем независимо друг от друга; такими блоками являются: блок контроля и диагностики (БКД), блок адреса результата (БАР), блок центрального управления (БЦУ) и блок выборки команд (БВК). В указанном порядке этим блокам присвоены приоритеты; блок контроля и диагностики имеет высший приоритет, а блок выборки команд, представляющий в ЦУУ первый уровень обработки кодов инструкций процессора, — низший приоритет. Запросы на обслуживание оперативной памятью этих блоков обрабатываются в схеме анализа запросов процессора, включающей в свой состав регистры, группы логических элементов и узел приоритетов запросов. Схема анализа всегда выделяет старший по приоритету запрос процессора.

Запросы на обслуживание оперативной памятью каналов обрабатываются в схеме анализа запросов каналов, построенной аналогично схеме анализа запросов процессора. Приоритеты для каналов, начиная с высшего, установлены в следующем порядке: СК1, СК2, МК, СК3, СК4, СК5, СК6; 1-й и 2-й селекторные каналы имеют более высокий приоритет по отношению к мультиплексному каналу, что объясняется местом размещения программ операционной системы.

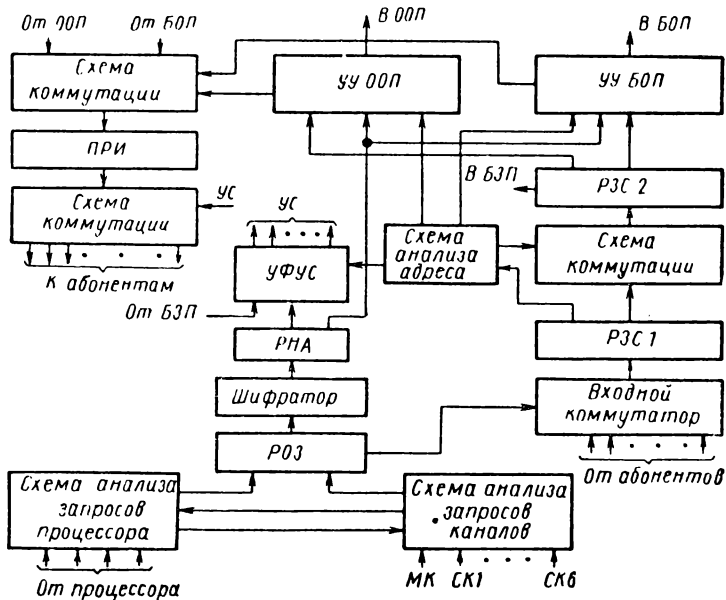


Рис. 7.22. Общая структура блока управления памятью

Обе схемы анализа запросов тесно взаимодействуют в процессе работы, так как любой запрос от каналов всегда выше по приоритету любого запроса процессора. При наличии хотя бы одного запроса от канала удовлетворение запросов от процессора временно блокируется (время, необходимое на обработку одного запроса, составляет 320 нс). Выявленный обоими схемами старший по приоритету на данный момент времени запрос фиксируется в виде кода 1 в соответствующем разряде регистра обобщенных запросов; фиксация запроса в РОЗ дает начало действиям по его удовлетворению, а схемы анализа начинают очередной цикл выявления запроса, подлежащего удовлетворению; циклы повторяются с периодом 320 нс, что отвечает принципу синхронной связи БУП с процессором и каналами.

Информация, зафиксированная в регистре обобщенных запросов, используется для формирования с помощью шифратора кода номера абонента, обслуживание запроса которого началось схемами блока управления. Абонентам присвоены следующие номера: СК1 — 1000, СК2 — 1001, МК — 1010, СК3 — 1011, СК4 — 1100, СК5 — 1101, СК6 — 1110, БКД — 0001, БАР — 0010, БЦУ — 0011, БВК — 0100. На завершающем этапе работы блока управления по удовлетворению запросов эти номера используются для организации выдачи информации из ООП или БОП.

Выявленный и зафиксированный в РОЗ, как старший по приоритету, запрос используется для организации приема от соответствующего абонента запросного слова, включающего в свой состав адрес ячейки оперативной памяти, признак записи или считывания, ключ защиты, информацию в виде двойного слова (в режиме записи) и несколько разрядов служебной информации. По старшим разрядам адреса, введенного в соответствующую часть РЗС1, схема анализа адреса устанавливает вид запрашиваемой памяти и номер блока внутри вида, т. е. номер блока ООП или номер блока БОП. После этого схема анализа адреса передает управление УУ ООП или УУ БОП. В случае незанятости запрашиваемого блока запросное слово передается с РЗС1 на РЗС2 через схему коммутации, обеспечивающую формирование кода адреса, непосредственно используемого в выбранном блоке оперативной памяти.

Блоки основной оперативной памяти всегда нумеруются кодами, записываемыми в 4-м и 5-м разрядах адреса; для этих разрядов код 00 отвечает ООП-1, 01 отвечает ООП-2, 10 отвечает ООП-3, 11 отвечает ООП-4.

Определение номера адресуемого блока большой оперативной памяти производится по значениям шести старших разрядов адреса с учетом комплектации ООП и БОП, т. е. с учетом количества блоков, реально имеющих в составе ООП и БОП.

Приемный регистр информации со входными и выходными схемами коммутации используется только при обращении к оперативной памяти по считыванию. Он обеспечивает временное хранение двойных слов, считываемых из ООП или БОП и их выдачу абонентам, обслуживаемым в установленном порядке. Прием и выдача информации всегда производится параллельными кодами.

Обслуживание блока защиты памяти в первую очередь заключается в передаче ему старших восьми разрядов адреса практически при каждом обращении к ОП (считается, что емкость памяти ключей защиты равна 256 словам). Кроме того блок управления памятью обеспечивает запись ключей в ПКЗ и их выборку из ПКЗ для отсылки в регистр слова состояния программы.

Общая последовательность действий узлов блока управления памятью по обработке запросов на обслуживание, поступающих от процессора и каналов, может быть разделена на три основных этапа, отвечающих обработке сигналов запросов, обработке запросных слов и работе с блоками оперативной памяти по записи или считыванию информации.

На первом этапе схемы анализа запросов в соответствии с принятыми приоритетами производят обзор содержимого своих входных регистров, в которых фиксируются сигналы запросов. Результатом этих действий является выделение старшего по приоритету запроса от каналов и старшего по приоритету запроса от процессора. Из этих двух запросов, если они выделяются на одном и том же цикле работы схем анализа, главным всегда признается запрос от каналов; этот запрос и принимается к обслуживанию. Если запросов от каналов нет, то к обслуживанию принимается старший по приоритету запрос от процессора. Анализ содержимого входных регистров запросов производится непрерывно по циклам, длительность которых равна 320 нс.

Завершающими для первого этапа являются действия по фиксации выделенного для обслуживания запроса в регистре обобщенных запросов, формированию кода номера абонента с фиксацией его в РНА и образованию сигнала, по которому осуществляется прием запросного слова от соответствующего абонента в РЗС1. После фиксации выделенного запроса в регистре обобщенных запросов начинается следующий цикл действий первого этапа.

Действия второго этапа связаны с анализом и некоторым преобразованием принятого в РЗС1 запросного слова. Анализ запросного слова состоит в расшифровке адреса и формировании управляющего слова. Собственно расшифровка адреса включает процедуры по контролю адреса на четность, определению вида памяти и номера блока ООП или БОП, а также по выявлению возможности обращения к адресуемому блоку (определению занятости его другими действиями). При расшифровке адреса

всегда определяется его действительность, т. е. принадлежность множеству принятых для оперативной памяти адресов. Основные действия по расшифровке адреса производит схема анализа адреса совместно с узлами управления обращением к ООП и БОП.

Управляющее слово, формируемое на рассматриваемом этапе, представляет собой 7-разрядный двоичный код, в котором старшие четыре разряда отведены под номер обслуживаемого по данному запросу абонента, следующие два — для размещения номера блока ООП или БОП; в младшем разряде располагается признак записи-считывания. Управляющие слова формируются отдельно для ОПП и отдельно для БОП с размещением их в соответствующих узлах управления; реализация управляющих слов обеспечивается на заключительном этапе действий по обслуживанию выделенного запроса.

В зависимости от готовности адресуемого блока памяти, вида запроса и вида используемой памяти возможны следующие направления действий на втором этапе обработки запроса:

1) запросное слово принадлежит каналу и обращение производится к занятому блоку ООП; в этом случае запросное слово сохраняется в РЗС1 до освобождения блока;

2) если запрос канала предназначен для занятого блока БОП, то РЗС1 освобождается для очередного запросного слова, а неудовлетворенный запрос канала размещается в специальном регистре, что обеспечивает его удовлетворение в первую очередь по освобождении адресуемого блока БОП;

3) если запрос процессора предназначен для занятого блока ООП или БОП, то РЗС1 освобождается для очередного запросного слова, а обработка запросов процессора приостанавливается до освобождения данного блока памяти;

4) если запросное слово адресует незанятый блок в памяти, то оно передается в РЗС2; на РЗС1 принимается новое запросное слово.

Передачей запросного слова в РЗС2 и формированием сигнала запуска ООП или БОП заканчивается второй этап действий блока управления памятью по обработке запроса абонента.

На третьем этапе производится запись информации, содержащейся в запросном слове, в адресуемый блок памяти или считывание двойного слова из блока ООП или БОП. При считывании двойное слово сначала записывается в приемном регистре информации (ПРИ), а затем в соответствии с содержимым управляющего слова выдается через схему коммутации соответствующему абоненту.

Для предотвращения наложения информации при считывании из ООП и БОП предусмотрено формирование специального сигнала в УУ БОП, по которому приостанавливается прием запросов от каналов и процессора до приема на ПРИ информации, считываемой из большой оперативной памяти.

7.6. СИСТЕМНЫЕ СРЕДСТВА УПРАВЛЕНИЯ

Под системными средствами управления, включаемыми в состав ЦУУ, понимаются блоки, предназначенные для организации вычислительного процесса в вычислительных системах, построенных на базе однотипных моделей, например, моделей ЕС-1050. Системные средства обеспечивают обмен управляющими сигналами и байтом управляющей информации между процессорами вычислительной системы через средства прямого управления, а также дают возможность программным средствам путем отсчета времени организовать правильное временное взаимодействие моделей, объединенных в вычислительную систему. К системным средствам управления относятся блок внешних связей и блок таймеров.

Блок внешних связей. Основным назначением блока внешних связей является обеспечение обмена сигналами прямого управления с другим процессором вычислительной системы при выполнении инструкций ПРЯМАЯ ЗАПИСЬ и ПРЯМОЕ ЧТЕНИЕ, а также формирование и обмен мультисистемными сигналами при работе процессоров вычислительной системы с общим полем оперативной памяти. Обмен сигналами прямого управления и мультисистемными сигналами осуществляется через отдельный интерфейс прямого управления.

При выполнении процессором инструкции ПРЯМАЯ ЗАПИСЬ блок внешних связей (ВС) осуществляет прием поля I_2 инструкции как содержимого РКТ [8—15] блока адреса результата и передачу разрядов этого поля во внешний процессор в качестве сигналов внешних прерываний; одновременно байт информации, выбранный из оперативной памяти по адресу операнда выполняемой инструкции, принимается в блок ВС из АЛБ и затем передается во внешний процессор. Сигналы прерываний от внешнего процессора, принимаемые блоком ВС, передаются в блок прерываний ЦУУ.

При выполнении инструкции ПРЯМОЕ ЧТЕНИЕ разряды поля инструкции аналогичным образом передаются во внешний процессор, а байт информации, принимаемый от внешнего процессора, передается в АЛБ и в конечном итоге записывается в оперативную память по адресу операнда выполняемой инструкции. В режиме обмена мультисистемными сигналами блок внешних связей осуществляет формирование и передачу во внешний процессор соответствующих сигналов в зависимости от значения разрядов [0—3] поля I_2 инструкции ПРЯМАЯ ЗАПИСЬ; формирование сигналов производится также при поступлении из блока контроля и диагностики сигналов машинной ошибки, а из блока прерываний — сигнала начальной загрузки программы. Мультисистемные сигналы, принятые от внешнего процессора в блок ВС, обрабатываются в зависимости от своего назначения либо как сигналы внешних прерываний (в блоке прерываний), либо как

сигналы, непосредственно воздействующие на работу блока центрального управления, блока таймеров и блока прерываний.

Блок внешних связей в соответствии с выполняемыми функциями разделяется на две основные части, одна из которых обеспечивает передачу информации и управляющих сигналов во внешний процессор, а вторая — прием информации и управляющих сигналов от внешнего процессора. Такая макроструктура блока отражена его схемой, приведенной на рис. 7.23. Для этого рисунка приняты следующие обозначения: ППЧ — признак инструкции ПРЯМОЕ ЧТЕНИЕ, ППЗ — признак инструкции ПРЯМАЯ ЗАПИСЬ, УФМС — узел формирования мультисистемных сигналов, РМС — регистр мультисистемных сигналов, РСС — регистр сигналов синхронизации (внешних сигналов), РВВ — выходной регистр информационного байта, УПД — усилители—передатчики, УПМ — усилители—приемники, УПМС — узел приема мультисистемных сигналов, РБП — приемный регистр информационного байта, РВС — приемный регистр внешних сигналов. Кроме указанных узлов в состав блока ВС входят управляющие триггеры и логические схемы, не показанные на рис. 7.23.

Работа блока внешних связей определяется режимами, принятыми для машин при объединении их в систему. Режимы работы задаются определенными положениями указателей на пульте управления системой и пульте реконфигурации.

В режиме «Модель» блок ВС осуществляет обмен сигналами прямого управления, т. е. передачу и прием разрядов поля I_2 инструкций ПРЯМАЯ ЗАПИСЬ и ПРЯМОЕ ЧТЕНИЕ в качестве сигналов внешних прерываний и обмен байтом информации. При блокировке прямого управления внешние сигналы не выдаются и не принимаются.

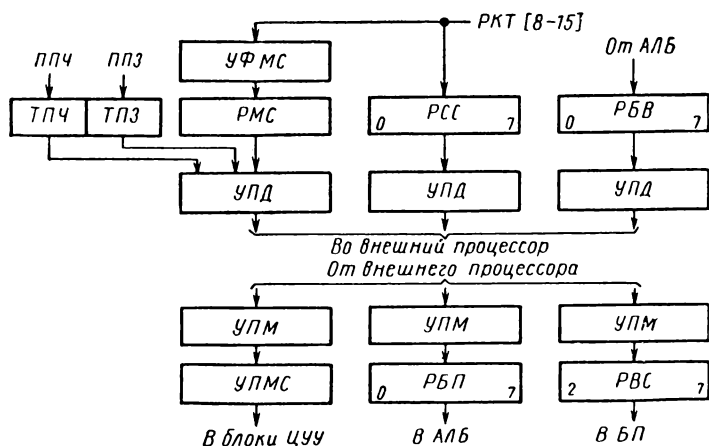


Рис. 7.23. Упрощенная схема блока внешних связей

В режиме «Мультисистема» четыре старших разряда поля I_2 инструкции ПРЯМАЯ ЗАПИСЬ расшифровываются с помощью УФМС как мультисистемные сигналы. Формирование мультисистемных сигналов производится также при наличии признаков машинной ошибки и начальной загрузки программы.

В режиме «Расчлененная система» блок внешних связей работает так же, как и в режиме «Мультисистема», но мультисистемные сигналы не выдаются и не принимаются.

Начало выполнения инструкций ПРЯМАЯ ЗАПИСЬ и ПРЯМОЕ ЧТЕНИЕ определяется моментом поступления соответствующего признака и переброса триггера прямой записи (ТПЗ), или триггера прямого чтения (ТПЧ) в состояние 1. Время выполнения операции ПРЯМАЯ ЗАПИСЬ составляет 960 нс, а время выполнения операции ПРЯМОЕ ЧТЕНИЕ — 1200 нс.

Блок таймеров. Основными узлами блока таймеров являются счетчики и регистры, составляющие основу интервального таймера и мультисистемного таймера; кроме этих таймеров в состав блока входят схемы контроля и диагностики.

Интервальный таймер предназначен для отсчета времени. Вместе с соответствующей программой он может служить как для получения отметок истинного времени, так и для измерения временных интервалов. Информация, отвечающая учитываемому времени, хранится в ячейке 80 (ячейке времени) оперативной памяти и представляет собой 32-разрядное слово; знак располагается в нулевом разряде. Ячейка времени доступна аппаратным средствам (интервальному таймеру) и программным (программам супервизора), которые организуют службу времени.

Интервальный таймер производит изменение содержимого ячейки времени либо с частотой сети 50 Гц при работе таймера с низкой разрешающей способностью (режим ТМН), либо с частотой 300 Гц при работе таймера с высокой разрешающей способностью (режим ТМВ). Выбор режима работы определяется положением переключателя на пульте управления. Начальное значение времени в ячейку 80 задается программными средствами службы времени.

Служба времени вычислительной системы, выполняемой на базе модели ЕС-1050, при использовании интервального таймера и совокупности программных средств, обеспечивает реализацию следующих функций:

- выдачу технического времени (часы, минуты) и даты,
- выдачу запроса на установление связи супервизора системы с программой пользователя после заданного периода времени,
- определение временного интервала после заданной отметки времени.

Все это позволяет эффективно использовать систему при ее работе в режиме разделения времени, при телекоммуникационном методе доступа, а также при отладке программ. При работе тай-

мера в режиме ТМН его разрешающая способность составляет 20 мс, а при работе в режиме ТМВ — 13 мкс; содержимое ячейки времени изменяется не чаще чем через 3,3 мс.

Мультисистемный таймер служит для выявления так называемых подвесок процессора, т. е. случаев, когда задерживается выполнение инструкций и время их выполнения превышает заданный временной интервал, а также для обеспечения небольших задержек для некоторых внешних по отношению к процессору сигналов.

Общая структура блока таймеров отражена на рис. 7.24, для которого приняты следующие обозначения: ФИ — формирователь импульсов, ГИ — генератор импульсов, СЧВ — счетчик интервального таймера с высокой разрешающей способностью, РСЧВ — регистр счетчика, ТП — триггер переноса (переполнения), ССС — схема сборки сигналов, СЧН — счетчик интервального таймера с низкой разрешающей способностью, ФК — формирователь кода, РТМ — выходной регистр результата; СЗ — сигнал запроса, СО — сигнал опроса, СУ — сигналы управления, СЧМТМ — счетчик мультисистемного таймера, РСЧ — регистр счетчика мультисистемного таймера, ЛС — логическая схема.

Мультисистемный таймер начинает работу по сигналам, не отмеченным на рис. 7.24. Содержимое его счетчика все время повторяется в регистре, триггеры которого управляют работой логической схемы. При поступлении на логическую схему сигналов опроса формируются сигналы управления, обеспечивающие реализацию общих функций мультисистемного таймера.

Интервальный таймер блока ВС в процессе своей работы непрерывно взаимодействует с блоками процессора, используя свои внешние связи, показанные на рис. 7.25. Дополнительные обозначения для этого рисунка: КВ — код вычитания, сформи-

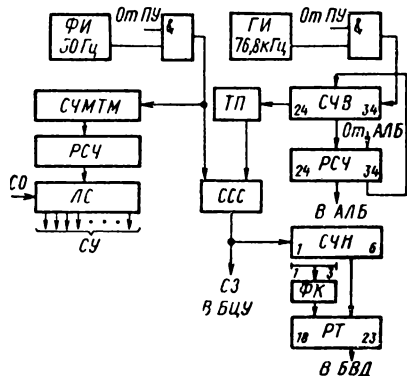


Рис. 7.24. Структура блока таймеров
300

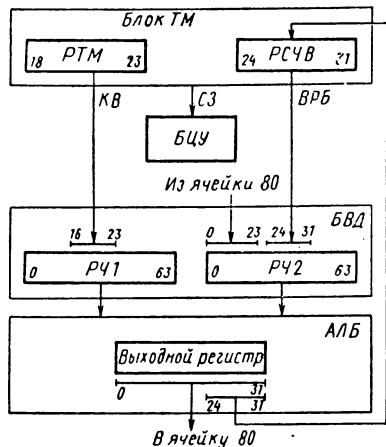


Рис. 7.25. Внешние связи блока таймеров

рованный в РТМ по содержимому СЧН; ВРБ — байт счета с высокой разрешающей способностью.

Коррекция времени, или продвижение таймера, производится блоками центрального управления (БЦУ) и АЛБ процессора. Для этого в РЧ2 [0—23] блока выборки данных (БВД) вводится содержимое ячейки времени, т. е. ячейки 80 оперативной памяти. При работе таймера в режиме ТМН в РЧ1 [16—23] вводится содержимое РТМ, отвечающее 20 мс, если удовлетворяется каждый запрос блока таймеров. Код вычитания в этом случае представляет собой байт, содержащий единицы в 21-м и 22-м разрядах. Коды из РЧ1 и РЧ2 передаются в АЛБ, где происходит изменение кода времени, содержащегося в ячейке 80. Новый код времени опять записывается в ячейку 80.

При выполнении указанных действий возможно неудовлетворение запросов интервального таймера блоками процессора. В случае задержки удовлетворения запроса, превышающей 160 мс, происходит потеря времени, учитываемого службой времени системы. Это считается равносильным появлению ошибки в работе системы.

Для получения более точных отсчетов времени используется режим ТМВ. В этом случае значение 31-го разряда слова времени изменяется с частотой 76,8 кГц, т. е. примерно через каждые 13 мкс. Изменение значения 23-го разряда происходит в этом режиме с частотой 300 Гц, т. е. в 6 раз больше, чем в режиме ТМН.

С целью исключения излишне частых обращений к оперативной памяти при работе интервального таймера в режиме ТМВ младший байт слова времени все время находится в РСЧВ блока ТМ. Изменение содержимого ячейки 80 происходит при каждом переполнении РСЧВ, т. е. через каждый 3,3 мс, чему отвечает единица в 23-м разряде.

Начальная установка времени всегда производится программно. При выполнении этой процедуры в режиме ТМВ байты 80, 81, 82 записываются в оперативную память, а байт 83 записывается в регистр РСЧВ блока ТМ. При обращениях к ячейке 80 по считыванию байты 80, 81, 82 принимаются из оперативной памяти, а байт 83 — из РСЧВ блока ТМ.

7.7. ПУЛЬТ УПРАВЛЕНИЯ

Пульт управления (ПУ) любой машины Единой системы предназначен для индикации состояния, контроля и управления режимами работы центрального процессора, а также ОП и каналов. Принципы построения и работы ПУ рассмотрим на примере пульта управления машины ЕС-1020.

В состав пульта управления входят отдельные регистры и наборы переключателей информации, рассматриваемые в процессе работы ПУ как регистры; кроме того, пульт содержит ряд

кнопок и лампочек индикации, располагаемых в панелях ПУ. Информация с пульта управления может передаваться на вход В арифметическо-логического блока процессора (АЛБ), обозначаемого на пульте ЕС-1020 как БА; прием информации на триггерные регистры ПУ может производиться с выхода С блока БА. Управление передачами информации, как и всей работой пульта управления, обеспечивается двумя микропрограммами: обслуживания пультовых операций (ОБСП) и первоначальной загрузки программы (ПЗП).

Для индикации информации, не отражаемой постоянно на панели индикации ПУ, используется отдельный регистр. Этот регистр информации представляет собой 18-разрядный триггерный регистр, в который может приниматься информация из оперативной памяти, из регистров процессора и каналов, не имеющих постоянной индикации.

Второй триггерный регистр пульта управления служит для фиксации признаков операций, выполняемых с ПУ, а также для счета временных интервалов. Этот регистр имеет шифр РР9 и является восьмиразрядным. Разряды [0—3] служат для запоминания признака типа операции, выполняемой с ПУ. Разряды [4—7] представляют собой двоичный счетчик, на вход которого с периодом $T = 20$ мс (при частоте $f = 50$ Гц для сети переменного напряжения) поступают импульсы стандартной формы. Максимальное содержимое счетчика отвечает временному интервалу, равному 300 мс. В процессе работы машины содержимое РР9 [4—7] используется для изменения значения таймера — числа, размещенного в стандартных ячейках (50—53) ОП и фиксирующего временные затраты на выполнение различных работ.

Конструктивно пульт управления состоит из двух панелей: панели управления и панели индикации.

Панель управления. На панели управления (рис. 7.26) расположены:

— переключатели набора информации АДРЕС ПАМЯТИ (А—Д) и АДРЕС КОМАНДЫ (Е—К);

— переключатели управления РЕЖИМ РАБОТЫ, КОНТРОЛЬ, СРАВНЕНИЕ АДРЕСОВ, ТИП ПАМЯТИ;

кнопки управления.

Набор переключателей АДРЕС ПАМЯТИ состоит из четырех переключателей (Б, В, Г, Д) на 16 положений каждый и одного переключателя (А) на 4 положения и предназначен для задания:

а) адреса оперативной памяти при считывании или записи информации (на переключателях А—Д);

б) адреса для сравнения с адресом оперативной памяти (на переключателях А—Д) или постоянной памяти (на переключателях Б—Д);

в) номера регистра канала или процессора при передаче их содержимого в регистр индикации пульта управления (на переключателе Д).

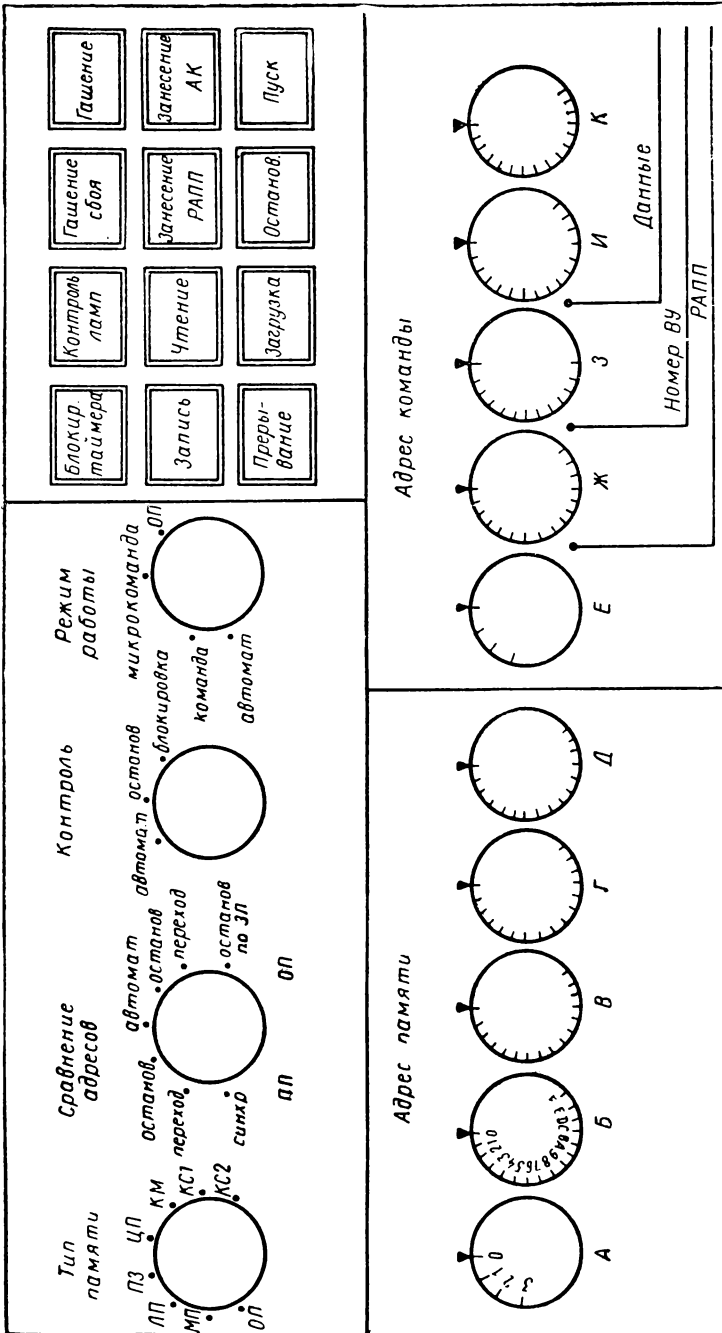


Рис. 7.26. Панель управления

Набор переключателей АДРЕС КОМАНДЫ состоит из одного переключателя на 4 положения (Е) и четырех переключателей (Ж, З, И, К) на 16 положений каждый; он предназначен для задания:

- а) адреса текущей команды (на переключателях Е—К);
- б) адреса постоянной памяти (на переключателях Ж—К);
- в) адреса внешнего устройства при вводе начальной программы (на переключателях И—К).

Переключатель ТИП ПАМЯТИ в зависимости от своего положения указывает, к какой памяти производится обращение с пульта управления:

- ОП — к основной памяти;
- МП — к мультиплексной памяти;
- ЛП — к локальной памяти;
- ПЗ — к памяти ключей защиты;
- ЦП — к регистрам центрального процессора;
- КМ — к регистрам мультиплексного канала;
- КС1 — к регистрам первого селекторного канала;
- КС2 — к регистрам второго селекторного канала.

Переключатель СРАВНЕНИЕ АДРЕСОВ указывает, с адресом какой памяти (постоянной или оперативной) должно производиться сравнение адреса, установленного на наборе переключателей АДРЕС ПАМЯТИ, и определяет реакцию процессора при сравнении этих адресов; эта реакция определяется положением переключателя и выражается как:

- останов по команде или микрокоманде;
- переход по адресу, заданному на переключателях АДРЕС КОМАНДЫ;
- останов при записи в оперативную память по адресу, установленному на переключателях АДРЕС КОМАНДЫ;
- выдача синхроимпульса при считывании указанной микрокоманды; в положении АВТОМАТ сравнение адресов не производится.

Переключатель РЕЖИМ РАБОТЫ позволяет задавать различные режимы работы процессора:

- АВТОМАТ (автоматический режим)—основной режим работы центрального процессора при автоматической реализации программ;
- в режиме «Команда» при каждом нажатии кнопки ПУСК выполняется одна команда;
- в режиме «Микрокоманда» при каждом нажатии кнопки ПУСК выполняется одна микрокоманда;
- режим ОП используется при обращении к ряду последовательно расположенных ячеек памяти.

Переключатель КОНТРОЛЬ определяет своим положением реакцию процессора при обнаружении сбоя:

- ОСТАНОВ — при возникновении сбоя осуществляются останов центрального процессора и индикация причины сбоя;
- БЛОКИРОВКА — осуществляется блокировка всех сигналов сбоя;

— АВТОМАТ — производится обработка данных, связанных с сигналом сбоя, по специальной программе.

На панелях пульта управления расположено 14 кнопок управления; двенадцать из них позволяют задавать различные режимы работы и вызывать определенные действия в основных устройствах машины, а две служат для включения и выключения цепей электропитания.

Кнопка «ГАШЕНИЕ» предназначена для перевода машины в исходное состояние независимо от того, в каком состоянии она до этого находилась (при включенном питании). В исходном состоянии подсвечиваются кнопка «ОСТАНОВ» и табло «РУЧНАЯ РАБОТА».

Кнопка «ЗАГРУЗКА» предназначена для запуска микропрограммы первоначальной загрузки программы (ПЗП). Во время выполнения первоначальной загрузки программы подсвечивается табло «ЗАГРУЗКА».

При нажатии кнопки «ОСТАНОВ» процессор переводится в состояние останова работы после выполнения текущей команды и обслуживания всех прерываний. В состоянии останова можно выполнять операции чтения, записи, занесения адреса команды и микрокоманды. При пребывании процессора в состоянии останова кнопка «ОСТАНОВ» подсвечивается.

Кнопка «ПУСК» предназначена для инициирования выполнения операции в режиме, заданном переключателем РЕЖИМ РАБОТЫ.

Кнопка «ЗАНЕСЕНИЕ АК» предназначена для передачи адреса текущей команды, набранного на соответствующих переключателях, в счетчик адресов команд (регистр РМФЕ процессора).

Кнопка «ЗАНЕСЕНИЕ РАПП» предназначена для передачи адреса микрокоманды, набранного на соответствующих переключателях, в регистр адреса постоянной памяти.

Кнопка «ЗАПИСЬ» предназначена для записи в ОП или регистры процессора информации, установленной на переключателях И, К, по адресу, указанному на наборе АДРЕС ПАМЯТИ. Запись информации выполняется по соответствующим микрокомандам. Во время выполнения операции кнопка «ЗАПИСЬ» подсвечивается.

Кнопка «ЧТЕНИЕ» служит для вывода на индикацию содержимого ячеек памяти и регистров процессора или каналов. Если нужно прочитать содержимое ячеек памяти, то при помощи переключателя ТИП ПАМЯТИ указывается одна из областей оперативной памяти (основная, мультиплексная, локальная) или память ключей защиты, а на наборе АДРЕС ПАМЯТИ указывается адрес требуемой ячейки. Считанная из ячейки информация индицируется в регистре информации. Если же необходимо прочитать содержимое регистра, то посредством переключателя ТИП ПАМЯТИ задается один из каналов или процессор, а на переключателе Д набора АДРЕС ПАМЯТИ устанавливается номер регистра. Считанная из выбранного регистра информация прини-

мается в регистр индикации. Во время выполнения рассматриваемой операции кнопка «ЧТЕНИЕ» подсвечивается.

Кнопка «ПЕРЕРЫВАНИЕ» служит для формирования запроса на внешнее прерывание, если оно не маскировано. При этом 25-й разряд текущего слова состояния программы (ССП) устанавливается в единичное состояние; это указывает, что источником внешнего прерывания является кнопка «ПЕРЕРЫВАНИЕ».

Кнопка «КОНТРОЛЬ ЛАМП» служит для проверки элементов и ламп системы индикации.

Кнопка «ГАШЕНИЕ СБОЯ» нажимается для сброса в исходное состояние всех индикаторов сбоев процессора.

Кнопка «БЛОКИРОВКА ТАЙМЕРА» управляет работой счетчика временных интервалов. Если эта кнопка подсвечивается, счет интервалов времени не производится.

Все указанные выше кнопки расположены непосредственно на панели управления. Описываемые же ниже — расположены на панели индикации.

Кнопка «ПИТАНИЕ ВКЛ.» нажимается для инициирования начала последовательности действий по включению питания. При включенном питании эта кнопка подсвечивается.

Кнопка «ПИТАНИЕ ОТКЛ.» нажимается для инициирования начала последовательности действий по отключению питания. Кнопка подсвечивается, если питание отключено.

Панель индикации служит для размещения элементов, обеспечивающих индицирование содержимого некоторых регистров и состояния отдельных управляющих триггеров мультиплексного канала, селективных каналов и процессора. В левой части панели расположены пять табло, характеризующие различные состояния процессора:

— ЗАГРУЗКА — выполняется программа первоначальной загрузки;

— СЖИДАНИЕ — процессор находится в состоянии ожидания работ;

— КОНТРОЛЬ — один из переключателей СРАВНЕНИЕ АДРЕСОВ, КОНТРОЛЬ или РЕЖИМ РАБОТЫ не находится в положении АВТОМАТ;

— РУЧНАЯ РАБОТА — процессор находится в состоянии останова или выполняется операция с пульта управления;

— СИСТЕМА — процессор выполняет одну из возможных микропрограмм.

Индикация регистров постоянной памяти. На панели индикации всегда показывается содержимое регистра микрокоманды и содержимое регистра адреса постоянной памяти (РАПП) (рис. 7.27). При работе в режиме МИКРОКОМАНДА или при останове по адресу постоянной памяти индицируются содержимое и адрес той микрокоманды, которая должна быть выполнена при следующем нажатии кнопки «ПУСК». При других режимах работы указанные регистры находятся в состоянии дина-

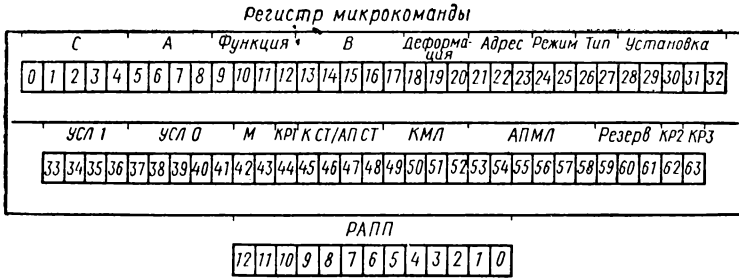


Рис. 7.27. Индикация регистров постоянной памяти

мического изменения находящейся в них информации, и считывание данных о их содержимом по элементам индикации невозможно.

Индикация мультиплексного канала. На рассматриваемой панели пульта управления индицируется состояние признаков интерфейса и контроля, а также содержимое регистра выходной информации мультиплексного канала (ШИН-К). Через информационные шины канала ШИН-К внешним устройствам передаются адреса устройств, коды команд и байты данных. Находящаяся на ШИН-К информация всегда идентифицируется признаками интерфейса, поэтому ее индицирование сопровождается индицированием соответствующих признаков. Схема индикации каналов приведена на рис. 7.28.

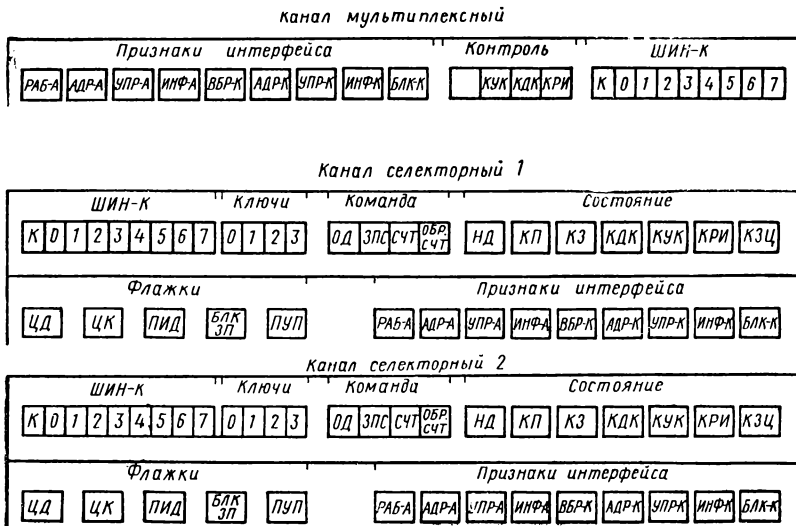


Рис. 7.28. Индикация каналов

На панели индикации подсвечиваются следующие признаки интерфейса:

— РАБ-А (работа внешнего устройства — абонента) — внешнее устройство выбрано и подключено к каналу;

— АДР-А (адрес от устройства) — внешнее устройство установило на выходные информационные шины (ШИН-А) свой адрес;

— УПР-А (управление от устройства) — внешнее устройство установило на ШИН-А байт состояния;

— ИНФ-А (информация от устройства) — выбранное внешнее устройство готово к передаче или приему байта информации;

— ВБР-К (выборка от канала) — канал опрашивает внешние устройства для определения ВУ, которое опознало свой адрес в начальной выборке или требует обслуживания (по ТРБ-А);

— АДР-К (адрес от канала) — канал установил на ШИН-К адрес внешнего устройства (если ВБР-К=1) или указывает ВУ: «отсоединиться от интерфейса» (если ВБР-К=0);

— УПР-К (управление от канала) — канал установил на ШИН-К байт кода команды обращения к внешнему устройству или указывает ВУ «запомнить байт состояния», если УПР-А = 1, или «остановиться», если ИНФ-А = 1;

— ИНФ-К (информация от канала) — канал указывает УВУ, что сигнал ИНФ-А или УПР-А воспринят;

— БЛК-К (блокировка от канала) — совместно с другими сигналами указывает о блокировке данных, блокировке состояния, селективном сбросе, цепочке команд канала.

Отражаемые на панели индикации признаки контроля мультиплексного канала означают:

— КУК (контроль управления канала) — на линиях интерфейса имеют место несовместимые признаки идентификации сигналов канала (УПР-К и ИНФ-К, АДР-К и ИНФ-К и т. д.);

— КДК (контроль данных канала) — принятый через интерфейс байт данных содержит неверный контрольный разряд;

— КРИ (контроль работы интерфейса): а) на линиях интерфейса одновременно имеют место несовместимые сигналы идентификации абонента: ИНФ-А и УПР-А, УПР-А и АДР-А и т. д.; б) байт состояния или байт адреса ВУ, принятый через интерфейс каналом, содержит неверный контрольный разряд.

Индикация селекторных каналов КС1 и КС2. На панели пульта управления размещаются элементы индикации выходных информационных шин селекторного канала ШИН-К, регистра ключей защиты, регистра команд, регистра состояния, регистра флажков и признаков интерфейса (см. рис. 7.28).

С помощью элементов индикации для ШИН-К индицируются как передаваемые каналом внешним устройствам адреса устройств, коды команд, байты данных, так и принимаемые от устройств байты состояния и байты данных. Находящаяся на ШИН-А информация идентифицируется посредством признаков интерфейса, отмеченных при рассмотрении индикации мультиплексного канала.

Регистр ключей защиты содержит ключ защиты, установленный в канале при выполнении инструкции НАЧАТЬ ВВОД—ВЫВОД.

Регистр команд содержит следующие признаки:

- ОД (операция передачи данных) — канал выполняет операцию передачи данных, т. е. ввод или вывод информации;
- ЗПС (запись) — канал передает информацию из ОП во ВУ;
- СЧТ (считывание) — канал принимает информацию из ВУ и записывает ее в ОП в порядке возрастания адресов;
- ОБР.СЧТ (обратное считывание) — канал принимает информацию из ВУ и записывает ее в ОП в порядке убывания адресов.

Для индикации состояния канала используются признаки:

- НД (неверная длина) — количество байтов данных, содержащихся в области памяти, используемой в данной команде, не равно количеству байтов, затребованных или передаваемых внешним устройством;
 - КП (контроль программы) — канал выявил программную ошибку;
 - КЗ (контроль защиты) — канал «пытался» записать (или произвести считывание) данные в защищенную область памяти; ключ защиты канала не равен ключу защиты области памяти, используемой в данной операции;
 - КДК (контроль данных канала) — принятый через интерфейс байт данных содержит неверный контрольный разряд;
 - КУК (контроль управления канала) — на линиях интерфейса одновременно имеют место несовместимые сигналы идентификации канала (например, УПР-К и ИНФ-К и т. д.); в оборудовании канала обнаружен сбой по четности в передаваемой информации;
 - КРИ (контроль работы интерфейса) — на линиях интерфейса одновременно имеют место несовместимые сигналы идентификации абонента; байт адреса или состояния, принятый каналом через интерфейс от ВУ, содержит неверный контрольный разряд;
 - КЗЦ (контроль зацепления) — содержимое счетчика байтов, заданное последним УСК по цепочке данных (в зацеплении по данным), меньше количества байтов, уже находящихся в буферном регистре.
- Индикация флажков отвечает признакам:
- ЦД (цепочка данных) — в УСК задана цепочка данных (зацепление по данным);
 - ЦК (цепочка команд) — в УСК задана цепочка команд;
 - ПИД (подавление индикации неверной длины) — указание о неверной длине при выполнении операции ввода-вывода блокируется, если нет ЦД;
 - БЛК ЗП (блокировка записи) — операция передачи данных будет выполняться без записи информации в ОП;

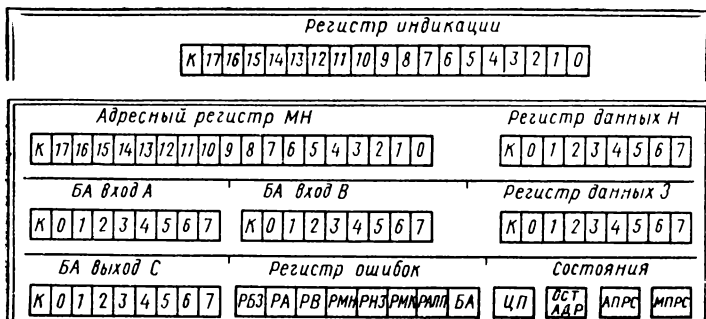


Рис. 7.29. Индикация процессора

— ПУП (программно-управляемое прерывание) — в УСК был задан признак ПУП, и канал будет формировать запрос на прерывание, когда это УСК будет введено в канал.

Регистр индикации обеспечивает индикацию содержимого регистров, не имеющих постоянной индикации, а также содержимого ячеек памяти. Если в регистр индикации (РИНД) вводится содержимое ячеек памяти или однобайтных регистров, то используются только [0—7] его разряды (см. рис. 7.29); старшие разряды РИНД при этом гасятся.

Индикация процессора. Элементы индикации состояний отдельных триггеров и регистров процессора показаны на рис. 7.29.

В регистре ошибок индицируются следующие признаки сбоев:

- РБЗ — сбой в регистре защиты;
- РА — сбой во входном информационном регистре РА блока БА;
- РВ — сбой во входном информационном регистре РВ блока БА;
- БА — сбой на выходных шинах БА;
- РМН — сбой в адресном регистре РМН оперативной памяти;
- РНЗ — сбой в регистрах РН, РЗ при считывании информации из оперативной памяти;
- РМК — сбой в информационном регистре микрокоманды;
- РАПП — сбой в адресном регистре постоянной памяти.

Индицируются состояния управляющих триггеров, отвечающие следующим состояниям процессора:

- ЦП — процессор никаких операций не выполняет;
- ОСТ АДР — адрес, установленный на наборе АДРЕС ПАМЯТИ, равен адресу постоянной или оперативной памяти (в зависимости от положения переключателя СРАВНЕНИЕ АДРЕСОВ);
- АПРС — установлен запрос на обмен данными с селекторным каналом (первым или вторым),
- МПРС — установлен запрос на выполнение обслужи-

вающей микропрограммы мультиплексного или селекторного каналов.

Пульты других моделей также включают в свой состав панели индикации и панели управления. Принципы их построения и использования те же, что и для рассмотренного пульта управления, но они всегда отражают особенности соответствующей модели, ее комплектацию и режимы работы. Так, например, панель индикации пульта управления ЕС-1050 не содержит в своем составе средств индикации состояния регистров постоянной памяти, не используемой в этой модели, но обеспечивает постоянную индикацию содержимого регистра слова состояния программы. Для проверки содержимого регистров процессора используются четыре восьмибайтных индикационных регистра, каждый из которых обслуживает девять регистров процессора. Для индикации состояния каналов используется один общий индикационный регистр с переключателем. Кроме того, с пульта управления можно непосредственно адресовать одно из внешних устройств.

ВЗАИМОДЕЙСТВИЕ ОСНОВНЫХ УСТРОЙСТВ ЭВМ

Выполнение любых операций в ЭВМ Единой системы происходит при тесном взаимодействии их основных устройств, входящих в состав процессора и оперативной памяти и составляющих так называемое центральное ядро системы. При выполнении операций ввода-вывода кроме этих устройств в работу включаются каналы и устройства ввода-вывода, которые более подробно рассматриваются в последующих главах. Здесь же основное внимание уделяется взаимодействию устройств центрального ядра для случаев наиболее часто используемых режимов работы центрального устройства управления, включая реализацию пультовых процедур, когда инициирование действий производится вручную с помощью соответствующих кнопок и переключателей пульта управления.

8.1. ВЫПОЛНЕНИЕ ПУЛЬТОВЫХ ПРОЦЕДУР

Элементы и схемы пультов управления всех моделей ЭВМ Единой системы позволяют выполнять различные действия, связанные не только с проведением профилактических работ, но и непосредственно с реализацией рабочих программ. Использование элементов пульта управления допускается только в состоянии останова, когда в процессоре не выполняются никакие команды и не обслуживаются прерывания. Исключение составляют элементы, позволяющие переводить машину, т. е. данную модель, в состояние останова.

С помощью элементов и схем пульта управления можно инициировать и в конечном итоге выполнять следующие действия:

— осуществлять приведение машины в исходное состояние (гашение);

— производить первоначальную загрузку программы (загрузку начальной программы);

— устанавливать адрес команды текущего слова состояния программы (для ЕС-1020 — в регистре РМФЕ);

— производить запись информации в регистры процессора, в том числе и регистры местной памяти, а также ячейки оперативной памяти;

— выводить на индикацию содержимое регистров процессора и ячеек памяти;

- задавать реакцию машины при сравнении адресов;
- задавать реакцию машины на сбой;
- устанавливать режим работы процессора;
- вводить заданную информацию в адресный регистр постоянной памяти (для ЕС-1020 и ЕС-1030);
- вводить набираемые на пульте данные в регистр слова состояния программы (для ЕС-1030 и ЕС-1050);
- производить гашение сигналов сбоев;
- производить пуск и останов процессора, а также выполнять ряд других действий.

В машинах с микропрограммным управлением, т. е. в ЕС-1020 и ЕС-1030, гашение, первоначальная (начальная) загрузка программы, занесение адреса текущей команды или всего ССП, запись и считывание информации, а также обслуживание таймера, выполняются по микропрограммам, а остальные функции реализуются схемным путем, т. е. аппаратно. Выполнение микропрограмм производится после нажатия соответствующих кнопок или клавиш на пульте управления. В ЕС-1050 все действия, связанные с выполнением пультовых процедур, реализуются схемным путем. При работе на пульте управления любой модели блокируются программные прерывания, машинные ошибки, сбой по защите и адресации.

Гашение переводит машину (систему) в состояние, исходное для начала обработки информации. Это такое состояние, при котором может быть начата обработка информации без устранения неисправностей и без предварительных проверок машины. Гашение всегда выполняется при включении питания, а также при нажатии кнопок «ГАШЕНИЕ» и «ЗАГРУЗКА» («НЗП» в ЕС-1050). При этом производится останов генератора импульсов и вырабатывается сигнал аппаратного гашения, по которому сбрасываются в нулевое состояние все регистры процессора, гасятся каналы и внешние устройства, аннулируются все прерывания, ждущие своей очереди. Далее в зависимости от используемого в данной модели принципа управления аппаратно или по специальной микропрограмме гашения производится проверка регистров процессора, корректируются контрольные разряды ячеек памяти и осуществляется перевод машины в состояние останова.

Первоначальная (начальная) загрузка программы. В процессе работы машины по решению набора задач может оказаться, что содержимое оперативной памяти или слово состояния программы непригодны для дальнейшего использования; такое состояние характерно перед вводом и запуском программ. Для смены содержимого памяти, а также ССП и производится с пульта управления первоначальная загрузка программы.

Рассмотрим процесс первоначальной загрузки программы на примере ЕС-1020, используя схему, приведенную на рис. 7.26. Адрес внешнего устройства, с которого вводится первоначальная программа, задается на переключателях З, И, К. При нажатии

кнопки «ЗАГРУЗКА» выполняется гашение машины, а затем — микропрограмма ПЗП, которая формирует управляющее слово канала (команду канала), которое отсылается в канал, к которому подключено выбранное внешнее устройство. Канал по этому управляющему слову передает в оперативную память машины первый блок информации, состоящий из 24 байт. Эти начальные байты записываются в ячейках 0—23 основной оперативной памяти. Первые восемь байт составляют слово состояния программы ПЗП, а последующие — первое и второе управляющие слова канала (УСК) для ПЗП. Дальнейший ввод информации, т. е. выполнение операции ввода, производится под управлением первого УСК ПЗП, расположенного в ячейках 8—15 ОП. Второе УСК ПЗП (ячейки 16—23 ОП) используется в случае реализации цепочки данных или команд. По окончании первоначальной загрузки программы управление передается новому слову состояния программы, находящемуся в ячейках 0—7 оперативной памяти.

Во время выполнения первоначальной загрузки программы ключи защиты памяти, программно-управляемые прерывания и индикация неверной длины игнорируются. При нажатии кнопки «ЗАГРУЗКА» подсвечивается табло «ЗАГРУЗКА»; гашение этого табло производится после успешного завершения всех операций по первоначальной загрузке программы.

Все действия по первоначальной загрузке программы, или загрузке начальной программы, во всех моделях ЭВМ Единой системы выполняются в одной и той же последовательности с записью первых 24 байт в ячейки 0—23 оперативной памяти. Но в ЕС-1050 все сигналы, управляющие действиями по загрузке программы, формируются схемным путем. Кроме того, адрес внешнего устройства, с которого производится прием информации, набирается на отдельной группе переключателей АДРЕС УСТРОЙСТВА и нажимается кнопка «НЗП».

При нажатии кнопки «НЗП» с помощью триггеров управления формируется общий сигнал сброса, направляемый во все устройства, входящие в систему ЕС-1050. Для обеспечения нормальной работы канала, связанного с выбранным внешним устройством, с помощью счетчика таймера (см. п. 7.6) производится задержка формирования последующих сигналов на 100 мс. После этого информация, набранная на переключателях пульта управления, вводится в РАТ [13—23] блока адреса результата (см. рис. 7.17). По РАТ [13—15] выбирается канал, к которому по радиальной линии направляется сигнал начала выполнения операции ввода; содержимое РАТ [16—23] направляется в выбранный канал, где используется для формирования управляющего слова. Далее канал через блок управления памятью вводит начальные байты в ячейки 0—23 оперативной памяти; адреса формируются каналом по исходному управляющему слову. После этого канал выбирает из ячеек 8—15 УСК, по которому завершает выполнение операции ввода.

По окончании работы канала блок выборки команд посылает запрос в БУП на выборку ССП. Так как при нажатии кнопки «НЗП» был произведен общий сброс и все адресные регистры «обнулены», то ССП выбирается из нулевой ячейки, куда оно было записано при загрузке программы. Новое слово состояния программы вводится в РССП; машина готова к выполнению новой программы.

Рассмотренная пультовая процедура, как явствует из вышеизложенного, не имеет принципиальных различий для машин Единой системы с различным построением центральных устройств управления. Это относится и к другим пультовым процедурам, что позволяет их рассмотреть на примере машины ЕС-1020, пульт управления которой достаточно подробно был описан в п. 7.7.

Адрес команды в текущее ССП в машине ЕС-1020 заносится микропрограммно при взаимодействии схем только процессора. Заносимый адрес набирается на переключателях Е—К и нажимается кнопка «ЗАНЕСЕНИЕ АК». При нажатии этой кнопки в регистр адреса постоянной памяти вводится адрес первой микрокоманды соответствующей микропрограммы. Реализация этой микропрограммы блоком центрального управления приводит к побайтной передаче адреса, установленного на переключателях Е—К набора АДРЕС КОМАНДЫ, в счетчик команд текущего ССП (регистр РМФЕ) и в регистр индикации. По состоянию регистра индикации осуществляется визуальный контроль правильности ввода. После ввода в указанные регистры последнего байта адреса команды реализуемая микропрограмма обеспечивает формирование управляющих сигналов, переводящих машину в состояние останова.

Запись информации с пульта управления позволяет вручную изменять содержимое ячеек памяти, а также регистров процессора. Для выполнения операции записи с пульта управления процессор переводится в состояние останова и осуществляется запуск микропрограммы, имеющей обозначение ОБСП (обслуживание пульта). Адрес требуемой ячейки памяти устанавливается на переключателях набора АДРЕС ПАМЯТИ, тип памяти — на переключателе ТИП ПАМЯТИ. Записываемая информация задается на переключателях И, К. После этого нажимается кнопка «ЗАПИСЬ», что вызывает выполнение микропрограммы ОБСП, которая осуществляет запись информации в указанную ячейку памяти или регистр процессора и останов машины.

Для записи информации в регистры процессора переключатель ТИП ПАМЯТИ устанавливается в положение ЦП, номер регистра указывается на переключателе Д, информация при записи в регистры РМФЕ, РГРИ, РПТУ — на переключателях Е—К, а при записи в однобайтные регистры — на переключателях И—К.

Иногда возникает необходимость записи одной и той же информации в несколько ячеек памяти, имеющих последовательные адреса. Для выполнения такого комплекса действий начальный

адрес передается в регистр РМФЕ (как при занесении адреса команды текущего слова состояния программы), а конечный адрес задается на переключателях А—Д. Переключатель СРАВНЕНИЕ АДРЕСОВ устанавливается в положение ОСТАНОВ ОП, переключатель РЕЖИМ РАБОТЫ — в положение ОП. При нажатии кнопки «ЗАПИСЬ» начинает выполняться микропрограмма ОБСП, которая обеспечивает последовательную запись в заданные ячейки ОП байта информации, установленного на переключателях И—К, и последующий останов машины. Кнопка «ЗАПИСЬ» всегда подсвечивается при выполнении операции записи любого вида.

При выполнении записи в последовательно расположенные ячейки памяти с блоком центрального управления и блоком управления памятью активно взаимодействует арифметическо-логический блок. После записи в каждую ячейку он обеспечивает увеличение на единицу содержимого регистра РМФЕ и, кроме того, сравнение вновь полученного адреса с адресом, набранным на переключателях А—Д.

Считывание информации. Операция считывания (чтения) информации позволяет передавать в регистр индикации содержимое любой ячейки памяти, регистров процессора и каналов, не имеющих постоянной индикации. Посредством переключателя ТИП ПАМЯТИ указывается один из каналов (КМ, КС1 или КС2), память (ОП, МП, ЛП или ПЗ) или регистры процессора (ЦП). Если в регистр индикации необходимо передать содержимое регистра процессора или канала, то на переключателе Д указывается номер регистра. Если же в регистр индикации необходимо передать содержимое ячейки памяти, то на переключателях А—Д устанавливается адрес ячейки памяти. При нажатии кнопки «ЧТЕНИЕ» запускается микропрограмма ОБСП, которая обеспечивает выполнение заданной операции и затем перевод машины в состояние останова. Считывание содержимого ряда последовательных ячеек памяти осуществляется так же, как и запись в последовательный ряд ячеек; отличие заключается только в том, что нажимается кнопка «ЧТЕНИЕ», а не кнопка «ЗАПИСЬ». При выполнении операции считывания любого вида кнопка «ЧТЕНИЕ» всегда подсвечивается.

Модификация таймера. Таймер, или счетчик временных интервалов, представляет собой одно полноразрядное (четырёхбайтное) слово, которое располагается в ячейках оперативной памяти с адресами 50—53 и периодически изменяется, т. е. модифицируется, в соответствии с содержимым некоторого физического счетчика временных интервалов. В машине ЕС-1020 используется только интервальный таймер, позволяющий регистрировать время выполнения рабочей программы и формировать сигналы для управления ходом выполнения программы по истечении заданного промежутка времени; схемы таймера непосредственно связаны с пультом управления. Управление интервальным таймером — микропрограмм-

ное. При модификации таймер рассматривается как целое число и обрабатывается по правилам арифметики для чисел с фиксированной запятой. Полный цикл таймера составляет 15,5 ч.

Начальный счет временных интервалов производится счетчиком, расположенным в пульте управления (четыре разряда регистра РР9); на счетчик подаются импульсы, формируемые по напряжению сети переменного тока. Этот счетчик, если его содержимое не равно нулю, вызывает запрос на обновление таймера, рассматриваемый как запрос на прерывание. Запросы на прерывание по внешним причинам, по вводу-выводу и на обновление таймера обслуживаются в конце выполнения каждой команды процессора. Так как таймер имеет самый низкий приоритет, то запрос на обновление таймера не всегда может быть удовлетворен в течение 20 мс (20 мс — длительность подсчитываемых счетчиком временных интервалов при частоте напряжения сети переменного тока $f = 50$ Гц). Поэтому счетчик имеет возможность накапливать до 15 интервалов времени по 20 мс каждый, т. е. $T_{сч} = 300$ мс. Если содержимое счетчика интервалов времени не будет учтено в течение 300 мс, то для интервального таймера оно будет потеряно. Отметим, что таймер всегда обновляется при переходе процессора в состояние ожидания или в состояние останова.

Собственно модификация, или обновление, таймера заключается в считывании содержимого ячеек 50—53 оперативной памяти и вычитании из него содержимого счетчика интервалов времени, умноженного на 6 (при частоте сети 60 Гц содержимое счетчика умножается на 5). Сначала в АЛБ производится операция умножения при однократном вводе операндов, так как они «вписываются» в формат одного байта. Далее из оперативной памяти считывается младшее полуслово таймера; из младшего байта производится вычитание числа-корректора, полученного в результате выполнения операции умножения. При возникновении отрицательного переноса в старшие байты производится уменьшение на единицу значения второго и, если необходимо, последующих байт таймера, выбираемых из оперативной памяти в составе старшего полуслова.

Результат вычитания записывается в ячейки таймера, а процессор переходит к выполнению прерванной программы. Если при вычитании значение таймера становится отрицательным, то формируется запрос на внешнее прерывание по таймеру (не путать с запросом на обновление таймера). Интервальный таймер может быть заблокирован при помощи кнопки «БЛОКИРОВКА ТАЙМЕРА». Если эта кнопка подсвечивается, то таймер заблокирован, т. е. обновление его значения не производится.

Сравнение адресов. Переключатель СРАВНЕНИЕ АДРЕСОВ позволяет сравнивать адрес, установленный на переключателях АДРЕС ПАМЯТИ, с адресом постоянной или оперативной памяти. В зависимости от положения этого переключателя определяется реакция процессора на сравнение адресов.

При равенстве адреса, установленного на переключателях, адресу оперативной памяти возможны режимы останова, останова по записи и перехода. При останове подсвечиваются табло «РУЧНАЯ РАБОТА», кнопка «ОСТАНОВ» и горит лампочка «ОСТ. АДР.»; продолжение автоматической работы процессора вызывается нажатием кнопки «ПУСК». Останов по записи происходит в случае совпадения адресов при выполнении микрокоманды записи в ОП. В этом случае горит лампочка «ОСТ. АДР.», в информационных регистрах ОП индицируется записываемая информация, а в адресном регистре РМН — ее адрес; продолжение автоматической работы процессора — при нажатии кнопки «ПУСК». В режиме перехода при совпадении адресов сначала осуществляется останов процессора, если он работал в автоматическом режиме, далее в регистре РМФЕ передается адрес, установленный на переключателях Е—К; процессор продолжает автоматическую работу, начиная с команды, адрес которой передан в РМФЕ. Если переключатель РЕЖИМ РАБОТЫ был установлен в положение КОМАНДА или ОП, то происходит зацикливание операции с пульта управления.

При равенстве адреса, установленного на переключателях, адресу постоянной памяти возможны режимы останова, перехода и синхроимпульса. В случае останова горит лампочка «ОСТ. АДР.»; микрокоманда, которая должна выполняться при продолжении работы ЦП, и ее адрес индицируются на пульте управления; продолжение работы ЦП — при нажатии кнопки «ПУСК». В случае перехода информация, набранная на переключателях Ж—К, передается в адресный регистр постоянной памяти; продолжение работы осуществляется с микрокоманды, выбираемой по введенному в РАПП адресу. В случае формирования синхроимпульса по совпадению адресов изменений в работе процессора не происходит; этот режим используется при наладке и проверке работы процессора.

Занесение адреса микрокоманды в адресный регистр постоянной памяти производится следующим образом. Переключатель РЕЖИМ РАБОТЫ устанавливается в положение МИКРОКОМАНДА, на переключателях Ж—К набирается адрес микрокоманды и нажимается кнопка «ПУСК». При нажатии этой кнопки формируется управляющий сигнал, который производит передачу информации с выходов шифраторов переключателей в РАПП. Таким образом эта пультовая процедура выполняется схемным путем.

Контроль работы процессора осуществляется по сбоям. Реакция процессора на сбой устанавливается переключателем КОНТРОЛЬ. Если этот переключатель установлен в положение АВТОМАТ, то при возникновении сбоя производится переход к выполнению диагностической программы ОБРАБОТКА СБОЯ. Если переключатель находится в положении ОСТАНОВ, то при возникновении сбоя осуществляется останов процессора с индикацией ошибки в регистре ошибок процессора или на индикаторах сбоя каналов.

Если же переключатель установлен в положение БЛОКИРОВКА, то сбои игнорируются; при их возникновении процессор продолжает работу в автоматическом режиме.

Пуск и останов процессора. Состояния процессора практически отражают состояния машины в целом. Поэтому можно рассматривать пуск и останов процессора как пуск и останов всей машины, тем более, что ни одно устройство не может находиться в состоянии работы, если процессор остановлен.

Начало процесса выполнения операций в ЦП тем способом, который задан переключателем РЕЖИМ РАБОТЫ, определяется моментом нажатия кнопки «ПУСК». При нажатии этой кнопки формируется сигнал пуска, который сбрасывает в нулевое состояние триггер останова. Процессор переходит либо к автоматическому выполнению последовательности команд (режим «Автомат»), либо к выполнению одной макрокоманды (режим «Команда»), либо к выполнению одной микрокоманды, адрес которой находится в адресном регистре постоянной памяти (режим «Микрокоманда»).

Перевод процессора в состояние останова производится после включения питания при гашении всех устройств, останова по адресу, по окончании выполнения команды, если переключатель РЕЖИМ РАБОТЫ установлен в положение КОМАНДА, и при нажатии кнопки «ОСТАНОВ». Собственно переход в состояние останова происходит тогда, когда закончено выполнение текущей команды и нет запросов на прерывание. При останове триггер останова перебрасывается в единичное состояние; в счетчике адресов команд и регистре индикации находится адрес команды, с выполнения которой будет продолжена работа при нажатии кнопки «ПУСК». В состоянии останова обеспечивается подсветка кнопки «ОСТАНОВ» и табло «РУЧНАЯ РАБОТА».

8.2. РАБОТА УСТРОЙСТВ ПРИ ВЫПОЛНЕНИИ АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ ОПЕРАЦИЙ

Арифметические и логические операции при действиях над числами с фиксированной или плавающей запятой выполняются в большинстве случаев по инструкциям (командам) форматов RR и RX. Реализация инструкций этих форматов связана с работой различных комплексов блоков и устройств машины, так как при формате RR используется только местная память, а при формате RX — и местная, и оперативная память, причем адрес ячейки ОП образуется путем модификации и смещения базового адреса, выбираемого из соответствующей ячейки местной памяти. Наиболее полно эти различия проявляются в случае использования схемных принципов управления и построения местной памяти в виде совокупности регистров, относимой к центральному устройству управления. Поэтому дальнейшее изложение материала базируется на принципах работы блоков ЦУУ машины ЕС-1050, описанных в пп. 7.3—7.5.

Реализация инструкций формата RR. По инструкциям формата RR выполняются арифметические и логические операции над числами с фиксированной запятой, имеющими разрядность слова, а также арифметические и логические операции над числами с плавающей запятой, имеющими разрядность слова или двойного слова. Оба операнда выбираются из регистров местной памяти непосредственно по адресам, содержащимся в коде инструкции; результат операции, как правило, записывается по адресу первого операнда. Признак длины операндов задается кодом выполняемой операции.

При рассмотрении взаимодействия узлов и блоков процессора в случае реализации инструкций формата RR полагаем, что участок программы уже выбран из оперативной памяти на регистры блока выборки команд и произведена распаковка команды, код которой зафиксирован в РКР. В связи с этим для схемы, поясняющей работу узлов и блоков процессора при выполнении инструкций формата RR (см. рис. 8.1), в качестве начального узла принят именно РКР, т. е. регистр распакованной команды. Для схемы рис. 8.1 приняты те же обозначения, что и для схем, рассмотренных в п. 7.3; символика межузловых соединений оставлена прежней, но сами связи упрощены в соответствии с упрощением и объединением отдельных узлов. Под АЛБ, т. е. арифметическо-логическим блоком, как и прежде, понимается объединение узлов и блоков, непосредственно выполняющих арифметические и логические операции.

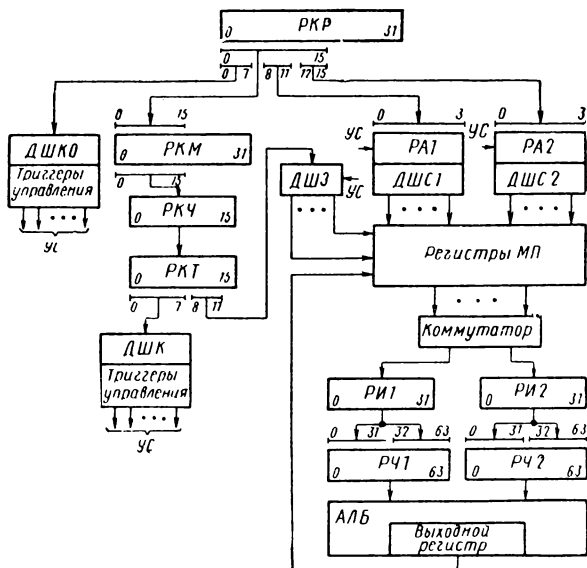


Рис. 8.1. Связи узлов и блоков процессора при выполнении инструкций формата RR

На первом этапе выполнения арифметической или логической операции по инструкции формата RR производится выдача из РКР кода операции в блок центрального управления, а адресов — в блок местной памяти; кроме того, весь код инструкции из РКР передается во входной регистр команд блока центрального управления, т. е. в РКМ. При этом используются различные группы выходных вентилях РКР, отвечающие частям кода инструкции, который располагается в старших разрядах РКР, т. е. РКР [0—15]. Действия первого этапа описываются соотношениями:

$$\text{РКМ} [0 — 15] : = \text{РКР} [0 — 15],$$

$$\text{РКР} [0 — 7] \rightarrow \text{ДШКО},$$

$$\text{РА1} : = \text{РКР} [8 — 11], \text{РА2} : = \text{РКР} [12 — 15].$$

Дешифратор кода операций (ДШКО) блока центрального управления расшифровывает операционную часть инструкции и обеспечивает формирование необходимых сигналов управления; одновременно устанавливается в состояние 1 триггер признака формата RR. Передача кода инструкции в РКМ необходима для того, чтобы освободить РКР для действий по распаковке очередной команды и сохранить этот код для обеспечения действий на завершающем этапе по записи в местную память результата выполнения операции. Ввод адресных частей кода инструкции в регистры адреса РА1 и РА2 блока местной памяти обеспечивает в дальнейшем выборку операндов.

Второй этап выполнения операции по инструкции формата RR является подготовительным. В блоке центрального управления осуществляются так называемые промежуточные действия по передаче кода инструкции из РКМ в РКЧ (эти действия никакого влияния на ход выполнения операции не оказывают), а в блоке местной памяти осуществляется выборка операндов из РОН или РПЗ и размещение их в выходных регистрах РИ1 и РИ2. Если обозначать содержимое ячейки памяти, выбираемое по адресу, находящемуся, например, в РА1, символами ((РА1)), то действия второго этапа можно описать следующими соотношениями:

$$\text{РКЧ} : = \text{РКМ} [0 — 15],$$

$$\text{РИ1} : = ((\text{РА1})), \text{РИ2} : = ((\text{РА2})).$$

Передача содержимого старших разрядов РКМ в соответствующие разряды РКЧ имеет своей целью подготовку действий последующих этапов, когда в работу включается блок адреса результата. Выборка операндов по адресам, находящимся в РА1 и РА2, производится в обычной последовательности при расшифровке адресов дешифраторами считывания ДШС1 и ДШС2. Во всех случаях из накопителя местной памяти, т. е. ее регистров, может считываться только 32-разрядное слово; выходные регистры блока местной памяти РИ1 и РИ2 имеют такую же разрядность. Это всегда учиты-

вается при выполнении операций над двойными словами. Отметим, что при считывании содержимое соответствующей ячейки местной памяти сохраняется; это также учитывается в тех случаях, когда выполняемая операция не требует записи результата по адресу первого операнда.

Третий этап выполнения операции по инструкции формата RR включает действия, производимые для формирования операндов. Если операция выполняется над операндами, имеющими формат слова, то действия третьего этапа сводятся к простой пересылке содержимого регистров РИ1 и РИ2 в старшие разряды РЧ1 и РЧ2, т. е. регистров блока выборки данных. Можно записать, что в этом случае реализуются соотношения:

$$РЧ1 [0 — 31] : = РИ1,$$

$$РЧ2 [0 — 31] : = РИ2.$$

Если же операция выполняется над двойными словами, то на третьем этапе выполняются дополнительные действия, включая обращение к местной памяти. Когда операнды представляют собой двойные слова, первое обращение к местной памяти всегда осуществляется по четным адресам. Поэтому для формирования адресов вторых частей операндов достаточно в младшие разряды РА1 и РА2 ввести единицы. Последовательность выполнения дополнительных действий третьего этапа выражается следующими соотношениями:

$$РА1 : = РА1 + 1, \quad РА2 : = РА2 + 1,$$

$$РИ1 : = ((РА1)), \quad РИ2 : = ((РА2)),$$

$$РЧ1 [32 — 63] : = РИ1, \quad РЧ2 [32 — 63] : = РИ2;$$

вместо первых двух соотношений, учитывая изложенное выше, можно записать:

$$РА1 [3] : = 1, \quad РА2 [3] : = 1.$$

Выборка вторых частей операндов из местной памяти ничем не отличается от выборки их первых частей. Передача же содержимого РИ1 и РИ2 в регистры блока выборки данных производится через другие группы вентиляей. Как правило, вторые части операндов вводятся в младшие разряды регистров РЧ1 и РЧ2. При выполнении отдельных операций содержимое РИ1 после второй выборки чисел из местной памяти может передаваться в старшие разряды РЧ1.

На четвертом этапе производится выполнение действий над операндами в арифметическо-логическом блоке и запись результата в местную память. Для этого содержимое РЧ1 и РЧ2 передается в АЛБ, а код инструкции из РКЧ принимается в РКТ блока адреса результата. После этих действий блоки центрального управления и выборка данных могут переключаться на работу по обработке кода

очередной инструкции. Операционная часть кода реализуемой инструкции используется для формирования управляющих сигналов, обеспечивающих действия четвертого этапа, включая запись результата выполнения операции в местную память. Последовательность действий четвертого этапа выражается следующими соотношениями:

$$\begin{aligned} \text{PC1} &\rightarrow \text{АЛБ}, \text{ PC2} \rightarrow \text{АЛБ}, \\ \text{РКТ} &:= \text{РКЧ}, \\ \text{РКТ} [0 - 7] &\rightarrow \text{ДШК}, \\ \text{РОП} &\rightarrow \text{РВ}, \text{ РКТ} [8 - 11] \rightarrow \text{ДШЗ}, \\ ((\text{РКТ} [8 - 11])) &:= \text{РВ} [0 - 31], \end{aligned}$$

где РВ — выходной регистр арифметическо-логического блока; РОП — результат выполняемой операции.

Необходимо отметить, что после передачи информации из PC1 и PC2 в АЛБ, а из РКЧ в РКТ схемы блока адреса результата и блока местной памяти не выполняют активных действий до завершения операции в арифметическо-логическом блоке. При записи результата операции в выходной регистр АЛБ содержимое РКТ [8—11], т. е. практически адрес первого операнда, подается на дешифратор записи блока местной памяти. После этого в ячейку местной памяти, ранее хранившую значение первого операнда, записывается содержимое старших разрядов выходного регистра АЛБ; этому отвечает запись

$$((\text{РКТ} [8 - 11])) := \text{РВ} [0 - 31].$$

Если результат выполнения заданной операции является двойным словом, то его запись в местную память производится в два приема. После вышеописанных действий осуществляется изменение адреса записи на единицу, что приводит к выборке соседней нечетной ячейки в блоке местной памяти (считается, что начальный адрес первого операнда — число четное); для РПЗ, например, это означает открывание группы вентилях для его младших разрядов (см. рис. 7.13). Далее содержимое РВ [32—63] передается для хранения в эту ячейку МП. На этом заканчивается реализация инструкции формата RR.

Реализация инструкций формата RX. По инструкциям формата RX выполняются операции над числами с фиксированной и плавающей запятой всех возможных форматов; это означает, что числа с фиксированной запятой могут иметь формат слова или полуслова, а числа с плавающей запятой — формат слова или двойного слова. Характерной особенностью инструкций формата RX является то, что второй операнд выбирается из оперативной памяти по модифицированному (индексируемому) адресу, причем базовый адрес (база) и индекс располагаются в местной памяти, а смещение содержится в коде инструкции. Первый операнд выбирается из

местной памяти; запись результата выполнения операции производится по адресу первого операнда. Как и в инструкциях формата RR, признак длины операндов задается кодом выполняемой операции.

При рассмотрении взаимодействия основных устройств ЭВМ и их блоков полагаем, как и прежде, что распаковка команд уже произведена и код реализуемой инструкции располагается в РКР блока выборки команд. В этом случае схема, поясняющая работу блоков и устройств при выполнении операций по инструкциям формата RX и приведенная на рис. 8.2, в качестве начального узла имеет РКР, т. е. регистр распакованной команды. Для этого рисунка приняты обозначения, использовавшиеся в п. 7.3; через РВ обозначен выходной регистр арифметическо-логического блока. Отметим, что схема, приведенная на рис. 8.2, отвечает предельному набору микроопераций, выполняемых по инструкциям формата RX; для некоторых операций, отвечающих инструкциям этого формата, схема связей и набор микроопераций упрощаются, так как они предусматривают обработку только второго операнда.

В общем случае реализация инструкции формата RX осуществляется поэтапно с выполнением на каждом этапе вполне определенного комплекса действий. Некоторые действия подобны дей-

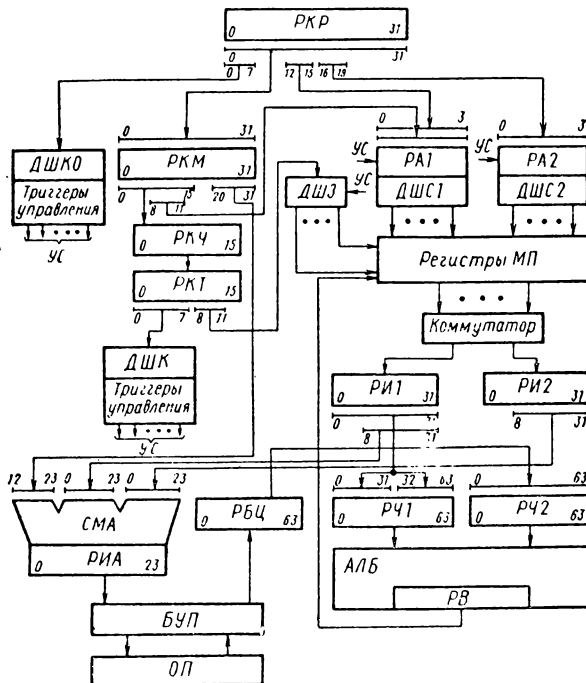


Рис. 8.2. Связи блоков и устройств при выполнении инструкций формата RX

ствиям, выполняемым при реализации инструкций формата RR, а некоторые полностью совпадают с ними.

На первом этапе, как и в случае реализации инструкций формата RR, производится передача кода инструкции и ее частей из РКР в регистры блока центрального управления и блока местной памяти при использовании различных групп выходных вентилях для РКР. Инструкция формата RX является 32-разрядной, поэтому ее код во входной регистр блока центрального управления выдается из РКР через полноразрядную группу вентилях. Поскольку выборка второго операнда связана с выполнением более сложных действий, чем выборка первого операнда, причем требующих больших затрат времени, то на первом этапе в адресные регистры блока местной памяти передаются из РКР адреса базы и индекса. После выдачи кода реализуемой инструкции и ее частей в другие блоки регистр распакованной команды считается свободным, и на уровне К возможна обработка очередной инструкции. Действия первого этапа описываются соотношениями:

$$\begin{aligned} \text{РКМ} &:= \text{РКР}, \\ \text{РКР} [0 - 7] &\rightarrow \text{ДШКО}, \\ \text{РА1} &:= \text{РКР} [12 - 15], \text{РА2} := \text{РКР} [16 - 19]. \end{aligned}$$

Передача содержимого восьми старших разрядов РКР на дешифратор кода операций обеспечивает в конечном итоге формирование необходимых управляющих сигналов и установку в состояние 1 триггера признака формата RX. Передача кода инструкции в РКМ позволяет не только освободить уровень К для обработки очередной инструкции, но и обеспечить выполнение действий на последующих этапах, в том числе и выборку первого операнда. Ввод содержимого РКР [12—15] и РКР [16—19] в РА1 и РА2 соответственно позволяет осуществлять в дальнейшем выборку базы и индекса из ячеек (регистров) местной памяти.

Основные действия второго этапа связаны с подготовкой информации, необходимой для формирования адреса второго операнда. Поскольку база и индекс всегда являются 24-разрядными, то для их выборки достаточно произвести по одному обращению к местной памяти. Используя уже принятые обозначения, запишем следующие соотношения, отражающие действия второго этапа:

$$\text{РИ1} := ((\text{РА1})), \text{РИ2} := ((\text{РА2})).$$

Передача кода базы в РИ1 и кода индекса в РИ2 производится через полноразрядные группы входных вентилях. Разрядность этих кодов учитывается при их размещении в регистрах общего назначения блока местной памяти. При выборке содержимое соответствующих регистров МП всегда сохраняется.

На третьем этапе производится формирование адреса второго операнда путем суммирования базы, находящейся в РИ1, индекса, находящегося в РИ2, и смещения, которое является частью кода

инструкции и располагается к началу действий третьего этапа в младших разрядах входного регистра РКМ блока центрального управления. Суммирование производится в блоке сумматора адреса, показанном на рис. 8.2 в упрощенном виде. Результат сложения, т. е. исполнительный адрес второго операнда устанавливается в выходном регистре блока, т. е. в РИА. Все действия третьего этапа можно описать соотношением:

$$\text{РИА} := \{\text{РИ1} [8 - 31] + \text{РИ2} [8 - 31] + \text{РКМ} [20 - 31]\}.$$

Наиболее сложный комплекс действий выполняется на четвертом этапе, когда осуществляется выборка операндов и их подготовка для непосредственного выполнения операции в арифметическо-логическом блоке. Все действия по выборке первого операнда выполняются блоками только процессора, а для выборки второго операнда необходимо подключение оперативной памяти с блоками управления. На этом же этапе производятся и промежуточные действия по передаче старшего полуслова кода инструкции из РКМ в РКЧ. Если операнды для выполняемой операции представляются двойными словами, то комплекс действий, выполняемых на четвертом этапе, отражается следующими обобщенными соотношениями:

$$\begin{aligned} \text{РИА} &\rightarrow \text{БУП}, \\ \text{РА1} &:= \text{РКМ} [8 - 11], \\ \text{РИ1} &:= ((\text{РА1})), \\ \text{РЧ1} [0 - 31] &:= \text{РИ1}, \quad \text{РА1} := \text{РА1} + 1, \\ \text{РИ1} &:= ((\text{РА1})), \\ \text{РЧ1} [32 - 63] &:= \text{РИ1}, \quad \text{РКЧ} := \text{РКМ} [0 - 15], \\ \text{РБЦ} &:= ((\text{РИА})), \quad \text{РЧ2} := \text{РБЦ}. \end{aligned}$$

Выборка второго операнда обеспечивается действиями, начинаемыми пересылкой его исполнительного адреса из РИА блока сумматора адреса в блок управления памятью (БУП). Далее этот запрос на обслуживание оперативной памятью обрабатывается в БУП согласно принятой приоритетности устройств машины (см. п. 7.5). Обработка запроса и собственно выборка второго операнда по заданному адресу из ОП может потребовать времени порядка нескольких десятых микросекунды или еще больше. Поэтому параллельно действиям по выборке второго операнда производятся действия по выборке первого операнда и его размещению в выходном регистре РЧ1 блока выборки данных.

Выборка первого операнда начинается с передачи содержимого РКМ [8—11], т. е. его адреса, в РА1 блока местной памяти. Далее содержимое регистра МП с четным номером выдается в выходной регистр РИ1 блока местной памяти. Из регистра РИ1 старшее слово первого операнда записывается в старшие разряды РЧ1, а содержимое РА1 увеличивается на единицу, что дает адрес млад-

шего слова операнда. Выборка этого слова и его размещение в младших разрядах РЧ1 производятся обычным порядком. Теперь код инструкции почти полностью реализован, но его старшее полуслово еще потребуется на заключительном этапе выполнения операции. Поэтому осуществляется передача содержимого РКМ [0—15] в РКЧ, т. е. выходной регистр команд блока центрального управления.

После обработки запроса процессора на выдачу из ОП двойного слова по адресу, сформированному в РИА, блок управления памятью в порядке, принятом для удовлетворения запросов, обеспечивает выборку из ОП кода второго операнда. Этот код направляется в РБЦ блока выборки данных, а затем переписывается в РЧ2 этого же блока. Фиксацией кода второго операнда в РЧ2 заканчиваются действия четвертого этапа.

На пятом этапе производится выполнение действий над операндами в арифметическо-логическом блоке и запись результата по адресу первого операнда. Для этого содержимое РЧ1 и РЧ2 передается в АЛБ, а содержимое РКЧ принимается в РКТ блока адреса результата. Действия пятого этапа выполнения инструкции формата RХ аналогичны действиям четвертого этапа выполнения инструкции формата RR, поэтому они выражаются следующими соотношениями:

$$\begin{aligned} \text{РЧ1} &\rightarrow \text{АЛБ}, \text{ РЧ2} \rightarrow \text{АЛБ}, \\ \text{РКТ} &: = \text{РКЧ}, \\ \text{РКТ} [0 - 7] &\rightarrow \text{ДШК}, \\ \text{РОП} &\rightarrow \text{РВ}, \text{ РКТ} [8 - 11] \rightarrow \text{ДШЗ}, \\ ((\text{РКТ} [8 - 11])) &: = \text{РВ} [0 - 31], \\ ((\text{РКТ} [8 - 11] + 1)) &: = \text{РВ} [32 - 63], \end{aligned}$$

где РОП — результат выполняемой по инструкции формата RХ операции.

После передачи содержимого РКЧ в РКТ, а содержимого РЧ1 и РЧ2 в АЛБ блоки выборки данных и центрального управления считаются свободными и могут переключаться на работу по обработке кода очередной инструкции. Передача результата выполнения операции в местную память производится в два приема по начальному адресу, содержащемуся в РКТ [8—11]. Записью результата в МП заканчивается реализация инструкции формата RХ.

8.3. РАБОТА УСТРОЙСТВ ПРИ ВЫПОЛНЕНИИ ВСПОМОГАТЕЛЬНЫХ ОПЕРАЦИЙ

Под вспомогательными понимаются операции, не связанные с непосредственным преобразованием информации. Сюда относятся операции ввода-вывода и пересылочные операции. Операции ввода-вывода выполняются по инструкциям формата SI, поэтому предва-

рительно рассматривается взаимодействие устройств при выполнении логических операций, задаваемых инструкциями этого формата.

Реализация инструкций формата SI. Характерной особенностью инструкций формата SI является использование принципа непосредственной адресации для второго операнда. Первый операнд всегда выбирается из оперативной памяти; для него в коде инструкции указывается адрес базы, т. е. номер регистра местной памяти, и смещение. Второй операнд размещается всегда непосредственно в коде инструкции и имеет однобайтный формат (разряды 8—15 кода инструкции). При выполнении отдельных операций результат записывается в оперативную память по адресу первого операнда.

Реализация инструкций формата SI производится в последовательности, близкой к последовательности реализации инструкций формата RX. Однако есть и заметные отличия, связанные с формированием адресов, а также выборкой операндов. Принимая обозначения, введенные ранее, получим схему связей и взаимодействия блоков и устройств при реализации инструкций формата SI, приведенную на рис. 8.3. Отметим, что эта схема отвечает максимальному набору микроопераций для этих инструкций, так как предусматривает запись результата в оперативную память.

На первом этапе реализации инструкций формата SI, как и для инструкций уже рассмотренных форматов, производится передача кода инструкции и ее частей из РКР в регистры блока центрального управления и блока местной памяти. В РКМ код инструкции передается полностью, а в адресный регистр PA1 местной памяти — содержимое РКР [16—19], т. е. адрес базы, используемой для формирования адреса первого операнда. После выдачи кода реализуемой инструкции и ее частей в другие блоки регистр распакованной команды, т. е. РКР, считается свободным, и на уровне *K* возможна обработка очередной инструкции. Действия первого этапа описываются соотношениями:

$$\text{РКМ} := \text{РКР},$$

$$\text{РКР} [0 — 7] \rightarrow \text{ДШКО},$$

$$\text{РА1} := \text{РКР} [16 — 19].$$

На втором этапе реализации инструкции выполняются действия, относящиеся к формированию адреса первого операнда. В отличие от случая инструкции формата RX исполнительный адрес для оперативной памяти формируется только по кодам базы и смещения. Значение базы выбирается из местной памяти, а значение смещения — из соответствующих разрядов РКМ. Суммирование осуществляется в блоке сумматора адреса, показанном на рис. 8.3 в упрощенном виде. Результат суммирования, т. е. исполнительный ад-

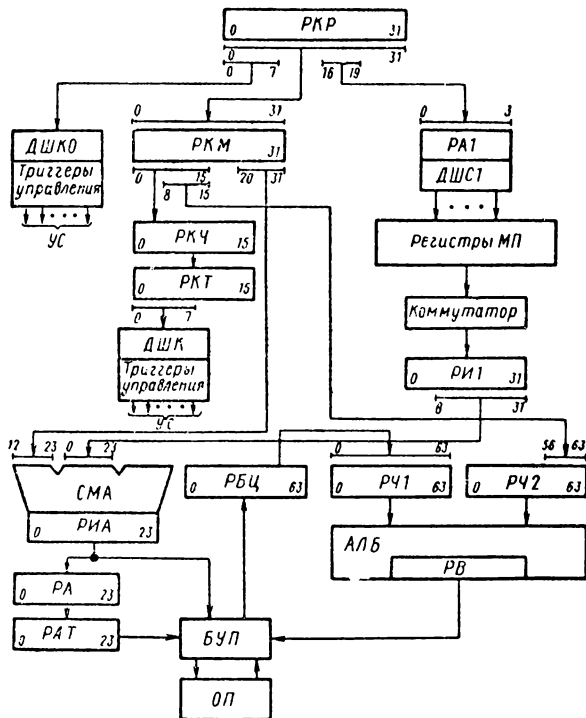


Рис. 8.3. Связи блоков и устройств при выполнении инструкций формата SI

рес первого операнда устанавливается в выходном регистре блока, т. е. в РИА. Действия второго этапа можно описать так:

$$РИ1 := ((РА1)),$$

$$РИА := \{РИ1 [8 - 31] + РКМ [20 - 31]\}.$$

Достаточно большой набор микроопераций выполняется на третьем этапе, когда осуществляются выборка операндов, подготовка их для непосредственного выполнения заданной операции в АЛБ, а также некоторые вспомогательные действия по передаче кодов из РКМ в РКЧ и из РИА в РА. Выполняемые действия отражаются следующими обобщенными соотношениями:

$$РИА \rightarrow БУП,$$

$$РКЧ := РКМ [0 - 15], \quad РА := РИА,$$

$$РЧ2 [56 - 63] := РКМ [8 - 15],$$

$$РБЦ := ((РИА)),$$

$$РЧ1 := РБЦ.$$

Выборка первого операнда обеспечивается действиями, начинаемыми пересылкой его исполнительного адреса из РИА в блок управления памятью (БУП). Поскольку выборка из ОП первого операнда длится сравнительно долго, то параллельно ей выполняются другие действия данного этапа. Поскольку адрес первого операнда должен быть сохранен для записи результата выполнения операции, то он передается для промежуточного хранения в РА блока центрального управления. Старшее полуслово кода инструкции используется на заключительном этапе выполнения операции. Поэтому осуществляется передача содержимого РКМ [0—15] в РКЧ, т. е. выходной регистр команд блока центрального управления. Второй операнд, как часть кода инструкции, вводится в младшие разряды РЧ2. После завершения действий в ОП и блоке управления памятью первый операнд принимается в РБЦ, а далее фиксируется в РЧ1; этим заканчиваются действия третьего этапа.

На четвертом этапе производится выполнение действий над операндами в АЛБ и запись результата в оперативную память по исполнительному адресу первого операнда. Для этого содержимое РЧ1 и РЧ2 передается в АЛБ, содержимое РКЧ принимается в РКТ, а содержимое РА — в РАТ блока адреса результата. Действия, выполняемые на рассматриваемом этапе, описываются так:

$$РЧ1 \rightarrow АЛБ, РЧ2 \rightarrow АЛБ,$$

$$РКТ : = РКЧ, РАТ : = РА,$$

$$РКТ [0 - 7] \rightarrow ДШК,$$

$$РОП \rightarrow РВ,$$

$$РАТ \rightarrow БУП, РВ \rightarrow БУП,$$

$$((РАТ)) : = РВ,$$

где РОП — результат выполняемой операции.

После освобождения блоков уровня Ч они могут использоваться для обработки кода очередной инструкции.

Реализация инструкций ввода-вывода. Все инструкции ввода-вывода с кодами операций 9С, 9D, 9Е, 9F являются инструкциями формата SI. Их реализация в блоках процессора сводится к формированию адреса канала и адреса внешнего устройства, с которым должен работать канал, и к передаче в выбранный канал кода операции и адреса устройства; в ЕС-1050 и ЕС-1060 адрес канала не выдается в каналы для опознавания, так как для выборки каналов используются индивидуальные каналные радиальные линии; в этих моделях вместо двухразрядного кода операции в каналы выдается сигнал по линии, отвечающей заданной операции. В коде инструкции непосредственный операнд не используется; формирование адресов канала и устройства осуществляется по базе и смещению.

Инструкции ввода-вывода, в отличие от других инструкций формата SI, выполняются по сокращенной микропрограмме. Это связано с тем, что арифметическо-логический блок не участвует в выполнении операций ввода-вывода, а оперативная память включается в процесс обмена информацией только после того, как в канале начнется выполнение команды или последовательности команд, заданных инструкцией ввода-вывода. Реализация инструкций ввода-вывода в процессоре производится поэтапно.

Первый этап действий полностью совпадает с первым этапом по выполнению логических операций, заданных инструкциями формата SI. Выполняются подготовительные микрооперации; освобождается уровень K. Поэтому начальная часть схемы связей блоков и устройств при реализации инструкций ввода-вывода (рис. 8.4) полностью повторяет начальную часть схемы, показанной на рис. 8.3. Действия второго этапа также не имеют отличий, но в результате сложения базы и смещения в РИА устанавливается не адрес операнда, а код, из которого на последующем этапе выделяются адреса канала и внешнего устройства.

Заключительный этап реализации инструкции ввода-вывода в процессоре начинается с освобождения блоков уровня Ч. Старшее полуслово кода инструкции передается из РКМ через РКЧ в РКТ, а содержимое РИА — через РА в РАТ. Схема, приведенная на рис. 8.4, построена применительно к центральному устройству управления ЕС-1050, поэтому расшифровка адреса канала осуществляется непосредственно в ЦУУ.

Дешифратор кодов (ДШК), связанный со старшими разрядами РКТ, по коду операции, содержащемуся в инструкции ввода-вывода, формирует управляющий сигнал и направляет его в соответствующую линию, идущую к каналам. Этот сигнал «настраивает» выбранный канал на выполнение заданной операции ввода-вывода.

Содержимое РАТ [13—15] определяет адрес канала. Этот адрес дешифрируется с помощью ДШНК — дешифратора номера канала. Возбуждается соответствующий триггер регистра номера канала

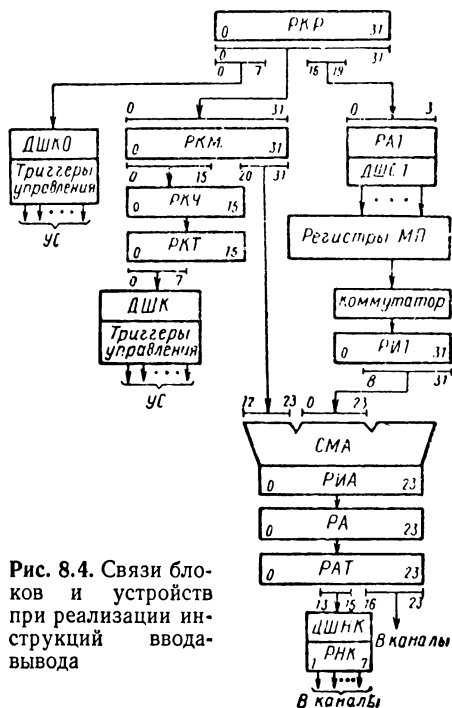


Рис. 8.4. Связи блоков и устройств при реализации инструкций ввода-вывода

(РНК), и в канал по радиальной линии посылается сигнал выборки, побуждающий его выполнить необходимые действия (эти действия рассматриваются в гл. 13). Содержимое РАТ [16—23] дает адрес внешнего устройства, с которым будет работать канал по выполнению заданной операции. Этот адрес выдается в выбранный канал по соответствующим шинам. Реализация инструкции ввода-вывода в процессоре заканчивается приемом из канала кода его состояния.

Реализация инструкций записи в оперативную память. К инструкциям записи в оперативную память относятся инструкции формата RX с кодами операций 40, 50, 60 и 70. По этим инструкциям код, выбираемый из местной памяти по первому адресу, посылается в ОП для записи по второму исполнительному адресу, образуемому при сложении базы, индекса и смещения. При выполнении действий, необходимых для реализации рассматриваемых инструкций, между основными блоками и устройствами машины устанавливаются связи, аналогичные тем, которые определяются схемой, показанной на рис. 8.2.

Формирование исполнительного адреса и выборка первого операнда производятся так же, как и в случае выполнения арифметических и логических операций, задаваемых инструкциями формата RX . Но запрос на обслуживание оперативной памятью посылается в БУП не для выборки, а для записи кода. Поэтому вместе с адресом, находящимся в РИА, в блок управления памятью посылается из РЧИ через арифметическо-логический блок код, выбранный по первому адресу. Записью кода в ОП завершается выполнение любой из рассматриваемых инструкций.

РАСПРЕДЕЛЕНИЕ И ЗАЩИТА ПАМЯТИ

9.1. ПРИНЦИПЫ РАСПРЕДЕЛЕНИЯ ПАМЯТИ

Для ЭВМ Единой системы предусматривается использование принципов динамического распределения памяти, при котором выделение ячеек памяти для размещения информационных массивов производится в процессе решения задач с учетом фактических длин массивов, а также порядка и времени их активизации; под активизацией понимается переход к непосредственной реализации данного информационного массива. Распределение памяти в процессе решения задач осуществляется управляющей программой, учитывающей характеристики рабочих программ по длинам их участков и порядку активизации участков и занятость массивов ячеек запоминающих устройств.

При динамическом распределении памяти заранее неизвестно, какие ячейки ЗУ будут выделены для размещения информационных массивов. Поэтому программирование может осуществляться только в условных, или логических, адресах. Переход к действительным адресам в процессе реализации программ обеспечивается специальными схемами, выполняемыми в соответствии с принятыми принципами организации и распределения памяти. Итак, использование принципов динамического распределения памяти возможно тогда, когда соответствующая модель ЭВМ Единой системы оснащена необходимыми аппаратно-программными средствами.

Динамическое распределение базируется на страничной организации памяти, при которой в оперативных и внешних запоминающих устройствах выделяются участки или зоны одинаковой емкости, называемые физическими страницами. Размеры страниц выражаются количеством ячеек памяти, т. е. емкостью, которая обычно равна целой степени числа 2.

В случае страничной организации вся память машины как бы сводится к одному уровню, называемому математической, или виртуальной, памятью. Можно сказать, что математическая память — это совокупность всех ячеек оперативной и внешней памяти, имеющих сквозную нумерацию. Математическая память, как и физическая, разделяется на страницы, причем размеры страниц математической памяти, или математических страниц, всегда равны размерам страниц физической памяти, или физических страниц.

Поскольку размеры математических и физических страниц одинаковы, то адреса ячеек в пределах одной математической страницы полностью совпадают с адресами ячеек соответствующей страницы, или участка, оперативной памяти. Это облегчает преобразование математических адресов в физические, т. е. в номера действительных ячеек оперативной памяти, при реализации рабочих программ, так как преобразованию подлежат только старшие разряды адреса, определяющие номер страницы.

Соответствие между номерами физических и математических страниц устанавливается управляющей программой при распределении памяти между рабочими программами. Формируется так называемая страничная таблица, или таблица страниц [12], входом в которую является номер математической страницы; обращение к таблице страниц дает номер физической страницы. При каждом перераспределении памяти между рабочими программами производятся необходимые изменения данных, содержащихся в таблице страниц.

Страничная таблица, или таблица страниц, содержит также сведения о распределении физических страниц между оперативной и внешней памятью. Если страничная таблица указывает, что необходимая информация размещена во внешней памяти, то до непосредственного обращения, т. е. до выборки из памяти соответствующего слова, осуществляется обмен страницами между внешней и оперативной памятью.

Преобразование математического адреса в физический сводится в случае страничной организации памяти к преобразованию страниц, т. е. к замене старших разрядов адреса, составляющих номер страницы, в соответствии с данными таблицы страниц. Если эту таблицу разместить в общей оперативной памяти, то при выборке любого слова из ОП необходимо производить по крайней мере два обращения к ней: а) для получения номера физической страницы по заданному в исходном адресе номеру математической страницы, б) для собственно выборки слова по полученному физическому адресу. Для уменьшения общего цикла выборки слова из ОП страничная таблица размещается в специальном устройстве типа сверхоперативной регистровой памяти малой емкости. С целью сокращения общего аппаратного состава в сверхоперативной регистровой памяти обычно хранится только часть общей таблицы страниц, отвечающая наиболее часто используемым на данном временном интервале участкам оперативной памяти.

Кроме страничной организации памяти в машинах Единой системы предусматривается сегментация математической памяти, заключающаяся в разделении всех математических страниц на группы-сегменты с независимой адресацией слов внутри каждой группы. В границах каждого сегмента математические страницы могут относиться как к оперативной, так и к внешней памяти. Теперь адрес любого слова состоит из трех частей: номера сегмента, номера страницы и адреса слова внутри страницы. Для преобра-

зования математического адреса в физический необходимо еще иметь сегментную таблицу, определяющую соответствие между номерами сегментов и таблицами страниц.

Рассмотрим принципы распределения памяти при ее страничной организации применительно к ЕС-1030. Полный 24-разрядный адрес используется для адресации ячеек математической памяти: 4 разряда определяют номер сегмента, 8 разрядов — номер страницы и 12 разрядов — номер ячейки внутри страницы. Таким образом совокупная математическая память разбита на 16 сегментов;

каждый сегмент делится на 256 страниц по 4096 байт в каждой. При емкости оперативной памяти, равной 512К байт, выделяется 128 физических страниц, располагаемых в ОП; очевидно, что номер физической страницы ОП является семиразрядным; разрядность частей адреса учитывается при построении соответствующих схем.

При решении некоторой задачи используется таблица сегментов и для каждого сегмента — таблица страниц. В процессе решения задачи эти таблицы используются для преобразования математического адреса в физический. Если оказывается, что нужная страница находится во внешней памяти, то решение данной задачи прерывается, управляющая программа организует вызов необходимой страницы в ОП. После записи нужной для прерванной программы страницы в оперативную память соответствующая задача ставится в очередь на обслуживание процессором.

Таблица сегментов и большие части таблиц страниц всегда размещаются в оперативной памяти. Это связано с ограниченной емкостью сверхбыстродействующей регистровой памяти, используемой при преобразовании математических адресов в физические. Для хранения строк страничных таблиц, отвечающих наиболее часто используемым физическим страницам, используется регистровая ассоциативная память (АП), включающая в свой состав 8 основных регистров и 8 схем сравнения; регистры 21-разрядные, а схемы сравнения — 12-разрядные.

Общая схема преобразования адресов приведена на рис. 9.1. Ассоциативная память (АП) представлена в виде одного блока, а для главных регистров схемы указаны лишь основные связи. Для рассматриваемой схемы приняты следующие обозначения: РМА — регистр математического адреса, КА — код адреса, РФА — регистр физического адреса, РТ — табличный регистр

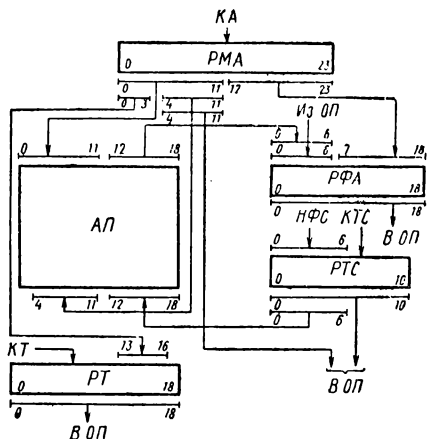


Рис. 9.1. Схема преобразования адресов

сегментов, РТС — табличный регистр страниц, КТ — код таблицы сегментов, КТС — код таблицы страниц, НФС — номер физической страницы.

Каждый основной регистр АП хранит одну строку страничной таблицы; разряды [0—11] содержат номер сегмента и номер математической страницы, разряды [12—18] — номер физической страницы. Разряд [19] каждого регистра ассоциативной памяти служит указателем использования соответствующей строки страничной таблицы; при обращении к данной строке, т. е. к некоторой физической странице, в разряд [19] записывается единица. Разряд [20] является указателем вида физической памяти; если в этом разряде записан нуль, то соответствующая страница находится во внешней памяти.

Работа рассматриваемой схемы по преобразованию адресов начинается с ввода в РМА кода адреса. Этот код необходимо преобразовать и получить по нему код физического адреса. Номер ячейки внутри страницы преобразованию не подлежит, поэтому содержимое РМА в разрядах [12—23]* непосредственно передается в РФА [7—18]. Номер же физической страницы в РФА [0—6] может быть введен только после выполнения некоторых дополнительных действий.

Наиболее просто номер физической страницы определяется тогда, когда она размещается в оперативной памяти, а отвечающая ей строка таблицы страниц находится в одном из регистров ассоциативной памяти. В этом случае дополнительные обращения к оперативной памяти не производятся; выборка операнда производится за минимальное время. Во всех остальных случаях необходимы дополнительные обращения к ОП, что увеличивает время, затрачиваемое на выборку операнда.

Действия по определению номера физической страницы всегда начинаются со сравнения содержимого РМА [0—11] с содержимым старших 12 разрядов всех основных регистров АП; сравнение производится параллельно с помощью восьми схем сравнения. Если содержимое 20-го разряда некоторого регистра АП равно нулю, то сравнение по этому регистру не производится (соответствующая страница размещена во внешней памяти). Совпадение кодов в одной из сравниваемых пар свидетельствует о том, что отвечающая математической странице физическая страница находится в ОП; номер этой физической страницы размещен в разрядах [12—18] соответствующего регистра АП. Выявленный номер физической страницы передается в РФА [0—6], а в 19-й разряд регистра АП заносится единица, как признак «активности» данной страницы. Полностью сформированный физический адрес из РФА передается в оперативную память для выборки операнда.

Если содержимое РМА [0—11] не совпадает с содержимым старших разрядов ни одного из регистров АП, то это указывает

* Далее для краткости слова «в разрядах» опускаются.

на необходимость поиска нужной строки таблицы страниц и ее размещения в АП. Для этого используются таблица сегментов и таблица страниц, размещенные в оперативной памяти. Сначала по таблице сегментов необходимо определить адрес начальной строки нужной таблицы страниц, а уж затем выявлять требуемую строку этой таблицы.

При активизации данной задачи, т. е. с началом реализации программ ее решения, управляющая программа вводит в табличный регистр РТ код таблицы сегментов (КТ), определяющий адрес первой строки этой таблицы. Чтобы получить адрес нужной строки таблицы сегментов, в разряды [13—16] табличного регистра передается номер сегмента как часть текущего математического адреса, т. е. производится действие

$$РТ[13—16]: = РМА[0—3].$$

Сформированный адрес из РТ направляется в оперативную память для выборки соответствующей строки сегментной таблицы, представляющей собой полное слово. Младшие разряды этого слова представляют собой код таблицы страниц (КТС), определяющий адрес первой строки соответствующей страничной таблицы. Этот код вводится в РТС. Для получения адреса искомой строки таблицы страниц к содержимому РТС приформировывается номер страницы как часть текущего математического адреса, т. е. содержимое РМА [4—11]. Сформированный таким образом адрес направляется в оперативную память для выборки требуемой строки таблицы страниц.

Выбираемое при данном обращении к ОП слово содержит номер искомой физической страницы только в том случае, когда эта страница размещена в оперативной памяти. О месте размещения страницы указывает значение 12-го (28-го) разряда. Если искомая страница находится в оперативной памяти, то ее номер вводится в РТС [0—6] и в РФА [0—6]. Полученный в РФА полный физический адрес посылается в ОП для выборки операнда. Итак, в рассматриваемом случае выборка операнда из ОП производится только при третьем обращении к оперативной памяти.

Размещение номера физической страницы НФС в РТС необходимо для обеспечения действий по записи новой строки таблицы страниц в ассоциативную память. Для записи этой строки выбирается первый в порядке номеров регистр АП, содержащий нуль в 19-м разряде. Считается, что физическая страница, определяемая строкой таблицы, находящейся в этом регистре, используется реже других (отметим, что при равенстве единице содержимого разряда [19] всех регистров АП осуществляется сброс этого разряда регистров в нулевое состояние). Собственно запись новой строки таблицы страниц в АП производится за счет передачи в разряды [12—18] выбранного регистра содержимого РТС [0—6], а в разряды [4—11] этого регистра — содержимого РМА [4—11]; если необходимо, то в разряды [0—3] вводится код номера сегмента.

В том случае, когда выбранная при втором обращении к оперативной памяти строка страничной таблицы свидетельствует о том, что нужная страница находится во внешней памяти, управляющая программа производит перераспределение за счет обмена страницами между ОП и внешней памятью. При этом корректируются таблицы страници, что требует дополнительных затрат времени. Реализация текущей программы прерывается и возобновляется только после выполнения всех действий по перераспределению памяти и обслуживанию первоочередных задач.

9.2. БЛОКИ ЗАЩИТЫ

Защита оперативной памяти осуществляется во всех моделях ЕС ЭВМ по ключам. Сущность такого метода защиты заключается в том, что каждой рабочей программе и отводимым для ее размещения массивам ячеек (страницам) основной оперативной памяти придается специальный код — ключ защиты. Обращение к памяти разрешается только в том случае, когда ключ программы совпадает с ключом защиты памяти, т. е. с ключом, приданным массиву ячеек (странице), к которому осуществляется обращение по записи или считыванию. Условно считается, что ключ 0 . . . 00 сравним со всеми остальными ключами. При несовпадении ключей должен формироваться сигнал о нарушении защиты памяти.

Принципы построения блоков защиты для всех моделей ЭВМ Единой системы одинаковы, хотя аппаратура систем защиты иногда рассматривается как часть блоков памяти ключей защиты. Общность принципов защиты памяти и их реализации позволяет выявить сущность построения соответствующих схем на примере блока защиты одной из моделей, хотя бы ЕС-1020. В модели ЕС-1020 блок защиты обеспечивает защиту памяти от недопустимых обращений не только со стороны рабочих или обслуживающих программ, но и со стороны селекторных каналов. При этом защита от недопустимых передач информации из селекторных каналов в ОП осуществляется также по ключам; каждому селекторному каналу всегда придается так называемый ключ канала.

Структурная схема блока защиты приведена на рис. 9.2, где показаны все его основные части. Не показаны только входные и выходные фильтры нижних частот, которые обеспечивают защиту

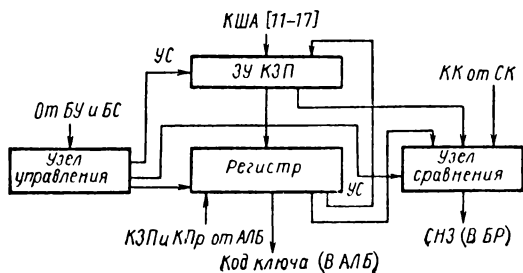


Рис. 9.2. Структурная схема блока защиты

кодовых и управляющих сигналов от высокочастотных помех; эти фильтры выполнены на соответствующих ТЭЗах печатным способом в виде распределенных индуктивностей. В состав блока защиты (БЗ) входят: узел управления, регистр кодов ключей, узел сравнения и запоминающее устройство ключей защиты памяти (ЗУ КЗП). От блока управления (БУ) и блока синхронизации (БС) центрального устройства управления рассматриваемый блок получает начальные управляющие сигналы, а от арифметическо-логического блока (АЛБ) и селекторных каналов (СК) — коды ключей программ (КПр) и ключей каналов (КК) соответственно.

Коды ключей из регистра блока защиты могут выдаваться в АЛБ; сигнал нарушения защиты (СНЗ) выдается в блок регистров (БР) процессора машины. Выборка ключей защиты памяти из ЗУ КЗП производится по семи старшим разрядам адреса ячейки оперативной памяти, поступающим из кодовых шин адреса КША [11—17]. Максимальная емкость ЗУ КЗП 128 шестиразрядных кодов. Время задержки формирования сигнала нарушения защиты относительно момента поступления кода адреса — 300 нс.

Регистр блока защиты, или РБЗ, служит для приема, хранения и выдачи в другие узлы и блоки кодов ключей, используемых в процессе защиты памяти. Он состоит из собственно 10-разрядного триггерного регистра с входными и выходными логическими схемами, а также стандартной схемы свертки для формирования контрольного разряда кода, передаваемого в арифметическо-логический блок (на вход А этого блока). В первые шесть триггеров регистра информация может вводиться как с выходов С арифметическо-логического блока, так и с выходов регистра информации ЗУ КЗП; это триггеры разрядов К, 0, 1, 2, 3, 4. В остальные четыре триггера регистра разрядов 4', 5, 6 и 7 информация может вводиться только с выходов С арифметическо-логического блока. Связи РБЗ с АЛБ полностью отвечают схеме рис. 7.7.

Разряды К, 0, 1, 2, 3, 4 представляют ключ защиты памяти. При этом собственно КЗП представляют разряды 0—3, разряд К является контрольным, а единица в разряде 4 свидетельствует о том, что защита памяти осуществляется в режиме считывания информации из ОП. Разряды 4'—7 представляют ключ программы, реализуемой процессором.

Внутри блока защиты информация из регистра передается в ЗУ КЗП и узел сравнения. При этом в ЗУ КЗП передается содержимое первых шести разрядов, т. е. разрядов К, 0, 1, 2, 3, 4, при смене ключей защиты памяти. Управление записью в ЗУ КЗП осуществляется сигналами, снимаемыми с единичных выходов соответствующих триггеров. В узел сравнения выдаются сигналы как с единичных (прямых), так и нулевых (инверсных) выходов триггеров разрядов 4', 5, 6 и 7; иначе говоря, в узел сравнения выдается как прямой, так и обратный код ключа программы.

Все триггеры рассматриваемого регистра имеют парафазные входы, поэтому нет специальной цепи сброса их в исходное нуле-

вое состояние. Коду единицы на входах и выходах триггеров отвечает сигнал высокого уровня.

Узел сравнения служит для формирования сигнала нарушения защиты по результатам анализа ключей защиты памяти и ключей программ или ключей селекторных каналов. Код ключа программы подается в узел сравнения из РБЗ, код ключа защиты памяти — из регистра информации ЗУ КЗП, код ключа селекторного канала — из регистра ключа канала (РКК). Сравнение ключа защиты памяти с ключом канала производится при действии управляющего импульса $УИ_k$ высокого уровня; в противном случае ключ защиты памяти сравнивается с ключом программы.

Сигнал нарушения защиты вырабатывается в виде сигнала высокого уровня, если не выполняется хотя бы одно из четырех равенств для двух случаев сравнения ключей:

$$\begin{aligned} 1) \text{ РКК } [0] &= \text{РИ } [0], & 2) \text{ РБЗ } [4'] &= \text{РИ } [0], \\ \text{РКК } [1] &= \text{РИ } [1], & \text{РБЗ } [5] &= \text{РИ } [1], \\ \text{РКК } [2] &= \text{РИ } [2], & \text{РБЗ } [6] &= \text{РИ } [2], \\ \text{РКК } [3] &= \text{РИ } [3], & \text{РБЗ } [7] &= \text{РИ } [3], \end{aligned}$$

где РКК $[i]$ — значение i -го разряда кода ключа, содержащегося в регистре ключа канала ($i = 0, 1, 2, 3$);

РИ $[i]$ — значение i -го разряда кода ключа защиты памяти, содержащегося в регистре информации ЗУ КЗП ($i = 0, 1, 2, 3$).

В соответствии с принципами формирования сигнала нарушения защиты можно записать следующее логическое соотношение:

$$\text{СНЗ} = (x_0 \oplus y_0) \vee (x_1 \oplus y_1) \vee (x_2 \oplus y_2) \vee (x_3 \oplus y_3), \quad (9.1)$$

где $y_i = \text{РИ } [i], i = 0, 1, 2, 3,$

$$x_i = x_i^1 \vee УИ_k \vee x_i^2 \overline{УИ_k}, \quad (9.2)$$

$$x_i^1 = \text{РКК } [i], i = 0, 1, 2, 3,$$

$$x_i^2 = \text{РБЗ } [i], i = 4', 5, 6, 7.$$

Так как в системе элементов серии 155 имеются только элементы И—НЕ и И—ИЛИ—НЕ, то соотношения (9.1) и (9.2) необходимо преобразовать, используя выражение для представления суммы по модулю два через элементарные логические операции, а также прием двойного инверсирования. Тогда

$$\text{СНЗ} = \overline{z_0 z_1 z_2 z_3}, \quad (9.3)$$

где

$$z_i = \overline{x_i y_i} \vee \overline{x_i y_i}, i = 0, 1, 2, 3,$$

Очевидно, что для реализации соотношений (9.3) необходимо использовать 4 элемента *И—ИЛИ—НЕ* с двумя двухвходовыми элементами *И* и один четырехвходовой элемент *И—НЕ*.

Получение прямых и инверсных значений величин x_i и y_i обеспечивается при реализации преобразованного соотношения (9.2):

$$x_i = \overline{x_i^1 \text{ УИ}_k \vee x_i^2 \overline{\text{УИ}_k}}. \quad (9.4)$$

Инверсируя обе части (9.2), получим

$$\overline{x_i} = x_i^1 \text{ УИ}_k \vee x_i^2 \overline{\text{УИ}_k}. \quad (9.5)$$

Поскольку в реальной схеме необходимо реализовать 4 соотношения (9.4) и 4 соотношения (9.5), то используются 8 элементов *И—ИЛИ—НЕ* с двумя двухвходовыми элементами *И* каждый.

Сигналы РКК [i] поступают в блок защиты не парафазно, поэтому дополнительно используются 4 инвертора типа одноходовых элементов *И—НЕ*; два инвертора необходимы для инверсирования сигнала УИ_k . Всего для построения схемы сравнения ключей необходимо использовать 12 элементов *И—ИЛИ—НЕ*, один элемент *И—НЕ* и 6 инверторов.

Схема сравнения ключей (рис. 9.3.) полностью отвечает приведенным выше соотношениям, обеспечивая на своем выходе формирование сигнала нарушения защиты СНЗ в виде сигнала высокого уровня. Если нарушения защиты памяти нет, то на выходе схемы сравнения имеет место сигнал низкого уровня; ТРИ [i] ($i = 0, 1, 2, 3$) эквивалентно РИ [i]; это же замечание относится к рис. 9.4 и 9.9.

Рассмотренная схема сравнения формирует сигнал нарушения защиты при несовпадении ключа защиты памяти с ключом программы или с ключом канала. В то же время при (РКК) = 0, т. е. при РКК [0] = 0, РКК [1] = 0, РКК [2] = 0, РКК [3] = 0, или при РБЗ [4'] = 0, РБЗ [5] = 0, РБЗ [6] = 0, РБЗ [7] = 0 сигнал нарушения защиты не должен учитываться, так как принято, что нулевые коды ключей сравнимы со всеми остальными кодами. Для выявления случаев равенства нулю кода ключа программы или кода ключа канала используется специальная схема, на выходе которой формируется сигнал блокировки. При наличии этого сигнала сигнал нарушения защиты игнорируется.

Узел управления служит для формирования всех управляющих сигналов, непосредственно используемых в блоке защиты. Исходными для него являются сигналы, поступающие из блока управления и блока синхронизации. Основу узла составляют схемы, реализующие конъюнкции и дизъюнкции исходных сигналов, а также инверторы, служащие не только для изменения уровня исходных сигналов, но и для развязки выходных цепей узла с управляемыми цепями блока защиты.

Работа блока защиты. Рассмотренный блок защиты может работать в трех режимах:

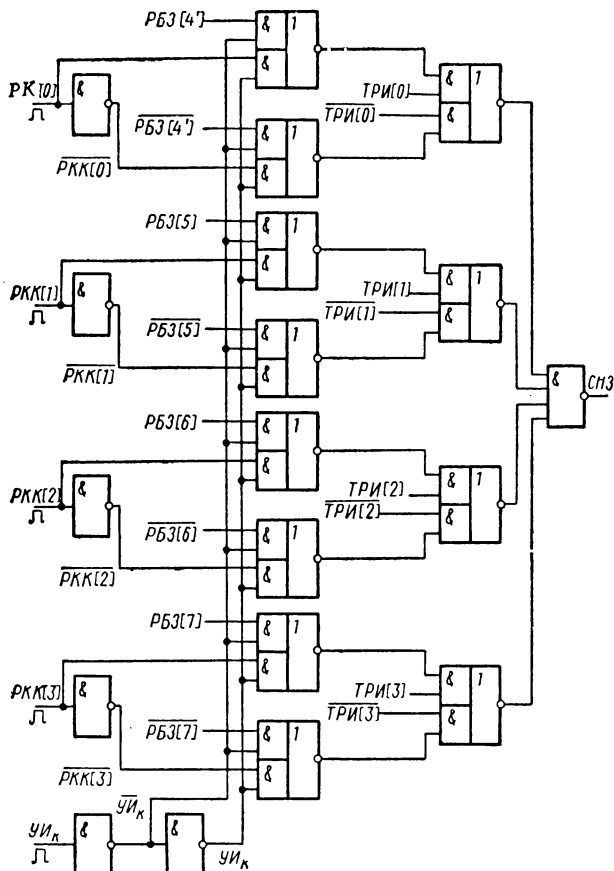


Рис. 9.3. Схема сравнения ключей

— автоматическом, который включается при каждом обращении к основной оперативной памяти;
 — режиме считывания информации из ЗУ КЗП;
 — режиме записи информации в ЗУ ЗКП;
 при работе блока защиты в его ЗУ КЗП постоянно передаются семь старших разрядов адреса, засылаемого в регистр адреса основной оперативной памяти.

Работа блока защиты в автоматическом режиме протекает при следующей последовательности действий:

- 1) подготовка ЗУ КЗП к считыванию информации;
- 2) считывание слова (кода ключа) из ЗУ КЗП;
- 3) регенерация считанной информации;
- 4) передача значения четвертого разряда считанного из ЗУ КЗП слова для анализа в блок регистров процессора;

5) контроль по нечетности считанного слова в узле контроля ЗУ КЗП;

6) сравнение кода ключа защиты памяти, считанного из накопителя ЗУ КЗП, с ключом программы, предварительно засланным в РБЗ (или с ключом канала);

7) при совпадении ключей на выходной шине узла сравнения (шины СНЗ) сохраняется низкий уровень сигнала; при равенстве нулю ключа программы или ключа канала формируется сигнал блокировки;

8) ключ защиты памяти из ЗУ КЗП передается в разряды [К, 0—4] регистра блока защиты;

9) на входные шины А арифметическо-логического блока выводится информация с нулевых выходов триггеров РБЗ;

10) схемы узла управления и узла сравнения блока защиты возвращаются в исходное состояние.

Работа блока защиты в режиме считывания информации из ЗУ КЗП отличается от работы в автоматическом режиме только тем, что независимо от значения кода ключа формируется сигнал блокировки, по которому сигнал нарушения защиты игнорируется.

Работа блока защиты в режиме записи информации в ЗУ КЗП отличается от работы в режиме считывания тем, что считанное из ЗУ КЗП слово в РБЗ не передается, в узле сравнения блокируется выдача сигнала нарушения защиты, а в ЗУ КЗП записывается содержимое разрядов [К, 0—4] регистра РБЗ, полученное в одном из предыдущих циклов работы машины из арифметическо-логического блока процессора.

9.3. ПАМЯТЬ КЛЮЧЕЙ ЗАЩИТЫ

Включение в состав процессоров блоков защиты, работающих в каждом цикле обращения к основной оперативной памяти, не должно сказываться на быстродействии, т. е. не должно увеличивать реальный цикл работы ОП. Очевидно, что это может быть обеспечено только при использовании для хранения ключей защиты ЗУ с малым циклом обращения; время обращения к таким ЗУ должно быть в несколько раз меньше, чем время обращения к основной оперативной памяти. В связи с этим в качестве памяти ключей защиты используются запоминающие устройства на туннельных диодах (ЕС-1020) и на тонких магнитных пленках (ЕС-1030, ЕС-1050).

ЗУ ключей защиты с накопителем на туннельных диодах. Запоминающее устройство ключей защиты памяти блока защиты ЕС-1020 служит для приема из РБЗ, хранения и выдачи кодов ключей в узел сравнения. Оно имеет емкость 128 шестиразрядных слов и построено в соответствии с общими принципами организации структур ЗУ с произвольным доступом к хранимой информации. Накопитель рассматриваемого ЗУ составлен из запоминающих элементов, выполненных в виде триггеров на туннельных диодах. Эта особенность учитывается составом узлов устройства, общая

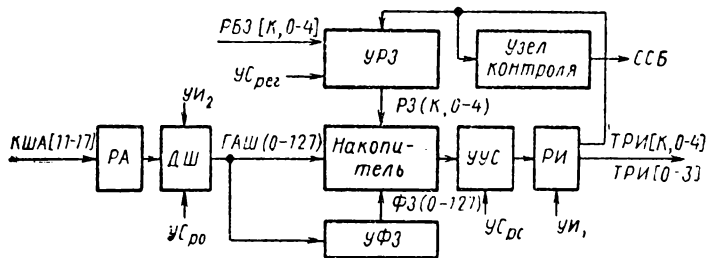


Рис. 9.4. Схема запоминающего устройства ключей защиты с накопителем на туннельных диодах

схема которого приведена на рис. 9.4, где использованы следующие обозначения: РА — регистр адреса, ДШ — дешифратор адреса, $УС_{po}$ — управляющий сигнал разрешения обращения к ЗУ, ГАШ (0—127) — сигналы на выходах дешифратора адреса, УРЗ — узел разрешения записи, $УС_{рег}$ — управляющий сигнал регенерации информации, РЗ (К, 0—4) — разрядные сигналы разрешения записи, УФС — узел формирователей записи, ФЗ (0—127) — сигналы на выходах формирователей записи, УУС — узел усилителей считывания, $УС_{pc}$ — управляющий сигнал разрешения считывания (СТРОБ), РИ — регистр информации (регистр числа) запоминающего устройства, ССБ — сигнал сбоя при контроле на нечетность.

Накопитель представляет собой матрицу из 768 запоминающих ячеек (ЗЯ), основу которых составляют триггеры на туннельных диодах. Запоминающие ячейки сгруппированы в 128 числовых линеек (ЧЛ), каждая из которых обеспечивает хранение кода одного ключа защиты, представляемого 6-разрядным двоичным словом. Структура накопителя приведена на рис. 9.5.

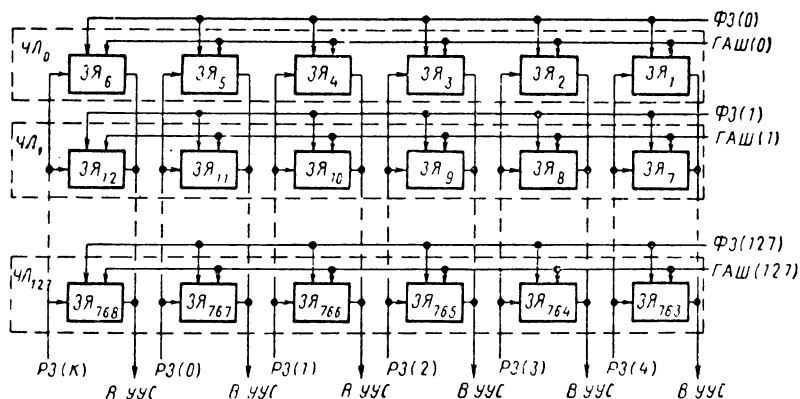


Рис. 9.5. Структура накопителя

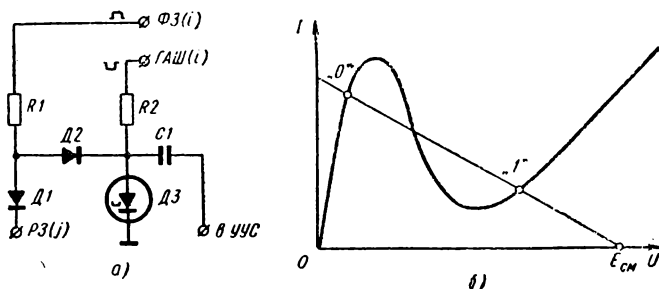


Рис. 9.6. Запоминающая ячейка:

а — принципиальная схема; б — вольт-амперная характеристика туннельного диода

Числовыми для накопителя являются шины, по которым передаются сигналы ГАШ и ФЗ. Сигнал ГАШ (i), где $i = 0, 1, \dots, 127$, представляет собой сигнал считывания из i -й числовой линейки, а сигнал ФЗ (i) — сигнал, по которому могут быть записаны единицы во все ЗЯ i -й числовой линейки. Шины передачи сигналов РЗ и выходные шины являются разрядными; единица в j -й запоминающей ячейке ($j = K, 0, 1, 2, 3, 4$) выбранной числовой линейки записывается только тогда, когда в соответствующую шину послан сигнал РЗ (j). Выходы ЗЯ группируются по восемь; поэтому к каждому разряду УУС из накопителя подходит 16 шин.

Запоминающая ячейка представляет собой триггер, образованный резистором R2 и туннельным диодом ДЗ, к выходу которого подключена схема совпадений, состоящая из диодов Д1, Д2 и резистора R1. Принципиальная схема запоминающей ячейки приведена на рис. 9.6, а; вольт-амперная характеристика туннельного диода и нагрузочная прямая показаны на рис. 9.6, б.

Туннельный диод может находиться в одном из двух устойчивых состояний: низковольтном (состояние «0») и высоковольтном (состояние «1»). При поступлении сигнала ГАШ (i) все туннельные диоды i -й числовой линейки переключаются в состояние «0», так как в этом случае $E_{см} \approx 0$. На анодах туннельных диодов, находившихся в состоянии «1», возникают отрицательные перепады напряжения, поступающие через емкость С1 в УУС (рис. 9.4).

При поступлении сигнала ФЗ (i) во всех запоминающих ячейках i -й числовой линейки могут быть записаны единицы. Однако запись единиц осуществляется только в тех ячейках, на которые подается сигнал РЗ (j) высокого уровня. При одновременном действии сигналов ФЗ (i) и РЗ (j) высокого уровня ток от формирователя записи протекает по цепи R1, Д2, ДЗ; туннельный диод ДЗ переключается в состояние «1» и сохраняет это состояние после окончания действия сигнала ФЗ (i). Если же сигнал РЗ (j) имеет низкий уровень (случай записи нуля), то ток от формирователя записи протекает по цепи R1, Д1; диод Д2 закрыт, и туннельный диод ДЗ остается в состоянии «0».

Регистр адреса РА предназначен для приема адреса при обращении к ЗУ ключей защиты. Он состоит из семи идентичных симметричных триггеров с парафазными входами. Адрес в РА поступает из КША в виде 7-разрядного парафазного кода.

Дешифратор адреса ДШ служит для формирования сигналов ГАШ (i) низкого уровня, отвечающих кодам, вводимым в регистр адреса при обращении к ЗУ ключей защиты. Он состоит из дешифраторов первой ступени ДШХ и ДШУ и дешифратора второй ступени, имеющего в своем составе 128 элементов, называемых элементами оконечной дешифрации (см. рис. 9.7). Дешифратор ДШХ имеет восемь выходов X_0, \dots, X_7 и четыре входа: три входа — кодовые, а четвертый — для управляющего сигнала УИ₂. Составлен дешифратор из восьми элементов И—НЕ на четыре входа каждый. Дешифратор ДШУ имеет 16 выходов Y_0, \dots, Y_{15} и шесть входов: четыре входа — кодовые, на пятый вход подается сигнал УИ₂, а на шестой — управляющий сигнал разрешения обращения УС_{р0}. Составлен дешифратор из 16 элементов И—НЕ на шесть входов каждый.

Элементы оконечной дешифрации представляют собой схемы И—НЕ на два входа, к выходам которых подключены эмиттерные повторители ЭП₀—ЭП₁₂₇ с симметричными ускоряющими транзисторами, обеспечивающими уменьшение длительности фронтов выходных сигналов. Формирование сигналов низкого уровня ГАШ (i) на выходах дешифратора осуществляется в соответствии с кодом адреса только при действии сигналов УИ₂ и УС_{р0}.

Узел усилителей считывания предназначен для усиления сигналов кодов, считываемых из накопителя ключей защиты. Выдача усиленных до стандартной для микросхем серии 155 величины сигналов из УУС в регистр информации производится только при действии управляющего сигнала УС_{р0}. В состав узла входят: двенадцать усилителей считывания (УС), по два на один разряд, шесть элементов разрешения считывания (РС) и эмиттерный повторитель (ЭП) управляющего сигнала УС_{р0}. Общая структура узла приведена на рис. 9.8.

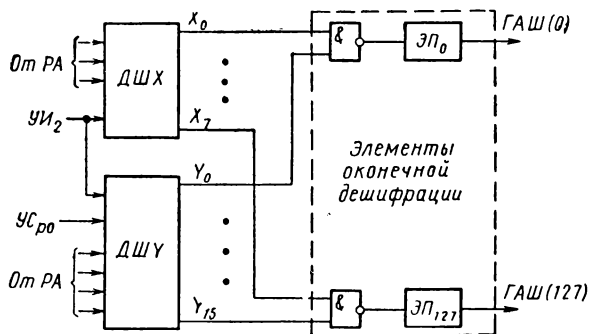


Рис. 9.7. Схема дешифратора адреса

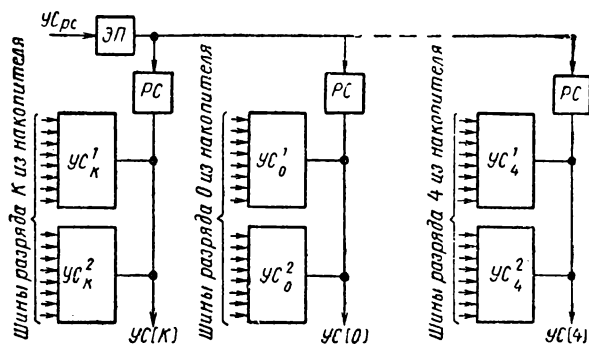


Рис. 9.8. Схема узла усилителей считывания

Регистр информации (РИ) предназначен для приема кодов, считанных из накопителя, управления регенерацией (перезаписью) кодов и выдачи кодов по шинам РИ (j) в другие узлы блока защиты. Он состоит из шести идентичных симметричных триггеров. Запись кодов в РИ производится сигналами УС (j); сброс триггеров регистра в исходное нулевое состояние производится сигналом УИ₁.

Узел разрешения записи предназначен для формирования сигналов РЗ (j), соответствующих коду слова, записываемого в накопитель. Управление работой схем узла производится сигналом УС_{рег}. Если этот сигнал имеет низкий уровень, то РЗ (j) формируется по коду, поступающему из регистра информации ЗУ; это отвечает случаю регенерации считанной информации. Если же УС_{рег} имеет высокий уровень, то РЗ (j) формируется по коду, поступающему из общего регистра блока защиты; это отвечает случаю записи новых кодов ключей в ЗУ КЗП. Узел состоит из инвертора сигналов УС_{рег}, шести элементов И—ИЛИ—НЕ и шести элементов разрешения записи (РЗ), как это показано на рис. 9.9.

Элементы И—ИЛИ—НЕ служат для переключения кодов, поступающих из РБЗ и РИ с выдачей их в проинвертированном виде на элементы РЗ в зависимости от уровня сигнала УС_{рег}. Записываемый в накопитель код еще раз инвертируется в схемах РЗ, так как каждый РЗ состоит из последовательно включенных эмиттерного повторителя и однокаскадного усилителя. Таким образом, если в j -м разряде ($j = K, 0, 1, 2, 3, 4$) записывается 1, то сигнал РЗ имеет высокий уровень. Выходные части схем элементов РЗ обеспечивают ограничение токов, протекающих через туннельные диоды запоминающих ячеек накопителя при переключении их в состояние «1».

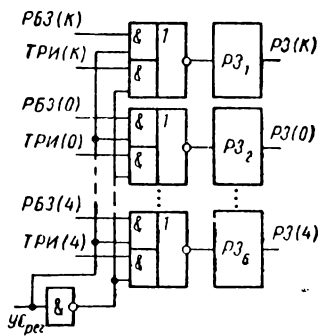


Рис. 9.9. Схема узла разрешения записи

Узел формирователей записи служит для формирования сигналов ФЗ (i) по сигналам ГАШ (i), поступающим с выходов дешифратора адреса. Он состоит из 128 идентичных формирователей, основу которых составляют каскады с общим эмиттером и выходными дросселями, обеспечивающими исключение влияния индуктивности шин РЗ (j) на работу запоминающих ячеек.

Узел контроля предназначен для контроля правильности считывания информации из накопителя и передачи ее в РИ. Практически осуществляется контроль по четности всего 6-разрядного кода, принятого из накопителя в РИ. Если этот код содержит нечетное количество единиц, то в узле контроля формируется сигнал сбоя ССБ. Основу рассматриваемого узла составляет схема свертки, построенная как трехкаскадный сумматор по модулю 2.

Цикл работы рассматриваемого ЗУ, как и любого запоминающего устройства с разрушающим считыванием, состоит из двух основных частей. В первой части цикла всегда осуществляется считывание слова по заданному адресу или стирание информации в соответствующей запоминающей ячейке (подготовка к записи). Во второй части цикла производится регенерация считанной информации или запись нового слова. При записи и считывании используются управляющие сигналы, действие которых указывалось при рассмотрении основных узлов запоминающего устройства ключей защиты с накопителем на туннельных диодах.

ЗУ ключей защиты с накопителем на тонких магнитных пленках. В модели ЕС-1050 защита оперативной памяти осуществляется при выделении страниц емкостью 2048 байт каждая. При полном составе ОП включает в свой состав 512 страниц; в соответствии с этим при полной емкости памяти ключей защиты используются два идентичных тонкопленочных ЗУ емкостью по 256 кодов. Коды ключей, размещаемые в этих ЗУ, являются 9-разрядными; собственно ключ защиты представляют разряды [0—3], 4-й разряд указывает на характер защиты (0 — защита по записи, 1 — защита по считыванию), 5-й разряд используется для размещения признака обращения к данной странице, 6-й — для размещения признака записи в данную страницу, 7-й — резервный, 8-й — контрольный. При работе блока защиты обычно используются только [0—3], 4-й и 8-й разряды.

Запоминающими элементами в рассматриваемой памяти ключей защиты являются планарные магнитные пленки, конструктивно выполняемые в виде матриц с общими размерами 256×9 , что отвечает необходимой емкости ЗУ. Запоминающие элементы группируются по девять, образуя так называемые числовые линейки и обеспечивая построение цепей выборки по системе 2D. С учетом времени, затрачиваемого на регенерацию считанной информации, полный цикл обращения к памяти ключей составляет 0,3 мкс; на собственно считывание затрачивается время порядка 0,22 мкс.

При построении ЗУ ключей защиты с накопителем на тонких магнитных пленках используются общие принципы формирования

схем запоминающих устройств, реализующих выборку кодов хранимых чисел по системе 2D. Поэтому общая схема рассматриваемого устройства, приведенная на рис. 9.10, включает в свой состав узлы, присущие практически всем ЗУ с записью при совпадении полутоков и считыванием полными токами. Для схемы приняты следующие обозначения: КА — код адреса, КК — код ключа, РА — регистр адреса, ДША — дешифратор адреса, ФА — формирователи адресные, РКК — регистр кода ключа, ФР — формирователи разрядные, УС — усилители считывания, В — группа выходных вентилялей, УСК — узел сравнения ключей. Цепь обратной связи по передаче кодов с выходов вентилялей в РКК обеспечивает регенерацию считываемой информации.

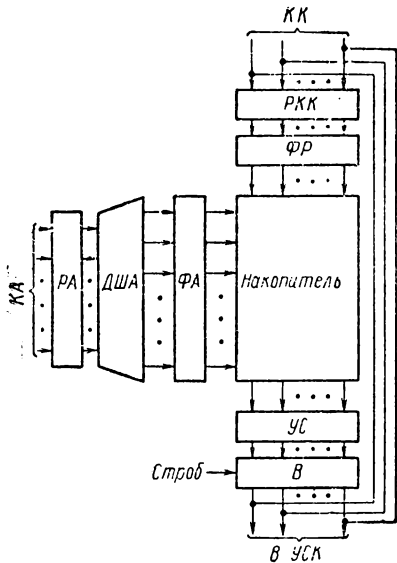


Рис. 9.10. Общая схема запоминающего устройства ключей защиты с накопителем на тонких магнитных пленках

Накопитель представляет собой матрицу с тонкопленочными запоминающими элементами (ЗЭ), поверх которой накладываются в два слоя взаимно перпендикулярные печатные проводники намагничивающей системы; один слой составляют адресные проводники (шины), а второй — разрядно-считывающие проводники. Фиксация проводников намагничивающей системы производится так, что каждому запоминающему элементу отвечает пространственное пересечение адресного провода с одним из разрядно-считывающих, как это показано на схеме накопителя, приведенной на рис. 9.11.

С целью уменьшения количества используемых адресных формирователей и упрощения дешифратора адреса в рассматриваемом ЗУ реализован принцип двухстороннего дешифрирования с разбиением 8-разрядного адреса на две 4-разрядные части. Соответственно используются два дешифратора на 16 выходов каждый и две группы адресных формирователей $ФА'_1, ФА'_2, \dots, ФА'_{16}$ и $ФА''_1, ФА''_2, \dots, ФА''_{16}$. Выходы формирователей посредством диодов попарно объединены по схеме элемента И. Таким образом, при расшифровке любого адреса адресный ток протекает по соответствующей цепи только одной числовой линейки ЧЛ.

Разрядно-считывающая цепь выполнена по мостовой схеме. Это обеспечивает уменьшение разрядной помехи при считывании кодов ключей, так как в этом случае на вход усилителя считывания (УС) поступает разностный сигнал. Отметим, что разрядно-

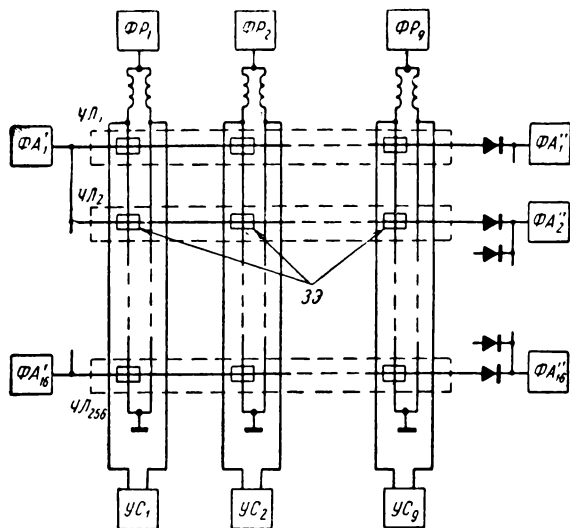


Рис. 9.11. Схема накопителя

считывающие шины ориентируются вдоль осей трудного намагничивания запоминающих элементов.

Общий цикл работы запоминающего устройства состоит из двух подциклов, первый из которых отвечает считыванию или стиранию информации, а второй — регенерации (перезаписи) считанной или записи новой информации. В первом подцикле всегда формируется только адресный ток с амплитудой порядка 0,8 А и длительностью до 40 нс. В разрядно-считывающих цепях, отвечающих элементам, находившимся до считывания в состоянии «1», наводятся сигналы с амплитудой до 1 мВ; эти сигналы поступают на входы усилителей считывания. Если осуществляется обращение по считыванию, то на выходные вентили подается стробирующий сигнал, разрешающий выдачу кода ключа в УСК для сравнения и в РКК для перезаписи. Если же осуществляется обращение по записи, то стробирующий сигнал не формируется, и считанная информация теряется в цепях ЗУ.

Во втором подцикле формируются и адресный ток, и разрядные токи. Адресный ток, как и в первом подцикле, формируется в соответствии с заданным адресом, а разрядные токи формируются по значению кода записываемого ключа. При этом разрядные токи, отвечающие записи 0 и 1, отличаются только своим направлением; амплитуда разрядных токов порядка 0,2 А, а их длительность примерно в 2 раза превышает длительность адресного тока. Те запоминающие элементы, на которые воздействуют суммарные поля адресного и разрядных токов, перемагничиваются в состояние «1»; воздействие разностного поля адресного и

разрядного токов не вызывает изменения исходного нулевого состояния соответствующего запоминающего элемента.

Начальная запись кодов ключей производится при распределении оперативной памяти. Любое перераспределение памяти приводит к записи новых ключей для соответствующих страниц ОП. Изменение ключей, а также их считывание из памяти ключей может производиться по специальным командам УСТАНОВИТЬ КЛЮЧ ПАМЯТИ и ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ.

Память ключей защиты в процессе решения рабочих задач работает при каждом обращении к основной оперативной памяти. По восьми старшим разрядам адреса ОП производится выборка кода ключа и передача его в узел сравнения ключей, где сначала осуществляется проверка принятого кода на нечетность. Далее анализируется значение 4-го разряда кода ключа и производится собственно сравнение ключей. Формирование сигнала нарушения защиты памяти производится по общим правилам, отвечающим принципам защиты памяти по ключам.

ИНТЕРФЕЙС ВВОДА-ВЫВОДА

10.1 ОСНОВНЫЕ ПОЛОЖЕНИЯ И ПОНЯТИЯ

Во всех машинах Единой системы передачи информации, выполняемые по инструкциям (командам) ввода-вывода, организуются с помощью специальных устройств — каналов. Каналы получают от процессоров только начальную управляющую информацию, инициирующую реализацию операций ввода-вывода. В процессе же выполнения этих операций аппаратно-программные средства процессоров не используются; вводимая информация через канал поступает непосредственно в оперативную память, при выводе информация также не проходит по цепям процессоров. Иначе говоря, при выполнении операций ввода-вывода создаются цепочки от оперативной памяти (ОП) до периферийных, или внешних, устройств (ВУ), включающие в свой состав каналы и устройства управления внешних устройств (УВУ).

К периферийным относятся все устройства ввода-вывода, внешние запоминающие устройства, устройства передачи информации по линиям телефонной и телеграфной связи, а также другие средства связи пользователя с машиной. Периферийные устройства управляются посредством УВУ, обеспечивающих взаимодействие ВУ с каналами и совмещающих в себе средства управления данными (данным) ВУ и средства организации обмена информацией с каналом в стандартной форме. Унификация связей между УВУ и каналами необходима для удобства организации обмена информацией, а также для обеспечения объединения в одной машине различных наборов внешних устройств. В случае, когда одно УВУ управляет работой нескольких устройств, также необходима унификация связей между УВУ и ВУ, обычно являющимися внешними запоминающими устройствами (ВЗУ).

Задачи унификации связей решаются с помощью так называемых интерфейсов: интерфейса ввода-вывода и интерфейса УВУ — ВЗУ, иногда именуемых соответственно внешним и внутренним. Интерфейс ввода-вывода — это унифицированная система связей и сигналов между каналами ввода-вывода и устройствами управления периферийными устройствами; интерфейс УВУ — ВЗУ — это унифицированная система связей и сигналов между устройством управления и соответствующими внешними запо-

минающими устройствами. Все устройства, подключаемые к каналу посредством интерфейса ввода-вывода, обычно называются абонентами.

Собственно интерфейс ввода-вывода является стандартным и предназначен для обеспечения унифицированных средств сопряжения и управления каналах различных моделей ЭВМ Единой системы с различными УВУ. При этом достигается единообразие принципов обмена информацией, форматов информации, последовательностей управляющих сигналов, параметров сигналов, схем и конструкций электрических связей.

При выполнении операций ввода-вывода информация между каналом и ВУ всегда передается по одному байту; каждый байт представляет адрес, команду, информацию о состоянии ВУ или своеобразный квант данных. Однобайтный адрес используется для непосредственной адресации ВУ и характеризует конкретный путь связи ВУ с каналом. Байт команды передается каналом для указания ВУ или УВУ конкретного действия, которое оно должно выполнить. Во время выполнения операции ввода-вывода ВУ может реализовать одну или несколько команд, связанных программно в цепочку. В последнем случае в канале и УВУ организуется режим выполнения операции по цепочке команд.

Информация о состоянии ВУ передается в канал по окончании определенного этапа выполнения команды или после окончания выполнения всей команды. Особым видом информации о состоянии ВУ является информация уточненного состояния, которая предназначена для указания каналу детальных условий, связанных с возникновением сбойных ситуаций в ВУ или УВУ, а также сведений о характере прохождения диагностических процедур.

Данными называется информация, которая считывается в ВУ с носителя для передачи в оперативную память или выводится из оперативной памяти для записи на носитель ВУ. Передача данных производится в последовательности байт и, как правило, не зависит от характера размещения информации на носителях конкретных ВУ. Все байты данных имеют контрольный разряд (контроль на четность). При обнаружении ошибки в передаче данных схемами контроля могут выполняться следующие действия:

- прекращение выполнения операции ввода-вывода каналом или УВУ;
- исправление значения контрольного разряда с фиксацией в канале или УВУ ошибки;
- фиксация в УВУ ошибки без прекращения выполнения операции ввода-вывода.

Характеристики и возможности интерфейса ввода-вывода. Интерфейс ввода-вывода обеспечивает побайтную передачу информации от канала к абоненту или от абонента к каналу. При этом:

- число непосредственно адресуемых ВУ — 256;

- количество линий интерфейса — 38;
- число подключаемых к линиям интерфейса УВУ — 8 (10);
- скорость передачи данных — не менее 1300 кбайт/с.

Линии интерфейса выполняются как шины из коаксиального кабеля типа ИКМ-2; при подключении десяти УВУ длина линий интерфейса не должна превосходить 65 м, что определяется допустимыми задержками сигналов.

Интерфейс ввода-вывода обеспечивает:

- стандартную организацию выполнения операций ввода-вывода независимо от скорости работы ВУ и каналов, а также режима выполнения операции;
- простоту программирования операций ввода-вывода;
- возможность обнаружения сбоя или неисправности в аппаратуре ввода-вывода;
- возможность обмена данными в системе с несколькими машинами;
- возможность наращивания мощностей по вводу-выводу информации с расширением круга подсоединяемых ВУ.

Возможности по обмену данными и расширению круга подсоединяемых ВУ реализуются при использовании УВУ различных типов. Обычно выделяют следующие типы устройств управления периферийными устройствами:

- УВУ для одного ВУ, или эдиночное УВУ;
- групповые УВУ (просто групповые и объединенные);
- разделенное УВУ.

Одиночное УВУ допускает подключение к нему только одного периферийного устройства. В этом случае УВУ и ВУ выполняются, как правило, в виде единой конструкции. Групповое УВУ предназначено для обслуживания нескольких одготипных ВУ или устройств, незначительно отличающихся по характеру используемой информации и принципам управления. Обычное групповое УВУ в данный момент времени может обслуживать только одно ВУ; объединенное УВУ может обеспечить одновременную работу всех подсоединенных к нему ВУ. Группа внешних устройств может подсоединяться к нескольким групповым УВУ через коммутатор. В этом случае каждое УВУ может работать с любым ВУ из группы, не связанным по работе в данный момент времени с другим УВУ.

Разделенное УВУ имеет средства подсоединения к двум каналам, но на все время выполнения операции ввода-вывода оно всегда связано только с одним каналом.

К особому типу УВУ относится адаптер связи каналов, или адаптер канал — канал (АКК). Он предназначен для передачи данных из одной области оперативной памяти в другую или для связи между каналами двух процессоров в системе. В процессе работы адаптер всегда организует связь сразу с двумя каналами ввода-вывода.

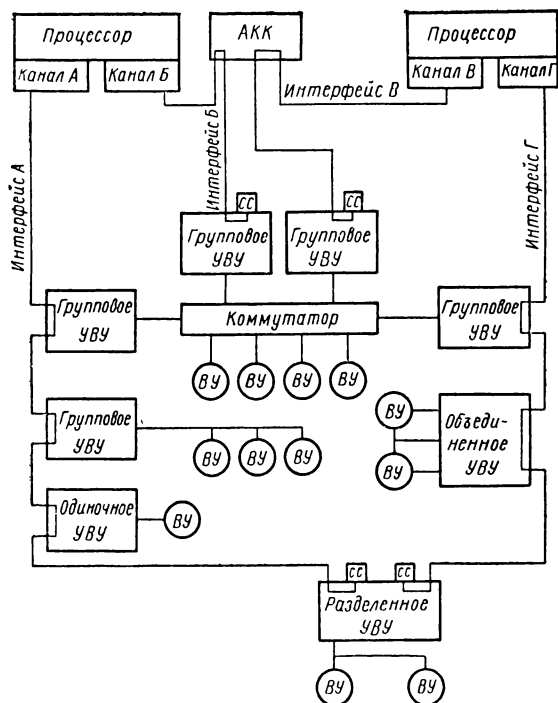


Рис. 10.1. Схема подсоединения внешних устройств

Возможности использования внешних устройств и способы их подключения к процессорам посредством УВУ и каналов ввода-вывода зависят прежде всего от того, к УВУ какого типа подключаются ВУ. Варианты подключения:

- ВУ связано только с одним процессором через один канал;
- ВУ связано с одним процессором по двум путям доступа через различные каналы;
- ВУ связано с двумя или несколькими процессорами.

Различные способы подключения внешних устройств к процессорам через каналы ввода-вывода и УВУ иллюстрируются схемой, приведенной на рис. 10.1.

Для согласования параметров линий интерфейса с параметрами узлов УВУ, применяются специальные схемы согласования (СС). Эти схемы на рис. 10.1 показаны для отдельных УВУ.

Организация связи УВУ с каналом заключается в логическом подключении первого ко второму; в любой момент времени только одно УВУ может быть логически подключено к каналу. Отключение УВУ от канала производится по окончании передачи информации, а также по специальным сигналам. Связь между УВУ и каналом осуществляется посредством обмена управляющими сигналами. Сигналы канала и абонента связаны по прин-

ципу «запрос — ответ», т. е. управление выдачей и сбросом всех сигналов в интерфейсе производится взаимосвязанными ответными сигналами.

10.2. СОСТАВ И НАЗНАЧЕНИЕ ЛИНИЙ ИНТЕРФЕЙСА

Информация и сигналы управления в интерфейсе ввода-вывода передаются посредством 34 функционально разделенных линий: 17 линий предназначены для передачи сигналов от канала к УВУ, 15 линий — для передачи сигналов от УВУ к каналу и две линии, так называемые линии выборки УВУ, образуют петлю, по которой сигнал от канала может последовательно обойти все УВУ и вернуться в канал.

Все сигналы, за исключением сигналов управления выборкой УВУ, поступают в канал и из канала по общим линиям. Это означает, что любой сигнал, выданный каналом, доступен всем УВУ; логическая же связь с каналом, как отмечалось выше, в любой момент времени обеспечивается только для одного УВУ. Подключение УВУ к каналу производится по специальному сигналу выборки в соответствии с установленным приоритетом.

Приоритет УВУ зависит от того, на каком месте оно включено в цепь «выборка—обратная выборка»; при этом с точки зрения обслуживания УВУ участки петлеобразной цепи, т. е. участки «выборка» и «обратная выборка», равнозначны. Таким образом, изменяя коммутацию схем управления выборкой внутри УВУ, можно устанавливать любой приоритет для УВУ, независимо от их удаленности от канала. Требуется только, чтобы любое УВУ, если оно в данный момент времени не работает, с малой задержкой ретранслировало сигнал выборки к следующему УВУ.

Принципы подключения УВУ к каналу через интерфейс ввода-вывода поясняются схемой, показанной на рис. 10.2,

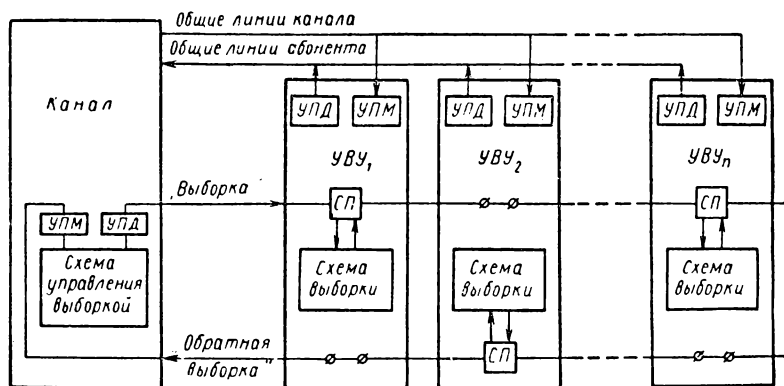


Рис. 10.2. Общая схема подключения УВУ к каналу через интерфейс ввода-вывода

где приняты следующие обозначения: УПД — усилители передатчики, УПМ — усилители-приемники, СП — схема переключений и ретрансляции сигналов выборки. В соответствии с принятыми принципами УВУ₁ имеет высший приоритет по обслуживанию, а УВУ₂ — низший.

Линии интерфейса по своему функциональному назначению делятся на четыре группы:

- информационные (18 линий);
- идентификации (6 линий);
- управления (7 линий);
- специальные (3 линии).

Информационные линии, служащие для побайтной передачи информации, включают в свой состав девять информационных шин канала (ШИН-К) и девять информационных шин абонента (ШИН-А); причем одна линия в ШИН-К и ШИН-А используется для передачи значения контрольного разряда. Линии идентификации служат для передачи сигналов, определяющих вид передаваемой информации и специальные последовательности сигналов интерфейса. По линиям управления передаются сигналы выборки и управления, а по специальным линиям — сигналы для смены состояния и измерения времени. Ниже приведены наименования и условные обозначения линий:

Наименование линий	Обозначения
Информационные:	
шины канала	ШИН-К
шины абонента	ШИН-А
Идентификации:	
адрес от канала	АДР К
адрес от абонента	АДР-А
управление от канала	УПР-К
управление от абонента	УПР-А
информация от канала	ИНФ-К
информация от абонента	ИНФ-А
Управления:	
работа канала	РАБ-К
работа абонента	РАБ-А
выборка	ВБР-К
разрешение выборки	РВБ-К
обратная выборка	ВБР-А
блокировка	БЛК-К
требование абонента	ТРБ-А
Специальные:	
измерение от канала	ИЗМ-К
измерение от абонента	ИЗМ-А
смена состояния	СМС-К

Информационные линии (шины). По шинам ШИН-К и ШИН-А передаются байты информации соответственно от канала к абоненту и от абонента к каналу. При этом принимается, что в случае передачи неполного байта информационные разряды располагаются на смежных линиях, начиная с линии, имеющей старший

(седьмой) номер; на неиспользуемых линиях в этом случае должны действовать сигналы кода 0. Контрольный разряд всегда передается по шине ШИН-КК (ШИН-АК); в передаваемом байте должно быть всегда нечетное число единиц, включая контрольный разряд.

При передаче сигналов по линиям интерфейса ввода-вывода возможны неравномерные задержки сигналов. Допустимые для большинства УВУ разбросы задержек сигналов на информационных шинах относительно сигналов идентификации, а также задержки на собственных схемах канала компенсируются каналом. В случае, когда задержки в схемах УВУ выходят за допустимые пределы, компенсация дополнительных задержек обеспечивается самим УВУ.

Информационные шины канала (ШИН-К) используются для передачи абонентам адресов ВУ, команд и данных, сопровождаемых сигналами¹ идентификации АДР-К, УПР-К и ИНФ-К соответственно. Период времени, в течение которого информация на ШИН-К считается действительной, определяется так:

— при передаче адреса ВУ информация на ШИН-К действительна с момента появления сигнала АДР-К до момента появления одного из сигналов РАБ-А, ВБР-А или сброса сигнала УПР-А (при последовательности сигналов выборки занятого УВУ);

— при передаче команды, другой управляющей информации или данных — с момента появления сигнала УПР-К (ИНФ-К) до момента сброса ответного сигнала на линиях идентификации абонента (АДР-А или ИНФ-А).

Информационные шины абонента (ШИН-А) используются для передачи каналу данных, адресов ВУ и информации о состоянии абонента, сопровождаемых сигналами идентификации ИНФ-А, АДР-А и УПР-А соответственно. Информация из УВУ выдается на шину ШИН-А только при действии сигнала в линии РАБ-А; исключением является случай последовательности сигналов выборки занятого УВУ, когда информация о состоянии выдается при отсутствии сигнала РАБ-А. Информация в канале считается действительной с момента появления сигнала идентификации от абонента до момента появления ответного сигнала идентификации от канала или до момента сброса сигнала ВБР-К в случае последовательности сигналов выборки занятого УВУ.

Линии идентификации. *Линия АДР-К* (адрес от канала) — это линия от канала ко всем абонентам, сигнал в которую выдается:

- а) в последовательности сигналов начальной выборки;
- б) при отключении от интерфейса.

При начальной выборке ВУ появление сигнала АДР-К означает, что на ШИН-К находится адрес ВУ, с которым канал начинает операцию. После выдачи АДР-К канал выдает сигнал ВБР-К. Если сигнал АДР-К сбрасывается до выдачи сигнала ВБР-К, то последовательность начальной выборки аннулируется. На

¹ Здесь и далее сигналы, проходящие по линиям, обозначаются как и линии.

сигналы АДР-К и ВБР-К опознавшее их УВУ отвечает сигналом РАБ-А; если УВУ занято, то оно отвечает сигналом УПР-А. В последовательности отключения от интерфейса сигнал АДР-К выдается, когда РВБ-К уже сброшен или выдача сигнала АДР-К сопровождается сбросом сигнала РВБ-К.

Линия АДР-А (адрес от абонента) — это линия от всех подсоединенных УВУ к каналу, действующий в которой сигнал указывает каналу, что адрес выбранного ВУ находится на ШИН-А. Канал на сигнал АДР-А отвечает сигналом УПР-К. Сигнал АДР-А должен сохраняться до появления сигнала УПР-К, после чего он сбрасывается, разрешая сброс УПР-К.

Линия УПР-К (управление от канала) — это линия канала ко всем УВУ, сигнал в которую посылается в ответ на сигналы АДР-А, УПР-А и ИНФ-А; сигнал УПР-К всегда должен сохраняться до сброса сигнала, его вызвавшего, т. е. АДР-А, УПР-А или ИНФ-А. Выдача сигнала УПР-К означает:

— в ответ на АДР-А в последовательности начальной выборки — выдачу каналом на ШИН-К байта команды, задающей выполняемую операцию;

— в ответ на АДР-А в последовательности сигналов, вводимой абонентом, — указание «Продолжить», по которому ВУ может продолжить последовательности, связанные с текущей операцией;

— в ответ на ИНФ-А — последовательность сигналов управления «Останов»;

— в ответ на УПР-А — последовательность сигналов управления «Запомнить состояние».

Во всех случаях, когда сигнал УПР-К означает указание «Продолжить» или последовательность сигналов управления, на ШИН-К должны находиться сигналы кода нуля; значение контрольного разряда безразлично, так как в указанных случаях контроль информации в канале или УВУ не производится.

Линия УПР-А (управление от абонента) — это линия от всех подсоединенных УВУ к каналу, сигнал в которой используется для указания каналу о том, что байт состояния выбранного ВУ находится на ШИН-А. Канал на сигнал УПР-А отвечает сигналом ИНФ-К или сигналом УПР-К. Выдача сигнала ИНФ-К означает, что байт состояния принят каналом. Если байт состояния не может быть воспринят каналом, то канал формирует последовательность сигналов управления «Запомнить состояние», выдавая сигнал УПР-К. Сигнал УПР-А сохраняется до появления ответного сигнала идентификации от канала или, в случае последовательности сигналов выборки занятого УВУ, до сброса сигнала ВБР-К.

Линия ИНФ-К (информация от канала) — это линия от канала ко всем абонентам, сигнал в которую направляется в ответ на ИНФ-А или УПР-А для сообщения ВУ, что эти сигналы восприняты каналом. При выполнении команд СЧИТАТЬ; СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ, УТОЧНИТЬ СОСТОЯНИЕ сигнал ИНФ-К, как ответ на ИНФ-А, указывает УВУ, что каналом при-

нята информация, находящаяся на ШИН-А, и что эта информация может сниматься с ШИН-А. При выполнении команд ЗАПИСАТЬ или УПРАВЛЕНИЕ сигнал ИНФ-К, как ответ на ИНФ-А, указывает УВУ, что на ШИН-К находится требуемая информация. Сигнал ИНФ-К, как ответ на УПР-А, означает: «Состояние принято». Во всех случаях сигнал ИНФ-К должен сохраняться, пока не будет сброшен соответствующий сигнал ИНФ-А или УПР-А.

Линия ИНФ-А (информация от абонента) — это линия от всех УВУ к каналу, сигнал в которую направляется для того, чтобы сообщить каналу, что УВУ необходимо получить или передать байт информации; вид информации зависит от выполняемой команды и типа работающего ВУ. При выполнении команд СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ, УТОЧНИТЬ СОСТОЯНИЕ сигнал ИНФ-А выдается как указание о наличии на ШИН-А байта данных или информации уточненного состояния. При выполнении команд ЗАПИСАТЬ или УПРАВЛЕНИЕ сигнал ИНФ-А означает запрос от УВУ на выдачу каналом байта данных или информации управления.

Канал отвечает на сигнал ИНФ-А сигналом ИНФ-К, УПР-К или сигналом АДР-К, если выполняется последовательность «Отключение от интерфейса». Сигнал ИНФ-А сбрасывается при поступлении одного из этих сигналов идентификации от канала. Для ВУ, в которых частота обмена информацией задается скоростью непрерывного движения носителя, возможна ситуация переполнения, когда канал не может вовремя ответить на запрос УВУ о передаче данных. Условия переполнения распознаются в ВУ, где формируются указатель состояния СБОЙ В УСТРОЙСТВЕ и указатель уточненного состояния ПЕРЕПОЛНЕНИЕ. Передача данных прекращается, но УВУ может сбросить сигнал ИНФ-А только после поступления одного из сигналов идентификации от канала.

Линии управления. Линия РАБ-К (работа канала) — это линия от канала ко всем УВУ, действующий в которой сигнал используется для разрешения подключения УВУ к каналу; все сигналы от канала, за исключением сигнала БЛК-К, действительны только при наличии сигнала РАБ-К. Сброс сигнала РАБ-К вызывает сброс всех сигналов абонентов, действие которых должно быть прекращено не более, чем через 1,5 мкс после окончания действия сигнала РАБ-К. Сброс сигнала РАБ-К при наличии сигнала БЛК-К означает переход к управляющей последовательности сигналов «Селективный сброс», в противном случае — к последовательности «Сброс системы».

Линия РАБ-А (работа абонента) — это линия от всех УВУ к каналу, действующий в которой сигнал указывает, что выборка УВУ осуществлена, и оно подключилось к каналу; адрес выбранного ВУ передается по ШИН-А и идентифицируется сигналом АДР-А. Сигнал РАБ-А выдается только при наличии на входной линии УВУ сигнала ВБР-К и блокировании его дальнейшего

прохождения по линиям выборки (на выходной линии УВУ сигнал ВБР-К отсутствует). Сигнал РАБ-А сохраняется на все время подключения УВУ к каналу и может быть сброшен только при сбросе каналом сигнала ВБР-К. Все сигналы абонента, за исключением ТРБ-А и ИЗМ-А, должны быть сброшены не более, чем через 1,5 мкс после сброса сигнала РАБ-А в УВУ.

Линии ВБР-К, ВБР-А, РВБ-К и соответствующие им сигналы обеспечивают управление выборкой УВУ. Линии ВБР-К и ВБР-А (выборка и обратная выборка) образуют петлю, подключение к которой схем выборки УВУ определяет их приоритет при установлении связей с каналом (см. рис. 10.2). Подключение схемы выборки УВУ к линии ВБР-К или к линии ВБР-А осуществляется с помощью переключателя или путем перестановки ТЭЗ коммутации. Каждое УВУ должно обеспечивать распространение сигнала ВБР-К независимо от того, подключено ли оно к интерфейсу, работает автономно или выключено; при этом задержка сигнала ВБР-К в одном УВУ не должна превышать 1,8 мкс.

При операции, вводимой каналом, выдаче сигнала ВБР-К должна предшествовать выдача сигнала АДР-К, причем не менее, чем на 400 нс. Канал сохраняет сигнал на линии ВБР-К до получения одного из сигналов ВБР-А (выборка не произошла — сигнал выборки возвратился в канал), РАБ-А (совместно с АДР-А) или УПР-А. УВУ считается выбранным и логически подключенным к каналу с момента выдачи им сигнала РАБ-А. Занятое УВУ подключается к каналу при выборке только для передачи байта состояния и выдает один сигнал УПР-А. Сигнал по линии РВБ-К, которая представляет собой линию от канала ко всем УВУ, всегда действует одновременно с сигналом ВБР-К; он обеспечивает сохранение сигнала в цепи выборки при изменении уровня сигнала ВБР-К во время переходных процессов в УВУ и уменьшение времени затухания сигнала ВБР-К в цепи выборки при сбросе. Сигнал РВБ-К может формироваться вновь после его сброса не ранее, чем через 4 мкс (2 мкс для быстродействующих каналов).

Линия БЛК-К (блокировка) — это линия от канала ко всем УВУ, сигнал в которую выдается для формирования следующих управляющих последовательностей: «Блокировка данных», «Блокировка состояния», «Цепочка команд», «Селективный сброс».

Линия ТРБ-А (требование абонента) — это линия от всех УВУ к каналу, сигнал в которую выдается тогда, когда УВУ необходимо установить связь с каналом для передачи данных или информации о состоянии абонента. Сигнал ТРБ-А сбрасывается:

— после выдачи сигнала РАБ-А, если не требуется дополнительных последовательностей выборки (задержка — не более 250 нс после сброса РАБ-А);

— в случае, когда УВУ не готово к выдаче данных или информации о состоянии абонента;

— если требование на выборку УВУ было реализовано по другому пути доступа.

Специальные линии. Линия СМС-К (смена состояния) — это линия от канала ко всем УВУ, сигнал в которую выдается для блокировки со стороны процессора возможности смены состояния УВУ в любое время; смена состояния УВУ может происходить только тогда, когда сигнал СМС-К сброшен. При этом смена состояния означает переход из состояния ДОСТУПЕН в состояние НЕ ДОСТУПЕН или наоборот, а также из режима «Работа» в режим «Автономная работа» или наоборот.

Линия ИЗМ-К (измерение от канала) — это линия от канала ко всем УВУ; сигнал, действующий в ней, используется подсоединенными к интерфейсу УВУ для включения счетчиков времени в УВУ и ВУ. Сигнал ИЗМ-К выдается при включении счетчика времени в процессоре.

Линия ИЗМ-А (измерение от абонента) — это линия от всех УВУ к каналу; сигнал, действующий в ней, может использоваться процессором для определения времени выполнения операции ввода — вывода. Сигнал ИЗМ-А выдается ВУ или УВУ в момент приема команды и сохраняется до конца операции, т. е. до формирования указателя ВУ КОНЧИЛО. Сигнал ИЗМ-А, как и сигнал ТРБ-А, может формироваться одновременно несколькими УВУ.

Взаимная блокировка сигналов. Сигналы, посылаемые каналом и УВУ в линии интерфейса ввода-вывода, представляют собой электрические сигналы постоянного тока. Формирование сигналов производится в соответствии с их взаимодействием, отвечающем обмену сигналами между каналом и абонентами по принципу «запрос—ответ». Так, сигнал идентификации от канала и сигнал идентификации от абонента составляют пару взаимноблокирующих сигналов. Это означает, что выдача сигнала абонента разрешает выдачу сигнала от канала, выдача сигнала от канала разрешает сброс сигнала абонента, сброс сигнала абонента разрешает сброс сигнала от канала, сброс сигнала от канала завершает работу пары сигналов и разрешает выдачу любого другого сигнала от абонента. Прием сигналов интерфейса в канале и УВУ всегда обеспечивается так, что принимающее устройство сначала фиксирует изменение входного сигнала, а потом формирует и выдает ответный сигнал.

Нормально функционирующие канал и УВУ всегда отвечают следующим требованиям на взаимную блокировку сигналов:

1) только один сигнал идентификации может быть выдан каналом в любой момент времени (исключение составляет сигнал АДР-К в течение действия последовательности сигналов отключения от интерфейса, когда он может выдаваться одновременно с другими сигналами);

2) только один сигнал идентификации может быть выдан абонентом в любой момент времени;

3) сигнал идентификации абонента может быть выдан только тогда, когда все сигналы идентификации канала сброшены (исключение делается только для последовательности сигналов выборки занятого УВУ);

4) сигнал идентификации абонента может быть сброшен только после выдачи ответного сигнала идентификации канала (за исключением сигнала УПР-А в последовательности сигналов выборки занятого УВУ);

5) сигналы ИНФ-К и УПР-К могут быть выданы только при наличии одного из сигналов идентификации абонента;

6) сигнал АДР-К в случае последовательности сигналов начальной выборки может быть выдан только тогда, когда будут сброшены сигналы ВБР-К и ВБР-А;

7) если сигналы АДР-К и ВБР-К выданы в последовательности сигналов начальной выборки, то АДР-К должен сохраняться до получения сигнала ВБР-А или РАБ-А или до сброса сигнала УПР-А;

8) если сигнал АДР-К выдан в последовательности сигналов отключения от интерфейса, то его нельзя сбросить до тех пор, пока не будет сброшен сигнал РАБ-А;

9) сигналы на всех выходных линиях канала, за исключением линии БЛК-К, должны быть сброшены, если сброшен сигнал на линии РАБ-К;

10) сигнал ВБР-К может быть выдан, если сигналы РАБ-А и ВБР-А сброшены;

11) сигнал РАБ-А может быть сброшен при выполнении одного из следующих трех условий:

— сброшен сигнал ВБР-К и ответный сигнал идентификации канала выдан в ответ на последний сигнал идентификации абонента для любой последовательности сигналов интерфейса;

— сброшен сигнал на линии РАБ-К;

— выполняется последовательность сигналов отключения от интерфейса;

12) сигнал РАБ-А не может быть выдан, если нет сигнала РАБ-К; он должен быть сброшен, если сброшен сигнал РАБ-К.

В том случае, когда в УВУ или канале выявляется нарушение требований на взаимную блокировку сигналов, производится формирование сигнала, свидетельствующего о наличии сбойной ситуации.

10.3. ВЫПОЛНЕНИЕ ОПЕРАЦИЙ

Операция ввода-вывода задается процессором при реализации им инструкции НАЧАТЬ ВВОД-ВЫВОД. В соответствии с этой инструкцией канал выбирает из оперативной памяти управляющую информацию и организует непосредственное выполнение операции ввода-вывода; завершив операцию, канал посылает в процессор запрос на прерывание текущей программы с целью выдачи информации о результатах выполнения операции. В процессе выполнения операции ввода-вывода может исполняться одна команда или несколько связанных в цепочку команд ввода-вывода. В последнем случае каждой команде отвечает отдельный этап операции ввода-

вывода, включающий в себя передачу команды УВУ, ее исполнение и получение от ВУ информации состояния, которая характеризует выполнение команды.

По каждой команде канал организует через интерфейс ввода-вывода связь с ВУ и занимает его до конца операции, связанной с данной командой. В операциях ввода-вывода условно выделяют три части: начальная выборка, передача данных, окончание. Выполняемость этих частей, а также связь между ними зависят от типа канала, режима выполнения операции, разновидности исполняемой команды, состояния ВУ и УВУ.

В селекторном канале возможен только монополюсный режим выполнения операций, когда канал монополизует интерфейс ввода-вывода до конца операции или до приема байта состояния по завершении передачи данных. Следует отметить, что для селекторного канала характерны операции ввода-вывода, выполняемые в виде цепочки команд. Канал занимает путь доступа к ВУ на время исполнения всех команд из цепочки, хотя между отдельными операциями в интерфейсе, связанными с каждой из команд, ВУ отключается от канала.

В мультиплексном канале операция может выполняться как в мультиплексном, так и в монополюсном режимах, причем режим работы задается ВУ. Мультиплексный режим обеспечивает параллельное выполнение в канале нескольких операций ввода-вывода и характеризуется тем, что каждая операция разбивается на интервалы, чередующиеся по выполнению с интервалами других операций. Интервал операции состоит из подключения УВУ к каналу и передачи одного или нескольких байт данных или байта информации состояния.

Продолжительность связи между каналом и УВУ в общем случае может задаваться как каналом, так и абонентом. Канал задает продолжительность связи длительностью своих сигналов ВБР-К и РВБ-К. В селекторном канале эти сигналы сохраняются на время всей операции или до приема байта состояния; УВУ тоже может задавать монополюсный режим, сохраняя сигнал РАБ-А после сброса сигналов ВБР-К и РВБ-К.

В мультиплексном канале сигнал ВБР-К сбрасывается сразу же после поступления от УВУ адреса абонента, поэтому режим работы зависит только от ВУ, которое задает его, сбрасывая или поддерживая сигнал РАБ-А. Канал выявляет режим выполнения операции по временному интервалу, в течение которого УВУ поддерживает связь с каналом через интерфейс. Считается, что передача данных осуществляется в мультиплексном режиме, если время, затраченное УВУ на связь с каналом по одной последовательности сигналов в интерфейсе, не превышает 32 мкс.

По характеру выполнения операции делятся на свободно выполняемые и немедленно выполняемые. К свободно выполняемым относятся операции, связанные с передачей информации между каналом и абонентом, к немедленно выполняемым — те,

которые не требуют передачи байтов данных (или другой информации), и прием команды совпадает с формированием указателя состояния КАНАЛ КОНЧИЛ.

Выполнение операций в интерфейсе ввода-вывода обеспечивается посредством набора основных последовательностей сигналов и последовательностей сигналов управления. К основным относятся следующие последовательности сигналов:

- начальная выборка;
- выборка занятого УВУ;
- выборка, вводимая УВУ;
- передача данных.

Последовательности сигналов управления организуют переходы от данной части операции к последующей, управляют ходом выполнения операции, обеспечивают прекращение выполнения текущей операции.

Начальная выборка. Эта выборка представляет собой первую часть любой выполняемой операции. Она предназначена для выборки адресуемого ВУ с целью передачи ему команды и получения от ВУ байта состояния, характеризующего возможность исполнения этой команды. Последовательность сигналов начальной выборки для мультиплексного канала приведена на рис. 10.3; аналогичная последовательность формируется и в случае селекторного канала.

Для начала операции канал выдает на ШИН-К адрес выбираемого ВУ и через 250 нс — сигнал АДР-К. Все УВУ декодируют адрес, но только одно из них опознает его, как собственный; при этом адрес воспринимается только в том случае, если он содержит правильный контрольный разряд. Через 400 нс после сигнала АДР-К канал выдает сигнал ВБР-К (и РВБ-К). Когда этот сигнал появляется на входе адресуемого УВУ, оно блокирует дальнейшее его распространение и выдает в канал сигнал РАБ-А.

В ответ на сигнал РАБ-А канал сбрасывает сигнал АДР-К. После этого УВУ выдает на ШИН-А адрес выбранного ВУ, сопровождая его сигналом АДР-А. Мультиплексный канал, получив сигнал по линии АДР-А, может сбрасывать сигналы ВБР-К и РВБ-К. Произведя анализ поступившего адреса на его совпадение с посланным, канал выдает на ШИН-К байт команды и сопровождает его сигналом на линии УПР-К.

Выбранное УВУ принимает команду и сбрасывает сигнал АДР-А, позволяя каналу сбросить сигнал УПР-К. Зафиксировав

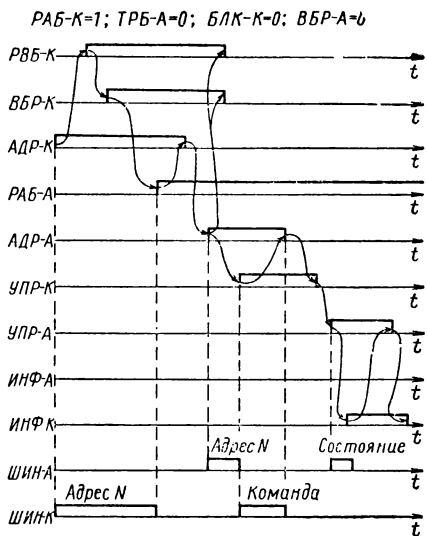


Рис. 10.3. Последовательность сигналов начальной выборки в мультиплексном канале

сброс сигнала УПР-К, УВУ помещает на ШИН-А байт состояния и сопровождает его сигналом идентификации по линии УПР-А. Канал отвечает на сигнал УПР-А выдачей сигнала ИНФ-К (или УПР-К), который разрешает УВУ сбросить сигнал УПР-А и тем самым завершить начальную выборку. После сброса сигнала УПР-А канал может сбросить сигнал ИНФ-К (УПР-К).

Выборка занятого УВУ. При обращении канала к некоторому внешнему устройству может оказаться, что УВУ, к которому оно подключено, занято или находится в состоянии ожидания прерывания для другого ВУ. В этом случае УВУ передает в канал байт состояния с указанием условий (причин) занятости либо в процессе выполнения последовательности сигналов начальной выборки, либо при выполнении последовательности сигналов выборки занятого УВУ. Способ передачи зависит от типа УВУ. Использование последовательности сигналов выборки занятого УВУ характерно для разделенных групповых УВУ; при этом байт состояния передается в канал с меньшими затратами времени, чем при использовании последовательности начальной выборки.

Последовательность сигналов выборки занятого УВУ начинается каналом так же, как и последовательность сигналов начальной выборки (рис. 10.4). Канал выдает на ШИН-К адрес ВУ, затем сигнал АДР-К и, наконец, сигнал ВБР-К. Выбираемое УВУ при поступлении сигнала ВБР-К выдает на ШИН-А байт состояния с указателями ЗАНЯТО и МОДИФИКАТОР, сопровождая этот байт сигналом идентификации УПР-А; дальнейшее распространение сигнала ВБР-К блокируется; сигнал РАБ-А в рассматриваемом случае не формируется.

После приема байта состояния канал сбрасывает сигнал ВБР-К. Вызывавшееся для установления связи УВУ отвечает сбросом сигнала УПР-А и отключается от интерфейса. Канал по сбросу УПР-А сбрасывает сигнал АДР-К, завершая тем самым выполнение рассматриваемой последовательности сигналов.

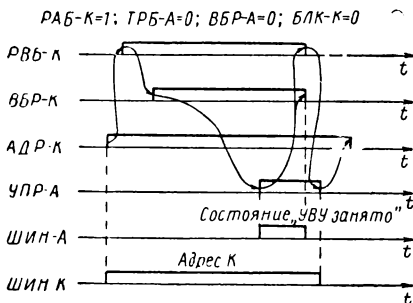


Рис. 10.4. Последовательность сигналов выборки занятого УВУ

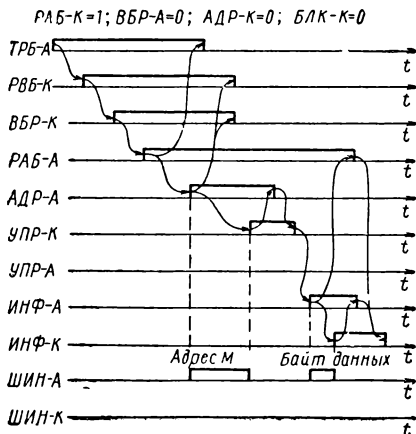


Рис. 10.5. Последовательность сигналов выборки, вводимой УВУ, на передачу данных

Выборка, вводимая УВУ.

Обращение к каналу со стороны УВУ для передачи байта данных или байта состояния производится путем послышки от УВУ к каналу сигнала ТРБ-А. Запрос УВУ на обслуживание удовлетворяется посредством выполнения последовательности сигналов выборки, вводимой УВУ (рис. 10.5).

Канал разрешает начать выполнение этой последовательности, выдавая сигнал ВБР-К (сигнал АДР-К при этом сброшен). УВУ, пославшее запрос на обслуживание, блокирует дальнейшее распространение сигнала ВБР-К, выдает сигнал РАБ-А и адрес ВУ на ШИН-А, сопровождая его сигналом идентификации АДР-А. Канал опознает адрес ВУ и посылает сигнал УПР-К, указывая УВУ, что последовательность сигналов может быть продолжена. В ответ УВУ сбрасывает сигнал АДР-А, после чего канал сбрасывает сигнал УПР-К. При передаче байта данных вслед за сбросом сигнала УПР-К на ШИН-А посылаются кодовые сигналы, сопровождаемые сигналом идентификации ИНФ-А. Канал в ответ формирует сигнал ИНФ-К; параллельно осуществляется сброс сигнала РАБ-А. УВУ в ответ на сигнал ИНФ-К сбрасывает сигнал ИНФ-А; после этого сбрасывается и сигнал ИНФ-К. Последовательность сигналов завершена. Отметим, что такая последовательность (и запрос на передачу байта данных) возможна только при выполнении операции в мультиплексном режиме.

В случае запроса на передачу информации о состоянии ВУ (УВУ) последовательность сигналов выборки, вводимая УВУ, включает передачу только одного байта состояния. Цикл передачи байта состояния выполняется так же, как при начальной выборке.

Передача данных. При выполнении операции ввода-вывода передача данных производится вслед за начальной выборкой, если принятая УВУ команда указывает на необходимость обмена данными между каналом и УВУ. Характер обмена также определяется командой. Так, команды СЧИТАТЬ, СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ, УТОЧНИТЬ СОСТОЯНИЕ задают действия по считыванию информации из УВУ, а команды ЗАПИСАТЬ и УПРАВЛЕНИЕ задают действия по записи информации в УВУ и далее в ВУ.

При считывании УВУ выдает байт данных на ШИН-А, сопровождая его сигналом ИНФ-А (рис. 10.6, а). Сигнал идентификации и данные на ШИН-А должны сохраняться до поступления из канала сигнала ИНФ-К, который как бы сигнализирует УВУ о том,

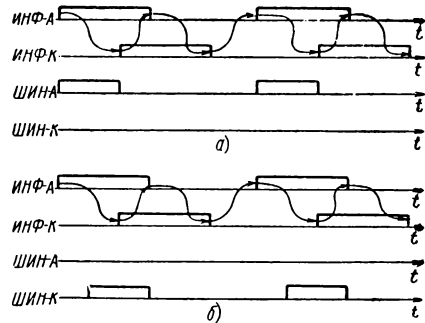


Рис. 10.6. Последовательность сигналов передачи данных при:

а — считывании; б — записи

что данные приняты. Сброс сигнала ИНФ-А разрешает сброс сигнала ИНФ-К. После этого УВУ может выдать новый байт данных.

При записи, путем выдачи сигнала ИНФ-А, работающее УВУ запрашивает байт данных от канала. В ответ канал выдает байт данных и с некоторой задержкой — сигнал идентификации ИНФ-К. Этот сигнал является признаком готовности данных, своеобразным разрешением УВУ принять байт данных. Канал сохраняет данные на ШИН-К и сигнал ИНФ-К до сброса сигнала ИНФ-А (см. рис. 10.6, б). После сброса сигнала ИНФ-А канал сбрасывает сигнал ИНФ-К, разрешая УВУ послать новый запрос на выдачу данных.

Операция передачи данных может осуществляться как в монопольном, так и в мультиплексном режиме. При реализации монопольного режима после начальной выборки УВУ остается подключенным к интерфейсу, и осуществляется передача всего блока данных, заданного выполняемой операцией. При реализации мультиплексного режима в конце начальной выборки УВУ отключается от интерфейса; для передачи одного байта или группы байт данных УВУ каждый раз посылает в канал запрос на обслуживание, который удовлетворяется путем выполнения последовательности сигналов выборки, вводимой УВУ. Количество байт в группе, передаваемой в ходе реализации одной последовательности сигналов выборки при мультиплексном режиме, определяется, исходя из того, что УВУ должно затрачивать на выполнение такой последовательности не более 32 мкс.

Последовательности сигналов управления. Такие последовательности, как указывалось выше, дают возможность каналу управлять ходом операции во время выполнения основных последовательностей сигналов, прерывать выполнение текущей операции, связывать две операции из цепочки, производить сброс УВУ и ВУ. К ним относятся следующие последовательности сигналов: «Останов», «Запомнить состояние», «Блокировка состояния», «Блокировка данных», «Цепочка команд», «Отключение от интерфейса», «Селективный сброс», «Сброс системы».

Признак последовательности «Останов» фиксируется УВУ при появлении сигнала УПР-К в ответ на сигнал ИНФ-А или как результат отключения от интерфейса, который произошел до поступления байта состояния с указателем КАНАЛ КОНЧИЛ для текущей операции. «Останов» используется для сигнализации ВУ о том, что канал заканчивает выполнение операции ввода-вывода. Это происходит либо вследствие того, что завершена передача заданного количества байт, либо при реализации в мультиплексном канале инструкции процессора ОСТАНОВИТЬ ВВОД-ВЫВОД. При получении признака «Останов» ВУ должно перейти от передачи данных к выполнению окончания операции; после этого УВУ уже не выдает сигнал ИНФ-А. ВУ остается занятым до тех пор, пока не будут сформированы и приняты каналом указатели КАНАЛ КОНЧИЛ и ВУ КОНЧИЛО.

Признаком последовательности сигналов «Запомнить состояние» является появление сигнала УПР-К в ответ на сигнал УПР-А. Этим канал указывает УВУ, что он не может принять байт состояния, который должен поэтому храниться в УВУ до его последующей выдачи. Восприняв указание «Запомнить состояние», УВУ отключается от интерфейса, сбрасывая сигнал РАБ-А сразу же после сброса сигнала ВБР-К. Канал сохраняет сигнал УПР-К до сброса сигнала РАБ-А.

«Блокировка состояния» выполняется, когда канал выдает сигнал БЛК-К; это осуществляется, если канал не может немедленно обработать байт состояния, вызывающий запрос на прерывание в процессор. Любая информация о состоянии становится блокируемой по получении УВУ признака «Запомнить состояние». Информация о состоянии, содержащая указатель КАНАЛ КОНЧИЛ, является неблокируемой до тех пор, пока в ответ на ее передачу канал не выдает признак «Запомнить состояние». Указатель КАНАЛ КОНЧИЛ допускает блокировку, если он сформирован в результате выполнения последовательности «Отключение от интерфейса». Информация о состоянии, содержащая указатель ВУ КОНЧИЛО, допускает блокировку, за исключением случая, когда УВУ определило наличие цепочки команд для ВУ, сформировавшего этот указатель. При такой ситуации информация о состоянии становится блокируемой по указанию «Запомнить состояние».

«Блокировка данных» используется для регулирования скорости передачи данных в монопольном режиме при работе с ВУ, имеющими буферную память, при работе с ВУ, для которых предусмотрен стартовый режим, а также при выполнении операции по передаче информации уточненного состояния. «Блокировка данных» производится путем выдачи сигнала БЛК-К не менее чем на 250 нс раньше сброса сигнала ИНФ-К или на 250 нс раньше выдачи этого сигнала. В этом случае УВУ не должно выдавать сигнал ИНФ-А для последующей передачи данных. Отметим, что первый байт данных после начальной выборки, как правило, не блокируется.

Признаком последовательности сигналов управления «Цепочка команд» служит появление сигнала ИНФ-К в ответ на сигнал УПР-А при наличии сигнала БЛК-К. Для ВУ, подключенного к интерфейсу, этот признак отвечает, что после представления каналу указателя ВУ КОНЧИЛО немедленно посредством новой начальной выборки будет передана очередная команда (если во время выполнения текущей операции не были обнаружены необычные для нормального течения операции условия). ВУ остается доступным до начала следующей операции из цепочки или до аннулирования цепочки команд. Если признак «Цепочка команд» фиксируется в процессе выдачи байта состояния с указателем ВУ КОНЧИЛО, то он действует до тех пор, пока не будет произведена новая выборка УВУ или не будет сброшен сигнал БЛК-К. Последующая выборка любого ВУ, подсоединенного к УВУ, которое

зафиксировало признак «Цепочка команд», отменяет условие «Цепочка команд» в УВУ. Указатели байта состояния СБОЙ В УСТРОЙСТВЕ, ОСОБЫЙ СЛУЧАЙ, ВУ КОНЧИЛО, ВНИМАНИЕ прерывают выполнение цепочки команд в канале.

Указание «Отключение от интерфейса» распознается УВУ, если действует сигнал АДР-К, а сигнал ВБР-К (РВБ-К) снят не менее чем за 250 нс до завершения любой последовательности сигналов. В этом случае сигнал АДР-К может действовать одновременно с любым другим сигналом идентификации от канала. Последовательность сигналов «Отключение от интерфейса» завершается сбросом сигнала АДР-К, что возможно после сброса сигнала РАБ-А (рис. 10.7). УВУ выполняет рассматриваемое указание путем сброса всех сигналов на линиях абонента, при этом сигнал РАБ-А должен быть сброшен в течение 6 мкс после получения УВУ указания «Отключение от интерфейса». В процессе выполнения этого указания УВУ не формирует байт состояния. При операции ввода данные на ШИН-А недействительны после выдачи сигнала АДР-К, а при операции вывода данные на ШИН-К действительны до сброса сигналов ИНФ-А или РАБ-А. Если сигнал АДР-К выдается одновременно с другим сигналом идентификации от канала, то информация на ШИН-К остается действительной до сброса сигнала идентификации от абонента, связанного с этим сигналом идентификации от канала, или до сброса сигнала РАБ-А.

«Селективный сброс» используется для прекращения текущей операции в УВУ со сбросом всех указателей состояния для этой операции и остановом ВУ, выполняющим эту операцию, только в случае обнаружения каналом ошибки или при нарушении временных соотношений. Признаком последовательности сигналов управления «Селективный сброс» является появление сигнала БЛК-К и сброс сигнала РАБ-К. В этом случае сбрасывается сигнал РАБ-А, а соответствующее ВУ прекращает работу. Путь связи и доступа к этому ВУ занимает на все время выполнения процедуры сброса. Сигнал РАБ-К может вновь действовать только после сброса сигнала РАБ-А (рис. 10.8).

«Сброс системы», вызывающий сброс сигнала РАБ-А, сброс всех УВУ, ВУ и подготовку ВУ к процедуре начальной загрузки

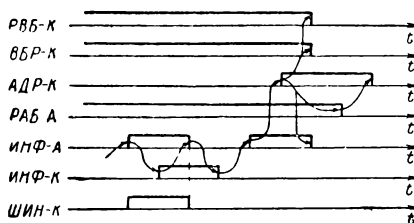


Рис. 10.7. Последовательность сигналов «Отключение от интерфейса»

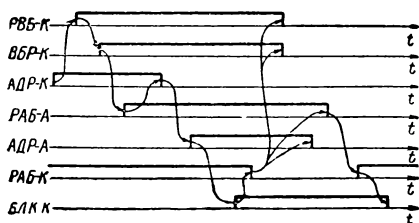


Рис. 10.8. Последовательность сигналов «Селективный сброс»

программы, выполняется при нажатии на пульте управления ЭВМ кнопки «СБРОС», при включении питания ЭВМ, при работе канала автономно от интерфейса, а так же как часть процедуры начальной загрузки программы. Признаком последовательности сигналов управления «Сброс системы» служит одновременное отсутствие действия сигналов РАБ-К и БЛК-К. Во время выполнения рассматриваемой последовательности все УВУ находятся в состоянии ЗАНЯТО.

Взаимосвязь последовательностей сигналов в операциях. Выполнение операции ввода-вывода, как правило, разделяется на три части: начальная выборка, передача данных, окончание; каждой части отвечает выполнение одной или нескольких последовательностей сигналов. Управление ходом выполнения и окончания определенной части и организация перехода от одной части к другой осуществляются посредством последовательностей управления.

При начальной выборке адресуемое ВУ выбирается только для передачи ему команды, определяющей действия ВУ во время выполнения операции. Команды ПРОВЕРИТЬ ВВОД-ВЫВОД и ХОЛОСТОЙ ХОД не требуют от ВУ выполнения каких-либо действий, поэтому они включают, как правило, только часть, отвечающую начальной выборке.

Если ВУ может выполнить команды, то в период начальной выборки оно выдает байт состояния, содержащий только нули и имеющий правильное значение контрольного разряда, или указатель КАНАЛ КОНЧИЛ для немедленно выполняемых команд. Если при обращении к ВУ с начальной выборкой оно занято выполнением операции, то в ответном байте состояния формируется указатель ЗАНЯТО.

Команда может быть отвергнута, например, по причине ее недействительности для данного ВУ. В этом случае в байте состояния, передаваемом в канал в процессе начальной выборки, УВУ формирует указатель СБОЙ В УСТРОЙСТВЕ, и операция в УВУ не начинается. При занятости УВУ последовательность начальной выборки может заменяться последовательностью сигналов выборки занятого УВУ. В случае неопознания адреса ни одним УВУ в канал возвращается сигнал ВБР-А, а последовательность сигналов начальной выборки аннулируется. Начальная выборка может прерываться при выполнении последовательностей сигналов управления «Отключение от интерфейса», «Селективный сброс», «Сброс системы».

Передача данных в монопольном режиме производится сразу по всему заданному количеству байт; скорость передачи может регулироваться посредством последовательности сигналов «Блокировка данных». В мультиплексном режиме передача одного байта или группы из нескольких байт производится по последовательности сигналов выборки, вводимой УВУ; при этом УВУ каждый раз посылает в канал запрос на обслуживание. Прерывание пере-

дачи данных в любом режиме производится последовательностями сигналов управления «Останов» или «Отключение от интерфейса». Передача байта уточненного состояния всегда производится с максимально возможной для данного ВУ скоростью.

Необходимость перехода к окончанию операции может определяться как каналом, так и ВУ. Если процедура окончания начата ВУ, то операция завершается выдачей байта состояния с указателями КАНАЛ КОНЧИЛ и ВУ КОНЧИЛО в одной последовательности, но при одновременном формировании этих указателей. В том случае, когда указатель ВУ КОНЧИЛО формируется не одновременно с указателем КАНАЛ КОНЧИЛ, байт состояния с указателем ВУ КОНЧИЛО передается посредством дополнительной последовательности сигналов после формирования этого указателя. Обычно после формирования указателя КАНАЛ КОНЧИЛ УВУ посылает запрос в канал на передачу байта состояния посредством последовательности выборки, вводимой УВУ.

Если во время реализации последовательностей сигналов в интерфейсе возникают сбойные ситуации, то канал идентифицирует их как состояние СВОЙ ИНТЕРФЕЙСА. Канал в этом случае прекращает выполнение операции посредством введения последовательностей сигналов управления «Отключение от интерфейса», «Селективный сброс».

10.4. УПРАВЛЯЮЩАЯ И СЛУЖЕБНАЯ ИНФОРМАЦИЯ

Команды ввода-вывода. В ходе реализации последовательности начальной выборки сигнал УПР-К означает, что канал выдал на ШИН-К байт команды. Младшие разряды байта команды определяют тип операции ввода-вывода, а старшие — модификацию основной операции, определяемую на уровне УВУ и ВУ и зависящую от специфики их построения и работы. Основные операции ввода-вывода и соответствующие им байты приведены в табл. 10.1, где буква М обозначает разряд модификатора, а К — разряд контроля по четности.

Т а б л и ц а 10.1

Кодирование основных команд ввода-вывода

Наименование команды	Разряды байта команды								
	К	0	1	2	3	4	5	6	7
СЧИТАТЬ (ЧИТАТЬ)	К	М	М	М	М	М	М	1	0
СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ	К	М	М	М	М	1	1	0	0
ЗАПИСАТЬ	К	М	М	М	М	М	М	0	1
УПРАВЛЕНИЕ	К	М	М	М	М	М	М	1	1
УТОЧНИТЬ СОСТОЯНИЕ	К	М	М	М	М	0	1	0	0
ПРОВЕРИТЬ ВВОД-ВЫВОД	1	0	0	0	0	0	0	0	0

Команды УТОЧНИТЬ СОСТОЯНИЕ, УПРАВЛЕНИЕ с нулевыми разрядами модификатора и команда ПРОВЕРИТЬ ВВОД-ВЫВОД выполняются всеми ВУ; они идентифицируются и декодируются в соответствии с данными табл. 10.2.

Т а б л и ц а 10.2

Кодирование дополнительных команд ввода-вывода

Наименование команды	Разряды байта команды								
	К	0	1	2	3	4	5	6	7
ОСНОВНОЕ УТОЧНЕНИЕ	0	0	0	0	0	0	1	0	0
ПРОВЕРИТЬ ВВОД-ВЫВОД	1	0	0	0	0	0	0	0	0
ХОЛОСТОЙ ХОД	1	0	0	0	0	0	0	1	1

По команде СЧИТАТЬ начинается операция передачи данных от УВУ к каналу и далее в оперативную память. Данные поступают в ОП в порядке последовательного увеличения адресов. Команда СЧИТАТЬ с нулевыми разрядами модификатора называется ОСНОВНОЕ СЧИТЫВАНИЕ и используется при загрузке программы (в соответствии с особенностями используемого ВУ). Команда ОСНОВНОЕ СЧИТЫВАНИЕ всегда должна быть первой для ВУ после реализации последовательности «Сброс системы».

По команде СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ производятся те же действия, что и по команде СЧИТАТЬ, но данные поступают в ОП в порядке последовательного уменьшения адресов. Эта команда предназначена для внешних устройств, которые имеют возможность перемещения носителя информации в направлении, которое противоположно используемому при выполнении операции СЧИТАТЬ.

При выполнении команды ЗАПИСАТЬ используются те же последовательности сигналов, что и для операции СЧИТАТЬ, но данные передаются от канала к абоненту.

Операция УПРАВЛЕНИЕ выполняется аналогично операции ЗАПИСАТЬ, но УВУ декодирует целиком весь код команды, включая и разряды модификатора, определяющие конкретную модификацию данной команды для абонента. Если при выполнении операции передача данных не требуется, то при реализации последовательности начальной выборки формируется и передается байт состояния с указателем КАНАЛ КОНЧИЛ. По команде ХОЛОСТОЙ ХОД внешнее устройство никакой операции не выполняет; исключения составляют случаи объединения операций в цепочку и ожидания момента готовности данных контроля.

Команда УТОЧНИТЬ СОСТОЯНИЕ используется для определения действительного состояния ВУ и получения точной информации о необычной ситуации, обнаруженной в ВУ при выполнении предыдущей операции. Она реализуется так же, как и команда СЧИТАТЬ, но информация поступает не с носителя информации,

а от индикаторов уточненного состояния абонента. Если при выполнении команды УТОЧНИТЬ СОСТОЯНИЕ в УВУ фиксируется ошибка, то в канал выдается байт состояния с указателями КАНАЛ КОНЧИЛ и СБОЙ В УСТРОЙСТВЕ. Команда ОСНОВНОЕ УТОЧНЕНИЕ, содержащая нули в разрядах модификатора, задает операцию считывания уточненного состояния для всех ВУ. Другие модификации команды УТОЧНИТЬ СОСТОЯНИЕ могут служить для обеспечения специальных диагностических процедур и операций.

Команда ПРОВЕРИТЬ ВВОД-ВЫВОД выполняется посредством нормальной последовательности сигналов начальной выборки, не вызывая каких-либо операций в ВУ. В ответ на эту команду ВУ всегда посылает байт состояния, который может быть нулевым, если нет информации состояния, подготовленной заранее для передачи в канал.

Адресация ВУ и УВУ. Для непосредственной адресации ВУ, подсоединенных к каналу посредством интерфейса, используется байт адреса; он позволяет адресовать 256 ВУ. Адрес ВУ присваивается при установке оборудования и характеризует конкретный путь доступа к ВУ со стороны канала. Присвоение адресов ВУ и УВУ производится в соответствии со следующими правилами.

1. Внешнему устройству, подключенному к одиночному УВУ, присваивается любой адрес от 0 до 255 при условии, что этот адрес не будет опознаваться любым другим УВУ. Такое ВУ представляет единое целое с УВУ, а ВУ и УВУ имеют один и тот же адрес.

2. При подключении к одному групповому УВУ нескольких ВУ (но не более 16), последним присваиваются смежные адреса из некоторой группы. Группа должна начинаться с адреса, в котором младшие разряды, определяющие состав группы, содержат нули. В этом случае старшие разряды байта адреса определяют УВУ, а младшие — номер ВУ для данного УВУ. Так, если к УВУ подключено семь внешних устройств, то адрес УВУ определяется разрядами (0—4) байта адреса, а адреса ВУ представляются так: 000, 001, 010, 011, 100, 101, 110.

3. При подключении к УВУ более 16 внешних устройств адреса присваиваются не обязательно смежными группами. Каждая группа содержит 16 адресов, за исключением той, которая дополняет число адресов до числа подключенных к УВУ внешних устройств. Так, если к УВУ подключается 56 ВУ, то три группы содержат по 16 адресов каждая, а одна — 8 адресов.

4. Внешние устройства, доступ к которым возможен от двух или более каналов, должны иметь различные адреса для каждого пути доступа. Для внешних устройств, подключенных к двум или более УВУ, часть адреса, определяющая номер ВУ, должна быть одинаковой для всех путей доступа.

Опознавание адреса ВУ осуществляется в УВУ, если поступивший байт адреса имеет правильный контрольный разряд и адрес принадлежит к числу присвоенных данному УВУ.

Считается, что ВУ находится в нерабочем состоянии, если ни одно УВУ не ответило каналу на посланный им адрес, т. е. сигнал ВБР-К возвратился в канал по линии ВБР-А. Нерабочее состояние ВУ фиксируется в следующих случаях:

- адрес не относится к числу адресов соответствующей группы;
- ВУ с данным адресом не подсоединено к ЭВМ;
- ВУ с данным адресом отключилось от системы в процессе выполнения программы.

УВУ отвечает на присвоенные адреса для ВУ, находящихся в состоянии ГОТОВ, а также для тех ВУ, находящихся в состоянии НЕ ГОТОВ, которые могут быть переведены в состояние готовности при обычном вмешательстве оператора. ВУ, находящееся в состоянии НЕ ГОТОВ, формирует указатель состояния СВОЙ В УСТРОЙСТВЕ и указатель уточненного состояния ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО.

Информация о состоянии ВУ и УВУ.
Появление сигнала на линии УПР-А означает, что на ШИН-А выдан байт состояния, относящийся к тому ВУ или УВУ, адрес которого выдавался на ШИН-А во время этапов опроса или выборки в последовательности сигналов. Байт состояния передается в канал:

- при начальной выборке;
 - для представления информации состояния с указателем КАНАЛ КОНЧИЛ по окончании передачи данных;
 - для представления информации состояния с указателем ВУ КОНЧИЛО и сопутствующими указателями;
 - для передачи в канал любой информации состояния, вызванного вмешательством извне;
 - для передачи хранящегося, ранее сформированного, байта состояния;
 - для представления информации состояния с указателями УВУ КОНЧИЛО или ВУ КОНЧИЛО с тем, чтобы сообщить каналу, что ранее занятое УВУ или ВУ освободилось для работы.
- Информация состояния, передаваемая в байте состояния, состоит из отдельных указателей, приведенных ниже.

Разряды байта	Наименование указателей	Разряды байта	Наименование указателей
К	Контрольный разряд	4	КАНАЛ КОНЧИЛ
0	ВНИМАНИЕ	5	ВУ КОНЧИЛО
1	МОДИФИКАТОР	6	СВОЙ В УСТРОЙСТВЕ
2	УВУ КОНЧИЛО	7	ОСОБЫЙ СЛУЧАЙ
3	ЗАНЯТО		

Указатель ВНИМАНИЕ не связан с началом, выполнением или окончанием операции ввода-вывода. Условия его формирования и способ представления в канал определяются типом ВУ. Представление в канал этого указателя блокирует выполнение цепочки команд.

Указатель МОДИФИКАТОР используется в следующих случаях:

1) если УВУ не может выдать байт состояния в ответ на команду ПРОВЕРИТЬ ВВОД-ВЫВОД, то в ходе начальной выборки оно выдает только один указатель МОДИФИКАТОР; это означает, что УВУ не может выполнить эту команду; такое УВУ выдает байт состояния только при реализации последовательности сигналов выборки, вводимой УВУ;

2) если УВУ занято (например, выполнением операции перемотки ленты для некоторого ВУ), то в ходе реализации последовательности сигналов начальной выборки оно выдает байт состояния с указателями МОДИФИКАТОР и ЗАНЯТО; это отмечает занятость именно УВУ, а не адресуемого ВУ;

3) если в УВУ предусматривается выдача специальных условий окончания операции ввода-вывода, то при выполнении заданных условий выдается указатель МОДИФИКАТОР вместе с указателем ВУ КОНЧИЛО.

Указатель УВУ КОНЧИЛО используется только теми УВУ, которые обслуживают несколько каналов, или групповыми УВУ. Он формируется при отдельном или совместном выполнении следующих условий:

— произошло обращение к занятому УВУ;

— УВУ, находясь в состоянии ЗАНЯТО, фиксирует необычную ситуацию после того, как канал воспринял указатель КАНАЛ КОНЧИЛ; в этом случае указатель УВУ КОНЧИЛО сопровождает выдачу указателя, характеризующего необычную ситуацию.

Рассматриваемый указатель выдается одновременно с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО или в промежутке между выдачами этих указателей состояния. Указатель УВУ КОНЧИЛО блокирует режим «Цепочка команд».

Указатель ЗАНЯТО формируется только при выполнении последовательности сигналов выборки, вводимой каналом. Он указывает на то, что ВУ или УВУ не может выполнить команду вследствие того, что выполняется ранее заданная операция или УВУ хранит непереданную в канал информацию состояния (последнее относится к разделенным УВУ, когда путь связи оказывается недоступным). Вместе с указателем ЗАНЯТО в байте состояния могут находиться и другие указатели состояния, сформированные к этому времени для адресуемого ВУ. Указатель ЗАНЯТО блокирует режим «Цепочка команд».

Указатель КАНАЛ КОНЧИЛ формируется при завершении части операции, связанной с передачей данных или управляющей информации между ВУ и каналом; при этом для каждой операции он формируется только один раз. Момент формирования рассматриваемого указателя зависит от типа ВУ и разновидности выполняемой операции. В операциях, которые не включают в свой состав передачу данных, указатель КАНАЛ КОНЧИЛ может

формироваться при реализации последовательности сигналов начальной выборки.

Указатель ВУ КОНЧИЛО формируется при завершении операции в ВУ или в результате перехода ВУ из состояния НЕ ГОТОВ в состояние ГОТОВ. При выполнении цепочки команд для программы оказывается доступным только тот указатель ВУ КОНЧИЛО, который связан с последней операцией цепочки. Если операция выполнена и нет признаков необычной ситуации, то наличие указателя ВУ КОНЧИЛО дает возможность каналу выдать следующую команду из цепочки; в противном случае выполнение цепочки приостанавливается без формирования и выдачи указателя ВУ КОНЧИЛО.

Переход любого ВУ из состояния НЕ ГОТОВ в состояние ГОТОВ вызывает, как отмечалось выше, формирование рассматриваемого указателя. Считается, что ВУ находится в состоянии НЕ ГОТОВ, если требуется вмешательство оператора, чтобы сделать его работоспособным. Состояние НЕ ГОТОВ соответствует, например, следующим случаям: не заправлена лента в накопителе на магнитной ленте, кончились карты в устройстве вывода на перфокарты и т. п.

Указатель СБОЙ В УСТРОЙСТВЕ фиксирует обнаружение в ВУ или УВУ необычных условий, информация о которых может передаваться по команде УТОЧНИТЬ СОСТОЯНИЕ. Формирование этого указателя происходит в случаях возникновения сбоев в работе оборудования ВУ или УВУ, наличия программной ошибки, а также при переходе ВУ в состояние НЕ ГОТОВ при исполнении команды. Указатель СБОЙ В УСТРОЙСТВЕ является общим для всех ошибок и сбоев; детальную информацию о них получают в результате исполнения команды УТОЧНИТЬ СОСТОЯНИЕ. Если ошибка или сбой не имеют отношения к уже введенной операции, то указатель СБОЙ В УСТРОЙСТВЕ не выдается в канал; однако, сбой может перевести устройство в состояние НЕ ГОТОВ.

Указатель ОСОБЫЙ СЛУЧАЙ формируется тогда, когда ВУ фиксирует положение, которое в обычных условиях не должно возникать (например, опознание ленточного маркера на магнитной ленте при выполнении операции СЧИТАТЬ). Этот указатель имеет единственное значение для каждой команды и каждого типа ВУ. В ответ на прием байта состояния с указателем ОСОБЫЙ СЛУЧАЙ введение операции УТОЧНИТЬ СОСТОЯНИЕ не требуется.

Информация уточненного состояния. Команда УТОЧНИТЬ СОСТОЯНИЕ позволяет получить подробную информацию о сбое или ошибке, которые возникли при выполнении последней операции ввода-вывода. Обычно информация уточненного состояния находится в первых двух байтах, а в последующих, количество которых обычно не ограничивается, располагается диагностическая информация. Общее количество информации, выдаваемой в канал по команде УТОЧНИТЬ СОСТОЯНИЕ, зависит от типа внешнего

устройства и определяется его особенностями. Устройство, работающее в мультиплексном режиме, например, должно передавать всю информацию уточненного состояния в течение 32 мкс.

Характер информации уточненного состояния определяется типом ВУ. Однако для большинства ВУ указатели, размещаемые в первых шести разрядах нулевого, т. е. начального, байта уточненного состояния одинаковы. Нулевому разряду отвечает указатель КОМАНДА ОТВЕРГНУТА; далее располагаются следующие указатели: ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО; ОШИБКА НА ШИН-К, СБОЙ В ОБОРУДОВАНИИ, ОШИБКА В ДАННЫХ, ПЕРЕПОЛНЕНИЕ.

Указатель КОМАНДА ОТВЕРГНУТА формируется в следующих случаях:

— ВУ обнаружило программную ошибку, т. е. полученная команда относится к числу запрещенных для данного ВУ;

— ВУ находится в состоянии, исключающем возможность выполнения команды;

— программа предназначена для оборудования, которое в данный момент времени не включено в работу;

— программа вызвала передачу неправильной управляющей информации;

— обнаружена неправильная последовательность команд.

Указатель ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО формируется в случаях, когда ВУ может выполнить последнюю заданную операцию только после вмешательства оператора для восстановления работоспособности ВУ.

Указатель ОШИБКА НА ШИН-К формируется тогда, когда ВУ или УВУ получает по линиям ШИН-К байт данных или байт команды с неправильным контрольным разрядом вследствие ошибки при передаче.

Указатель СБОЙ В ОБОРУДОВАНИИ формируется в тех случаях, когда ВУ или УВУ в процессе выполнения операции устанавливают наличие сбоя в логических схемах, включенных между интерфейсом ввода-вывода и носителем информации. По этому указателю прекращается передача данных, т. е. осуществляется преждевременное завершение операции ввода-вывода.

Указатель ОШИБКА В ДАННЫХ формируется при обнаружении внешним устройством или УВУ ошибки в данных, отличающейся от ошибки на ШИН-К. Обычно этот указатель отвечает ошибкам, связанным с носителем информации.

Указатель ПЕРЕПОЛНЕНИЕ формируется в тех случаях, когда канал не в состоянии вовремя обслужить запрос внешнего устройства. Этот указатель формируется также тогда, когда при выполнении цепочки команд ВУ получает новую команду с недопустимым запаздыванием. При наличии указателя ПЕРЕПОЛНЕНИЕ передача данных, как правило, прекращается.

О режимах работы УВУ. По отношению к интерфейсу подсоединенное к нему УВУ может находиться в режиме «Работа»

или «Автономная работа». В режиме «Работа» УВУ может подключаться к интерфейсу для выполнения операций ввода-вывода. В режиме «Автономная работа» УВУ логически отключено от интерфейса, но обеспечивает ретрансляцию сигнала ВВР-К через логические цепи схемы выборки.

Режим «Автономная работа» устанавливается оператором вручную с помощью переключателя РАБОТА — АВТОНОМНАЯ РАБОТА. С помощью переключателя ДОСТУПЕН — НЕДОСТУПЕН также возможно логическое отключение УВУ от интерфейса, но без перевода его в режим «Автономная работа»; этот переключатель позволяет, например, отключить разделенное УВУ от одного из интерфейсов с сохранением для другого интерфейса состояния УВУ ДОСТУПЕН.

Переключение УВУ из режима «Работа» в режим «Автономная работа» или из состояния ДОСТУПЕН в состояние НЕДОСТУПЕН должно производиться так, чтобы исключить возможность сбоя ЭВМ вследствие действий оператора, т. е. после завершения данным УВУ всех последовательностей сигналов и передачи им в канал необходимой информации состояния.

10.5. ИНТЕРФЕЙС УВУ — ВЗУ

В машинах Единой системы интерфейс УВУ—ВЗУ обеспечивает стандартизацию подключения групп однотипных внешних запоминающих устройств к соответствующим устройствам управления и унификацию наборов сигналов, передаваемых между УВУ и собственно ВЗУ в процессе работы. Практически каждому типу внешних ЗУ отвечает отдельный интерфейс УВУ—ВЗУ, что объясняется спецификой построения, работы и использования различных накопителей. Основными считаются: интерфейс УВУ — накопители на магнитных лентах и интерфейс УВУ — накопители на магнитных дисках; кроме того, используется интерфейс УВУ — накопители на магнитных барабанах. Интерфейс УВУ — накопители на магнитных лентах (НМЛ) имеет две модификации, которые отвечают использованию в НМЛ двух способов записи: а) без возвращения к нулю (НВН-1) и б) фазокодированного.

Любой из указанных интерфейсов обеспечивает физическое подключение к соответствующему УВУ до 8 внешних ЗУ при логическом подсоединении к УВУ в любой момент времени только одного ВЗУ из числа подключенных. Кроме того, обеспечивается стандартизация выполнения операций в ВЗУ, включаемых в состав различных моделей ЕС ЭВМ, а также унификация последовательностей управляющих сигналов при работе с однотипными ВЗУ.

Интерфейс УВУ — накопители на магнитных лентах. Этот интерфейс обеспечивает подключение к одному устройству управления до 8 накопителей на магнитных лентах. При использовании

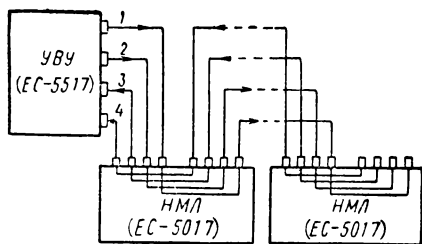


Рис. 10.9. Схема подключения накопителей на магнитных лентах к УВУ

1 — линии передачи управляющей информации;
 2 — линии передачи информации от УВУ к НМЛ (шины записи);
 3 — линии передачи сигналов о состоянии НМЛ;
 4 — линии передачи информации от НМЛ к УВУ (шины считывания, или шины воспроизведения).

Всего используется 55 линий, из которых по 28 информация передается от УВУ к НМЛ и по 27 — от НМЛ к УВУ.

Передача данных как от УВУ к НМЛ, так и от НМЛ к УВУ осуществляется побайтно. Поэтому вторая и четвертая группы линий включают в свой состав по 9 шин; по восьми шинам передаются значения информационных разрядов, а по девятой — значение контрольного разряда. В байте данных, включая контрольный разряд, всегда должно быть нечетное число единиц.

Линии передачи управляющей информации служат для передачи сигналов, непосредственно управляющих ходом выполнения операций записи и воспроизведения, сигналов, управляющих движением ленты, и сигналов выборки НМЛ.

Логическое подключение НМЛ к устройству управления осуществляется при посылке в накопитель сигнала выборки по соответствующей линии. Таких линий 8 по числу НМЛ. Если, например, УВУ расшифровало полученный из канала адрес, как адрес НМЛ-6, то формируется сигнал выборки ВБР-6, посылаемой по линии, имеющей то же обозначение; линии выборки обозначаются как ВБР-0, ВБР-1, ВБР-2, ВБР-3, ВБР-4, ВБР-5, ВБР-6, ВБР-7.

Для организации выполнения в НМЛ операций записи и воспроизведения используется в общем случае 6 сигналов, передаваемых по отдельным линиям:

- УСЗ — установить состояние записи;
- ИСЗ — импульс сопровождения записи;
- СТЗ — сброс триггеров записи;
- ЭСЗ — эхо-сигнал записи;
- УСВ — установить состояние воспроизведения;
- НЗЗ — нет защиты записи.

Сигнал УСЗ используется для установления в накопителе режима выполнения операции записи. Длительность этого сигнала соответствует времени записи в НМЛ, если одновременно действует

способа записи без возвращения к нулю в качестве накопителей используются, например, устройства ЕС-5017, а в качестве устройства управления — ЕС-5517. Передача сигналов осуществляется по функционально разделенным линиям, проходящим последовательно через все НМЛ, как это показано на рис. 10.9. Все линии разделяются на четыре группы:

сигнал НЗЗ. Если сигнал НЗЗ отсутствует, то режим записи не устанавливается. Накопитель находится в состоянии записи до тех пор, пока не будет возбужден сигнал УСВ или один из сигналов управления движением ленты (за исключением сигнала ДВИЖЕНИЕ).

Каждый байт данных при записи сопровождается сигналом ИСЗ, передаваемым по соответствующей линии. Этот сигнал обеспечивает прохождение данных, а также их выравнивание. Для обеспечения записи в конце каждой зоны строки продольного контроля используется сигнал СТЗ, по которому сбрасываются в состояние нуля триггеры записи в НМЛ.

Сигнал ЭСЗ передается из НМЛ в УВУ при записи каждого байта. Он формируется, когда хотя бы в одной головке записи произошло изменение направления тока. Отсутствие этого сигнала указывает на неисправность в цепях записи НМЛ.

Сигнал УСВ используется для установления в накопителе режима выполнения операции воспроизведения. Длительность этого сигнала соответствует времени считывания (воспроизведения) данных в НМЛ и передачи их в УВУ. Состояние воспроизведения сохраняется в НМЛ до тех пор, пока не будет возбуждена линия УСЗ.

Для управления движением магнитной ленты в накопителе используются следующие сигналы:

ДВ — движение;

НЗД — назад;

ПРМ — перемотать;

ПИР — перемотать и разгрузить.

Сигнал ДВ вызывает движение ленты в НМЛ на время своего действия. Если одновременно с этим сигналом действует сигнал НЗД, то движение ленты осуществляется в обратном направлении; если сигнал НЗД не действует, то лента перемещается в прямом направлении. Сигнал ПРМ вызывает движение ленты в обратном направлении до маркера НЛ (начало ленты); операция перемотки осуществляется без участия УВУ. Сигнал ПИР вызывает в НМЛ полную перемотку ленты на подающую катушку.

Линии передачи сигналов о состоянии НМЛ служат для передачи информации состояния, определяющей различные ситуации, возникающие в накопителе при выполнении операций; всего таких линий 17. По этим линиям передаются следующие сигналы:

ВГТ-1, ВГТ-2, ВГТ-3 — сигналы выборки и готовности;

НГТ-0, НГТ-1, . . . , НГТ-7 — сигналы неготовности накопителей с номерами 0, 1, . . . , 7;

НЛ — начало магнитной ленты;

НКЛ — не конец магнитной ленты;

СВ — состояние воспроизведения;

СНП — состояние низкой плотности;

СДВ — состояние движения;

СДВН — состояние движения назад.

Сигналы ВГТ-1, ВГТ-2 и ВГТ-3 определяются скоростью обмена данными. Любой из них, принятый УВУ, свидетельствует о том, что произошла выборка адресуемого накопителя, который готов к выполнению заданной операции.

Сигнал НГТ от адресуемого накопителя свидетельствует о его неготовности (требуется вмешательство оператора) или о выполнении им действий по перемотке ленты в соответствии с управляющими сигналами ПРМ или ПИР. Различные сочетания сигналов ВГТ и НГТ дают подробную информацию о состоянии НМЛ:

Состояние НМЛ	Сигналы	
	ВГТ	НГТ
Выключено питание	}	0 0
Не подключен к УВУ		
Нет ответа на сигнал выборки НМЛ		
Не готов к выполнению операции	}	0 1
Не переведен на работу с УВУ		
Выполняет действия по сигналу ПИР		
Требуется вмешательство оператора		
Выбран и готов выполнять любую операцию	}	1 0
Выбран и готов, но выполняет действия по сигналу ПРМ; после их выполнения НМЛ будет готов для работы с УВУ	}	1 1

Сигнал НЛ передается из накопителя в УВУ при прохождении под магнитными головками маркера «Начало ленты»; этот сигнал формируется при выполнении всех операций, за исключением операции по перемотке и разгрузке, вызываемой сигналом ПИР.

Сигнал НКЛ поступает в УВУ из выбранного накопителя непрерывно, пока под магнитные головки не подойдет маркер «Конец ленты».

Сигнал СВ, выдаваемый в УВУ, свидетельствует о том, что накопитель находится в состоянии воспроизведения. Отсутствие этого сигнала свидетельствует о состоянии записи.

Сигнал СНП, поступающий в УВУ из накопителя, свидетельствует о том, что операция записи или воспроизведения должна производиться при низкой плотности размещения информации, равной 8 бит/мм. Если этот сигнал не выдается в УВУ, то операции производятся при высокой плотности размещения информации, равной 32 бит/мм.

Сигнал СДВ передается непрерывно из накопителя в УВУ при движении ленты; сброс сигнала производится только тогда, когда происходит полный останов ленты. Этот сигнал блокирует в УВУ формирование очередного управляющего сигнала ДВ, который инициирует движение ленты в накопителе.

Сигнал СДВН, поступающий в УВУ из накопителя, свидетельствует о том, что при выполнении предыдущей или текущей операции было использовано движение магнитной ленты в обратном направлении. Считается, что НМЛ находится в состоянии движения

назад до тех пор, пока из УВУ не поступит сигнал УСЗ (УСВ). Сигнал СДВН не формируется при действиях по перемотке ленты, вызываемых сигналами ПРМ и ПИР.

Взаимодействие сигналов при операции записи иллюстрируется временной диаграммой, приведенной на рис. 10.10. Записываемая информация на этом рисунке показана только для одного разряда, отвечающего одной из шин записи ШЗ. При этом последняя информационная строка (последний байт данных) обозначена как ПИС; СПК обозначает запись строки продольного контроля. Время выполнения записи $T_{оп}$ отсчитывается от начала действия сигнала УСЗ (установить состояние записи) до окончания действия сигнала СТЗ, по которому производится сброс триггеров записи.

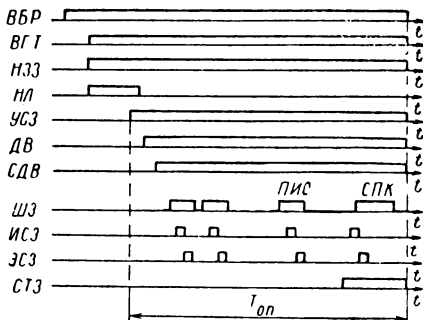


Рис. 10.10. Временная диаграмма выполнения операции записи

Кроме рассмотренных в интерфейсе УВУ — накопители на магнитных лентах используются *сигналы режимов*:

- ВСВ — включить счетчик времени;
- УНП — установить низкую плотность.

По сигналу ВСВ, поступающему из УВУ в накопитель, включается счетчик времени, фиксирующий время работы НМЛ. Счетчик времени НМЛ прекращает работу при выполнении одного из следующих условий:

- прекращение действия сигнала ВСВ;
- под магнитными головками находится маркер НЛ («Начало ленты»);
- НМЛ разгружен (лента перемотана на подающую катушку).

Сигнал УНП, поступая в накопитель, переводит его в состояние низкой плотности, если под магнитными головками находится маркер НЛ. Этот сигнал свидетельствует о том, что последующая операция (запись или воспроизведение) будет выполняться при низкой плотности размещения информации.

Интерфейс УВУ — накопители на магнитных дисках. Этот интерфейс обеспечивает подключение к одному устройству управления ЕС-5551 до 8 накопителей на магнитных дисках ЕС-5050 ЕС-5051, ЕС-5056 в различных комбинациях. Передача сигналов осуществляется по функционально разделенным линиям. Используются селекторные линии, по которым осуществляется связь УВУ только с одним вполне определенным накопителем, и шинные линии, которые проходят через все накопители, но по которым осуществляется связь только с выбранным накопителем, т. е. с логически подсоединенным к УВУ. Наборы сигналов в рассматриваемом интерфейсе отвечают особенностям накопителей,

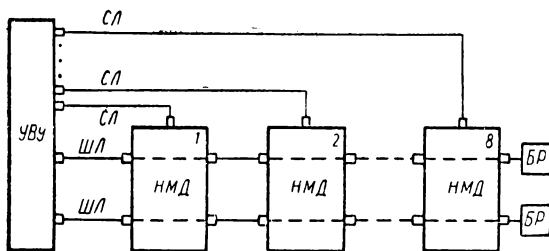


Рис. 10.11. Схема подключения накопителей на магнитных дисках к УВУ

для выполнения операций записи или считывания в которых необходимо выбрать не только накопитель, но и соответствующий цилиндр, а также вполне определенную магнитную головку. При этом под цилиндром понимается совокупность одноименных кольцевых дорожек всех магнитных дисков одного пакета. Перед началом операции возможно перемещение магнитной головки на необходимую дорожку диска.

Общая схема подключения накопителей на магнитных дисках (НМД) к УВУ приведена на рис. 10.11. Обозначения для этой схемы: СЛ — селекторные линии; ШЛ — шинные линии, БР — блок резисторов, используемый для согласования линий.

Селекторные линии служат для передачи следующей информации:

- СЛ1 — сигнал «ВЫБОР НАКОПИТЕЛЯ»;
- СЛ2 — данных для записи на диски;
- СЛ3 — сигналов «НАКОПИТЕЛЬ ПОДКЛЮЧЕН»;
- СЛ4 — сигналов «ВНИМАНИЕ»;
- СЛ5 — данных, считываемых с дисков.

Шинные линии делятся на несколько функциональных групп в соответствии с передаваемой управляющей и служебной информацией:

МШМ — группа из восьми шин М0—М7, по которым из устройства управления передаются сигналы управления выбранным накопителем;

МШР — группа из четырех шин идентификации И0—И3, по которым из УВУ передаются сигналы, сопровождающие и определяющие информацию, передаваемую по МШМ;

МШОС — группа из девяти шин, по которым из накопителя в УВУ передается информация состояния;

МЛА — группа из восьми шин Ц0—Ц7, по которым из накопителя передается адрес цилиндра, указывающий местонахождение головок в момент опроса;

МШ — группа из четырех шин, по которым из УВУ в накопитель передаются дополнительные управляющие сигналы о разрешении отключения или включения системы питания и привода пакета дисков.

Сигналы на шинах группы МШР означают:

И0 — управление;

- И1 — ввод цилиндра;
- И2 — ввод головки;
- И3 — ввод разницы (по номерам цилиндров).

Сигналы на шинах группы МШМ идентифицируются в соответствии с наличием сигналов на шинах группы МШР, как это показано в табл. 10.3.

Т а б л и ц а 10.3

Условия идентификации сигналов на линиях М0÷М7

Шины группы МШМ	Идентификация сигналов на МШМ в зависимости от наличия сигналов на шинах группы МШР			
	И0	И1	И2	И3
М0	ЗАПИСЬ	Цилиндр 128	Головка 128	Разница 128
М1	ЧТЕНИЕ (СЧИТЫВАНИЕ)	» 64	» 64	» 64
М2	НАЧАЛО ПОИСКА	» 32	» 32	» 32
М3	СБРОС РЕГИСТРА АДРЕСА ГОЛОВКИ	» 16	» 16	» 16
М4	СТИРАНИЕ	» 8	» 8	» 8
М5	ВЫБОР ГОЛОВКИ	» 4	» 4	» 4
М6	ВОЗВРАТ К ЦИЛИНДРУ 000	» 2	» 2	» 2
М7	ГОЛОВКУ ПРОДВИНУТЬ	» 1	» 1	» 1

Для группы шин МШОС сигналы в соответствии с их номерами идентифицируются так:

- 0 — ТОК ЗАПИСИ;
- 1 — ГОТОВ;
- 2 — НАКОПИТЕЛЬ ПОДКЛЮЧЕН;
- 3 — НЕИСПРАВНОСТЬ;
- 4 — ОШИБКА ДОСТУПА;
- 5 — КОНЕЦ ЦИЛИНДРА;
- 6 — ИНДЕКС;
- 7 — ГОЛОВКИ ОТВЕДЕНА;
- 8 — ТОЛЬКО СЧИТЫВАНИЕ.

Для сигналов, передаваемых по шинам МЛА, имеет место следующее соответствие: сигнал на шине Ц0 определяет цилиндр 128, на шине Ц1 — цилиндр 64 и т. д.

Выбор накопителя. Для выполнения любой операции УВУ выдает в адресуемый накопитель сигнал выбора по селекторной линии СЛ1. Если накопитель исправен, то он отвечает сигналам по селекторной линии СЛ3 и подсоединяется логически ко всем шинам интерфейса.

Операция установки на заданную дорожку. При выполнении этой операции УВУ посылает сигнал в накопитель по селекторной линии СЛ1. Накопитель отвечает сигналом по МШОС-2 и информацией по одной из шин МЛА, указывающей, на каком цилиндре находятся головки. Определив состояние накопителя, УВУ по

шинам МШМ в сопровождении сигнала идентификации по МШР-И1 посылает в накопитель адрес (номер) требуемого цилиндра; затем, если необходимо, аналогичным образом посылается разница между адресами прежнего и требуемого цилиндров. После этого в накопитель передается адрес головки и направление ее перемещения. Вся информация, получаемая накопителем по шинам группы МШМ, запоминается в соответствующих регистрах.

Для инициирования действий по выполнению собственно операции установки головки на заданную дорожку из УВУ в накопитель передаются сигналы «НАЧАЛО ПОИСКА» и «УПРАВЛЕНИЕ». После этого УВУ может отключиться от накопителя. Завершив поиск требуемого цилиндра, т. е. переместив на него головки, накопитель выдает в УВУ сигналы «ВНИМАНИЕ» и «ГОТОВ».

Передача данных. Для выполнения операции записи информации в накопитель используются сигналы, передаваемые по шинам М0 («ЗАПИСЬ»), М4 («СТИРАНИЕ»), М5 («ВЫБОР ГОЛОВКИ») и сопровождаемые сигналом «УПРАВЛЕНИЕ» по шине И0. Собственно передача данных из УВУ в накопитель по селекторной линии СЛ2 начинается после получения из накопителя ответных сигналов состояния по МШОС-1, 2, 6 и 8.

При считывании информации из УВУ в накопитель посылаются сигналы по шинам М1 и М5 с сопровождением их сигналом по шине И0. Накопитель, в случае возможности выполнения операции считывания, отвечает на эти сигналы сигналами состояния по МШОС-1, 2, 6 и передает по селекторной линии СЛ5 в УВУ данные, считываемые с выбранной дорожки пакета дисков, установленного в накопителе.

Продвижение головок При последовательном выборе головок от нулевой до девятой используется сигнал «ГОЛОВКУ ПРОДВИНУТЬ», передаваемый по шине М7 в сопровождении сигнала по шине И0. Поступление этого сигнала в накопитель увеличивает содержимое регистра адреса головок на единицу, определяя тем самым номер включаемой в работу магнитной головки.

Информация состояния. В случаях программных ошибок и аварийных ситуаций накопитель посылает в УВУ сигналы по определенным шинам группы МШОС. Так, передача сигнала по МШОС-3 свидетельствует об аварийной ситуации в накопителе, по МШОС-4 — о том, что за заданное время в накопителе не осуществлено заданное перемещение головок. Если сигнал передается по МШОС-5, то это означает, что адрес головки в регистре адреса головки превышает число 9 для накопителей со сменными пакетами дисков или число 192 для ЕС-5051.

ВНЕШНИЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЕС ЭВМ

11.1. СОСТАВ И ХАРАКТЕРИСТИКИ ВЗУ

В состав ВЗУ ЕС ЭВМ входят [19, 36]:

- накопители на магнитных лентах ЕС-5010, ЕС-5017, ЕС-5014;
- устройства управления накопителями на магнитных лентах ЕС-5511; ЕС-5517, ЕС-5514;
- накопители на магнитных дисках ЕС-5050, ЕС-5056, ЕС-5051;
- устройство управления накопителями на магнитных дисках ЕС-5551;
- накопитель на магнитном барабане ЕС-5033;
- устройство управления накопителем на магнитном барабане ЕС-5533;
- накопитель на магнитных картах ЕС-5071.

Накопители на магнитных лентах предназначены для хранения больших массивов информации. Они используются при обмене информацией между разными моделями ЕС ЭВМ. Оперативное хранение достаточно больших массивов информации осуществляется главным образом с помощью НМД и частично с помощью НМБ. Накопители на магнитных картах, отличающиеся невысокими характеристиками по времени выборки, предназначены для хранения очень больших массивов информации, обращение к которой производится довольно редко.

Устройства управления накопителями, подключаемые к каналам ввода-вывода моделей ЕС ЭВМ, осуществляют управление соответствующими ВЗУ:

- устройство ЕС-5511 — накопителем ЕС-5010;
- устройство ЕС-5517 — накопителем ЕС-5017;
- устройство ЕС-5514 — накопителем ЕС-5014;
- устройство ЕС-5551 — накопителями ЕС-5050, ЕС-5051, ЕС-5056, ЕС-5071;
- устройство ЕС-5533 — накопителем ЕС-5033.

Основные технические характеристики накопителей и устройств управления накопителями приведены в табл. 11.1, 11.2, и 11.3.

Таблица 11.1

Характеристики накопителей на магнитных лентах

Характеристики	Шифр накопителя		
	ЕС-5010	ЕС-5017	ЕС-5014
Емкость накопителя, Мбайт	20		40
Плотность записи, бит/мм	8 и 32		63
Скорость записи-считывания, кбайт/с	64		126
Число каналов записи		9	
Номинальная скорость движения ленты, м/с		2	
Время разгона ленты, мс	5,5		5
Время останова ленты, мс	3,5		5
Время перемотки катушки, с	—		150
Время смены катушки, с	—		30
Питание от трехфазной сети		380/220 В, 50 Гц	
Потребляемая мощность, кВт	2		2,2
Масса, кг	500		450
Габаритные размеры, мм:			
ширина	1786		1750
глубина	705		790
высота	745		750
Условия эксплуатации:			
температура, °С		+5 — +40	
влажность при 30° С	65±15%	До 95%	
атмосферное давление, мм рт. ст.	760±30	От 460 до 790	
запыленность воздуха, при раз- мере частиц не более 3 мкм, мг/м ³			1

Таблица 11.2

Характеристики устройств управления накопителями на магнитных лентах, дисках, барабанах

Характеристики	Шифр устройства				
	ЕС-5511	ЕС-5517	ЕС-5514	ЕС-5533	ЕС-5551
Число подключаемых накопителей			8		
Скорость передачи данных, кбайт/с	64	128		1200	156
Число каналов обмена информацией		2		1	2
Способ обмена информацией		Параллельный побайтовый			Последовательный
Первичное питание от трехфазной сети		220/380 В, 50±1 Гц			
Потребляемая мощность, кВт	1,0		1,5	2,3	2,0
Габаритные размеры, мм:					
ширина			1600		
глубина			1200		
высота			750		

Таблица 11.3

Характеристики накопителей на магнитном барабане ЕС-5033, на магнитных дисках ЕС-5050, ЕС-5051, ЕС-5056, на магнитных картах ЕС-5071

Характеристики	Шифр накопителя				
	ЕС-5033	ЕС-5050	ЕС-5051	ЕС-5056	ЕС-5071
Емкость, Мбайт	5,6	7,25	100	7,25	125
Плотность записи, бит/мм	42	29—44	29—42	29—44	32
Скорость записи-считывания, кбайт/с	1200	156	83,3	156	52
Число каналов передачи информации	8 + 1		1		8 + 2
Среднее время доступа к информации, мс	21	90		90	5000
Способ обмена информацией	Параллельный		Последовательный		Параллельно-последовательный
Характеристика носителя информации	Барабан с магнитным покрытием, число дорожек 800	В сменном пакете 6 дисков, число рабочих поверхностей в пакете 10. Число дорожек на каждой поверхности 200 + 3	Диски стационарные, число дисков 32 + 1 + 1, на каждой поверхности 3 зоны, в зоне 128 дорожек 3,0	То же, что и для ЕС-5050	Магнитные карты, емкость одной карты 2048 байт, на карте 16 + 4 дорожек, в кассете 128 карт, число кассет 512
Потребляемая мощность, кВт	1,1	0,8		1,5	4,0
Габаритные размеры, мм:					
ширина	1200	800	1200	700	—
глубина	750	700	950	900	—
высота	1600	1000	1630	1100	—

11.2. НАКОПИТЕЛИ НА МАГНИТНЫХ ЛЕНТАХ

Накопитель ЕС-5010 используется в качестве ВЗУ в составе моделей ЕС-1020, ЕС-1030, ЕС-1050. Он работает с устройством управления ЕС-5511 и подключается к нему через интерфейс.

В состав ЕС-5010 входят пульт управления, лентопротяжный механизм, блок магнитных головок, блок управления моторами, блок управления клапанами, вакуумные колонки, панель электроники, распределительная панель, панели разъемов, блоки питания и блок управления питанием.

Перемещение магнитной ленты относительно неподвижного блока магнитных головок обеспечивается с помощью двух следящих систем привода катушек. Время разгона, останова и реверса магнитной ленты выдерживается в заданных пределах за счет буферного устройства, представляющего собой вакуумные колонки с запасом лент для устранения влияния инерционности привода катушек.

Запись (при движении ленты справа налево) и воспроизведение (при движении ленты в обоих направлениях) информации осуществляется блоком магнитных головок, включающим девять головок записи и девять головок воспроизведения. Одновременно с записью производится стирание ранее записанной на ленте информации.

На панели НМЛ собраны основные электронные узлы, обеспечивающие запись и воспроизведение информации, управление следящими системами привода катушек, управление загрузкой ленты, обнаружение маркеров начала и конца ленты, контроль аварийного состояния накопителя, прием команд из устройства управления НМЛ и выдачу команд в это устройство.

Для питания цепей накопителя имеются три стабилизированных источника питания —20 В, —6,3 В, +10 В и один нестабилизированный источник питания —27 В.

Пульт управления предназначен для управления накопителем при подготовке его к работе и в автономном режиме.

Накопитель ЕС-5017 предназначен для работы в качестве ВЗУ моделей ЕС-1020, ЕС-1030, ЕС-1050. Он имеет в своем составе пульт управления, панель управления, панель записи-воспроизведения, плату катушек, счетчик моточасов, устройство ведения ленты, блок магнитных головок, блок силовой катушек, блок силовой моноролика, блок трансформаторов, вентилятор наддува, насос вихревого типа, фильтр сетевой, унифицированные блоки питания, блок управления питанием.

Движение магнитной ленты относительно блока магнитных головок осуществляется с помощью двигателя моноролика и двух электродвигателей, управляемых двумя следящими системами привода катушек. Следящая система привода моноролика обеспечивает постоянство скорости моноролика. Устранение влияния инерционности катушек при пуске и останове магнитной ленты осуществляется, как и в накопителе ЕС-5010, за счет вакуумных

колонок. Запись и воспроизведение информации производится двоянным блоком магнитных головок (в блоке 9 головок записи и 9 головок воспроизведения), к которому крепится стирающая головка, перекрывающая всю ширину ленты. Стирание ранее записанной информации осуществляется при записи на ленту.

Накопитель ЕС-5014 используется в качестве ВЗУ в составе модели ЕС-1050. Основные части накопителя: пульт управления, устройство ведения ленты, оборудование управления и автоматики, блок записи, блок воспроизведения, сервосистема привода катушек, узел магнитных головок, схема выбора НМЛ, схема обнаружения маркеров конца и начала магнитной ленты, механизм защиты записи, оборудование электропитания.

Постоянство скорости перемещения ленты относительно блока магнитных головок обеспечивается устройством ведения ленты. Оборудование управления и автоматики предназначено для реализации команд, поступающих от устройства управления НМЛ в режиме дистанционного управления или с пульта управления в режиме местного управления, а также для выработки аварийных сигналов.

Блок записи служит для приема информации из устройства управления НМЛ и записи ее на магнитную ленту с помощью магнитных головок. Блок воспроизведения осуществляет усиление и обработку сигналов, поступающих с блока магнитных головок при считывании.

В состав узла магнитных головок входят блок магнитных головок (БМГ) и механизм поворота, осуществляющий отвод БМГ от ленты в режимах загрузки, разгрузки, перемотки и в аварийных случаях. Блок магнитных головок — это девятиканальный двоянный блок головок записи и головок воспроизведения, в котором имеется одна головка стирания.

Назначение схемы выбора НМЛ заключается в переключении номера накопителя по желанию оператора. Каждому накопителю может быть присвоен номер от 0 до 7 без переключения кабелей связи между устройством управления и накопителем.

Схема обнаружения маркеров начала и конца магнитной ленты обеспечивает выработку соответствующих сигналов и выдачу их в оборудование управления накопителя в моменты прохождения маркеров начала и конца ленты около датчиков начала и конца ленты.

Механизм защиты записи обеспечивает защиту записанной на ленте информации от случайной записи путем отключения питающего напряжения от усилителей записи, если отсутствует специальный сигнал разрешения записи. Предусматривается также защита записанной информации при включении и выключении электропитания и при аварийном отключении накопителя.

В состав оборудования электропитания входят:

— унифицированные блоки питания с питающими напряжениями 5; 6,3; 12,6; 20 и —12,6 В;

— выпрямители, служащие для выработки нестабилизированного напряжения питания, —27, —30 и —60 В;

— трансформаторы, вырабатывающие трехфазное переменное напряжение 80 В;

— блок управления питанием, который осуществляет включение и выключение питания при нажатии соответствующей кнопки на пульте управления накопителя, дистанционное включение питания, защиту от коротких замыканий и пропадания вторичного питания, защиту от пропадания фазы входного напряжения и отключение напряжения 20 В от усилителей записи при выключении накопителя.

В конструктивном отношении накопитель ЕС-5014 выполнен аналогично накопителю ЕС-5017.

11.3. УСТРОЙСТВА УПРАВЛЕНИЯ НАКОПИТЕЛЯМИ НА МАГНИТНЫХ ЛЕНТАХ

Устройство управления ЕС-5511 предназначено для подключения накопителей на магнитной ленте ЕС-5010, в которых используется метод записи «невозвращение к нулю», к каналам ввода-вывода моделей ЭВМ ЕС-1020, ЕС-1030, ЕС-1050. Устройство ЕС-5511 может подключаться через интерфейс ввода-вывода как к селекторному каналу, так и к селекторному подканалу мультиплексного канала этих моделей и работает только в монопольном режиме. Оно обеспечивает согласованную работу быстродействующих цепей каналов СК или МК с цепями низкого быстродействия, управляющими накопителями.

Основные функции устройства управления (УУ): прием и выполнение команд, передача информации от НМЛ к каналу ввода-вывода (СК или МК) или в обратном направлении, проверка верности информации, передаваемой в накопитель и от него, выработка информации, описывающей состояние УУ и условия, при которых была окончена последняя операция.

Основные блоки устройства управления ЕС-5511 показаны на структурной схеме (рис. 11.1). Рассмотрим назначение этих блоков.

Блоки интерфейса А и Б, аналогичные по устройству и назначению, представляют собой группы линий связи, соединяющих УУ с каналами СК или МК. В состав этих блоков входят также усилители-приемники и усилители-передатчики.

Блок подключения к интерфейсу (БПИ) осуществляет логическое подключение УУ к одному из интерфейсов: по сигналу «Сопряжение КА» к интерфейсу А, а по сигналу «Сопряжение КБ» к интерфейсу Б. В автономном режиме такое подключение производится с помощью переключателя на панели управления.

Блок связи с каналом (БСК) вырабатывает сигналы, управляющие работой всех остальных блоков устройства управления. Это

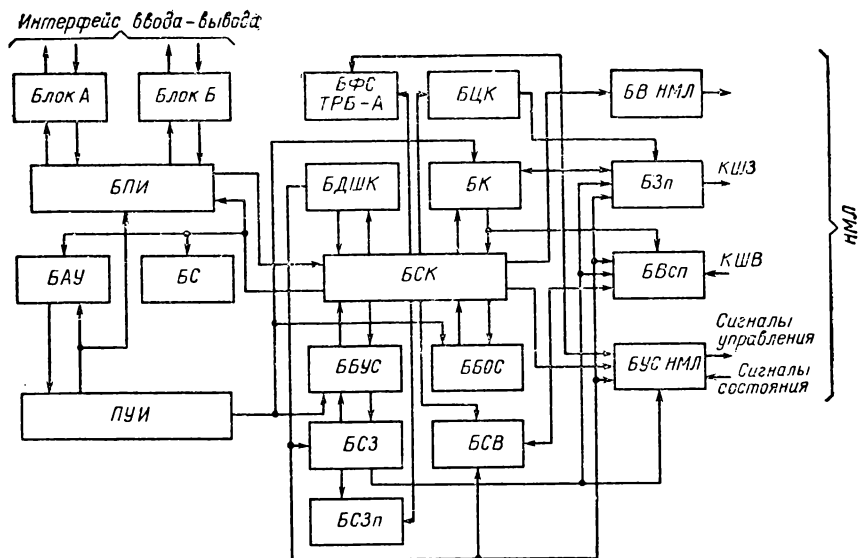


Рис. 11.1. Схема устройства управления ЕС-5511

сигналы, необходимые для установления связи с СК или МК, сигналы синхронизации работы блока в последовательности начальной выработки, сигналы в последовательности ТРБ-А, сигналы для управления занесением номера НМЛ в регистр выбора НМЛ, сигналы для управления занесением информации в регистр информации. В БСК формируется также код байта состояния УУ и НМЛ.

Блок дешифратора команд (БДШК) в зависимости от заданной команды выдает на один из своих выходов сигнал, который разрешает работу цепей, участвующих в выполнении этой команды. Он включает регистр и дешифратор основных команд и регистр и дешифратор вспомогательных команд. Код команды (командный байт) заносится в регистр команд после проверки на четность.

Блок контроля (БК) предназначен для контроля информации, проходящей через УУ, и контроля работы определенных цепей УУ. Он содержит все схемы обнаружения ошибок, кроме схемы циклического контроля.

Блок циклического контроля (БЦК) обеспечивает обнаружение и коррекцию (исправление) ошибок в информации, считанной с одной дорожки магнитной ленты, а также обнаружение ошибок в информации, считанной с нескольких дорожек ленты.

Коррекция ошибок осуществляется в устройстве управления только при плотности записи 32 бит/мм.

При записи информации на магнитную ленту в конце каждой зоны после последнего байта данных записывается строка циклического контроля, необходимая для обнаружения дорожки, на которой произошла ошибка в воспроизводимой информации. В блоке циклического контроля имеется схема, формирующая новую строку циклического контроля во время воспроизведения информации с магнитной ленты. Сравнение считанной и вновь сформированной строк циклического контроля обеспечивает выявление магнитной дорожки, на которой в воспроизведенной информации произошла ошибка.

Блок байта основного состояния (ББОС) предназначен для формирования кода байта основного состояния, который передается в канал (СК или МК) в конце последовательности начальной выборки и по завершении выполняемой команды. В состав блока входят триггеры основного состояния, фиксирующие окончание работы канала, устройства управления и накопителя, а также наличие сбоя в УУ, кроме того, в составе блока имеются цепи включения этих триггеров. Сигналы триггеров основного состояния поступают в регистр информации для последующей передачи в канал.

Блок байтов уточненного состояния (ББУС) служит для выдачи в канал подробных сведений о состоянии УУ и НМЛ, о сбое или ошибке, появившейся при выполнении предыдущей операции ввода-вывода.

Блок формирования сигнала (БФС) ТРБ-А предназначен для выработки сигнала — требования на обслуживание со стороны канала. По этому сигналу осуществляется связь с каналом и передача к нему байта состояния.

Блок счетчика задержек (БСЗ) обеспечивает формирование сигналов задержки пуска и останова НМЛ, сигналов, формирующих и контролируемых межзонные промежутки и расстояния между строками на ленте при записи и при воспроизведении информации, синхроимпульсов, необходимых для синхронизации работы УУ и НМЛ, а также сигналов для своевременного включения цепей записи и воспроизведения.

Блок синхронизатора записи (БСЗп) формирует сигналы, синхронизирующие запись кодов байтов на ленту. За один цикл работы синхронизатора на ленту записывается код одного байта. Основные элементы блока — четырехразрядный счетчик и дешифратор.

Блок синхронизатора воспроизведения (БСВ), включающий трехразрядный счетчик и дешифратор, формирует сигналы, синхронизирующие работу схем УУ при считывании информации с ленты.

Блок выбора (БВ) НМЛ осуществляет логическое подключение выбранного НМЛ к УУ. В его составе имеется регистр адреса НМЛ (адрес устанавливается при подготовке к работе) и дешифратор для дешифрации адреса накопителя.

Блок записи (БЗп) обеспечивает передачу информации по кодовым шинам записи (КШЗ) в НМЛ. Он включает девять триггеров (по числу каналов записи) и схему занесения информации на входе этих триггеров.

Блок воспроизведения (БВсп) предназначен для кратковременного хранения считанной с ленты информации, которая поступает на вход блока по кодовым шинам воспроизведения (КШВ). Он состоит из регистра перекода, девяти одновибраторов, схемы контроля на четность и схемы распознавания кода маркера группы зон.

Блок управления и состояния (БУС) НМЛ вырабатывает сигналы управления и выдает их в выбранный накопитель, а также принимает из накопителя сигналы, характеризующие состояние НМЛ.

Блок автономного управления (БАУ) является имитатором канала, он используется при автономной проверке работы УУ и НМЛ.

Блок сбросов (БС) служит для выработки сигналов, управляющих цепями сбросов триггеров в различных блоках устройства управления. Сброс триггеров производится в различных блоках в разное время: при включении питания, в процессе выполнения команды, в конце выполнения команды. Блок сбросов связан почти со всеми другими блоками УУ (на рис. 11.1 эти связи не показаны).

Пульт управления и индикации (ПУИ) предназначен для управления работой УУ в автономном режиме, а также для визуальной проверки работы УУ и НМЛ.

Устройство управления ЕС-5514 предназначено для осуществления логической и физической связи накопителей на магнитной ленте типа ЕС-5014, в которых используется фазокодированный способ записи информации, с каналами ввода-вывода в составе модели ЕС-1050.

Устройство управления содержит цепи связи с каналом и цепи управления накопителем. Цепи связи с каналом, взаимодействуя через стандартный интерфейс, выполняют все функции по связи УУ с каналом. Цепи управления накопителем формируют сигналы, необходимые для выполнения заданной команды, и передают их в НМЛ по линиям интерфейса УУ—НМЛ.

В УУ хранится каждый байт, посылаемый каналом в НМЛ до тех пор, пока накопитель его не примет. Точно так же, при передаче байта, считанного с ленты или сформированного в самом УУ для передачи в канал, этот байт хранится в устройстве управления до приема его каналом.

Основные функции, выполняемые устройством управления при взаимодействии с каналом, состоят в следующем: прием информации из канала и передача ее в канал, контроль поступившей из канала информации, дешифрация кодов команд, принятых из канала, передача в канал информации о состоянии УУ и НМЛ, накопление и хранение информации о состоянии УУ и НМЛ.

Основные функции УУ по управлению работой НМЛ:

- выбор заданного накопителя;
- управление движением магнитной ленты в накопителе;
- прием информации из НМЛ и передача информации в НМЛ, а также обеспечение контроля этой информации;
- преобразование кодов сообщений в коды, принятые в НМЛ, и обратное преобразование;
- формирование временных задержек, необходимых для правильного выполнения записи-воспроизведения на магнитной ленте;
- контроль амплитуды сигналов на КШВ;
- управление завершением ряда операций (перемотка ленты, разгрузка и др.) без участия канала.

В состав устройства ЕС-5514 входят те же блоки, которые были указаны для устройства ЕС-5511. Функции, выполняемые этими блоками, также аналогичны. Отсутствуют только блок циклического контроля и блок синхронизатора записи. Кроме того, блок воспроизведения отличается как по структуре, так и по выполняемым функциям.

В устройстве ЕС-5514 блок воспроизведения предназначен для усиления, формирования, декодирования сигналов, поступающих по КШВ из НМЛ. Блок является девятиканальным, причем в каждом канале имеется усилитель-ограничитель, усилитель выделения огибающей, усилитель-детектор, усилитель импульсов, генератор автоматической подстройки, усилитель сигналов управления, демодулятор. Общими для всех каналов являются цепи выравнивания перекоса ленты. На вход каждого канала блока воспроизведения из НМЛ поступают аналоговые сигналы амплитудой 0,5 В, несущие информацию, закодированную в соответствии с фазокодированным способом записи.

В конструктивном отношении устройства ЕС-5514 и ЕС-5511 полностью аналогичны.

Устройство управления ЕС-5517 предназначено для подключения накопителей на магнитной ленте типа ЕС-5017, ЕС-5012, ЕС-5022, в которых используется способ записи «невозвращение к нулю», к каналам ввода-вывода моделей ЕС-1020, ЕС-1030, ЕС-1050. Устройство работает только в монопольном режиме, оно подключается к селекторному каналу или к селекторному подканалу мультиплексного канала через стандартный интерфейс ввода-вывода. Связь УУ с накопителями осуществляется через интерфейс УУ—НМЛ.

Устройство ЕС-5517 по своей структуре, выполняемым функциям и конструктивному оформлению очень близко к устройству ЕС-5511. Наиболее существенное отличие состоит в том, что устройство ЕС-5517 может управлять работой накопителей на магнитной ленте трех модификаций, отличающихся скоростью движения ленты. Номинальная скорость движения ленты для этих модификаций НМЛ соответственно равна 2, 3 и 4 м/с.

11.4. НАКОПИТЕЛИ НА МАГНИТНЫХ ДИСКАХ

Появление НМД обусловлено необходимостью иметь такую внешнюю память, которая сочетала бы большую емкость с достаточным быстродействием. В настоящее время емкость НМД достигает сотен и тысяч миллионов бит, а скорость записи-считывания — до миллиона бит в секунду и более.

Рассмотрим сначала общие принципы построения НМД. Магнитные диски (рис. 11.2) изготавливаются обычно из алюминия с ферромагнитным покрытием. Диаметр диска от 250 до 650 мм, толщина 2,0—2,5 мм. Толщина ферромагнитного покрытия (носителя информации), наносимого с обеих сторон диска, составляет 1,0—2,0 мкм. Запись информации производится по концентрическому магнитным дорожкам диска. В целях сокращения времени поиска необходимой дорожки каждая сторона диска может разбиваться на кольцевые зоны. На рис. 11.2, б показаны три зоны: внешняя, средняя и внутренняя.

Диски жестко закрепляются на ведущем валу. Число дисков, составляющих пакет, может быть от 4 до нескольких десятков, расстояние между дисками 7—8 мм.

Запись-считывание информации осуществляется магнитными головками 3, укрепленными на подвижных рычагах 4 (рис. 11.2, а). Рычаги жестко закреплены на каретке 5, которая может перемещаться вдоль направляющего вала 6. При движении каретки магнитные головки перемещаются вдоль радиуса дисков в пределах своей зоны (каждая зона диска обслуживается своей головкой).

Диски имеют постоянную частоту вращения, равную 1000—3000 об/мин. Вследствие высокой частоты вращения дисков невозможны как точная механическая установка зазора между головками и дисками, так и контактная запись. Поэтому необходимый зазор (порядка 8—15 мкм) поддерживается при помощи

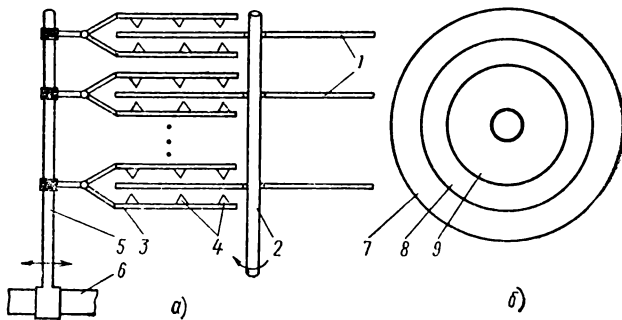


Рис. 11.2. Запоминающее устройство на магнитных дисках:

а — конструкция; б — расположение магнитных дорожек на поверхности диска; 1 — магнитные диски; 2 — ведущий вал; 3 — рычаги с магнитными головками; 4 — магнитные головки; 5 — каретка; 6 — направляющий вал; 7, 8, 9 — зоны магнитного диска

воздушной опоры: капсула головки содержит несколько небольших сопел, через которые подается поток воздуха, создающий воздушную «подушку» между головкой и покрытием диска. Головка подвержена действию усилия, прижимающего ее к диску, а воздушная «подушка» препятствует образованию контакта. В случае прекращения подачи воздуха блок головок автоматически отходит от диска.

Выбор необходимой магнитной дорожки для записи или считывания информации производится по адресу, поступающему от устройства управления НМД. Пусть, например, в НМД имеется 32 диска, поверхность которых разбита на зоны по 3 зоны на каждой стороне диска. В каждой зоне располагаются 64 магнитные дорожки. Тогда адрес необходимой дорожки представляется 14-разрядным двоичным числом: пять разрядов для выборки диска, один разряд для выборки стороны диска (верхней или нижней), два разряда для выборки зоны диска и шесть разрядов для выборки дорожки в зоне.

Запись и считывание информации осуществляется в последовательном коде. Если плотность размещения информации на всех дорожках диска одинакова (эта плотность равна 20—45 бит/мм), то при постоянной скорости вращения дисков частота ввода-вывода информации будет различной для различных зон: максимальная частота будет при обращении к дорожкам внешней зоны и минимальная частота — при обращении к дорожкам внутренней зоны. Можно добиться постоянной скорости ввода-вывода для всех зон диска. Для этого необходимо, чтобы плотность размещения информации увеличивалась при перемещении от внешней зоны к внутренней. В зависимости от размеров дисков, скорости их вращения, плотности записи скорость ввода-вывода информации в ЗУ на магнитных дисках колеблется в пределах 50 000— 1×10^6 бит/с.

Рассмотренная конструкция НМД, когда для каждого диска имеются свои магнитные головки, перемещающиеся только вдоль радиуса диска, не является единственной. В более простой конструкции в блоке магнитных головок НМД имеется всего две головки записи-считывания информации, одна из которых обслуживает верхние стороны всех дисков, а другая — нижние. При выборке требуемого диска и дорожек на нем блок магнитных головок сначала перемещается вертикально вдоль каретки до диска, указанного в адресе, а затем вместе с кареткой — горизонтально до соответствующей пары магнитных дорожек. В этом случае время выборки требуемой дорожки существенно увеличивается и составляет 0,1—0,8 с.

В ЕС ЭВМ в качестве внешних запоминающих устройств используются три типа накопителей на магнитных дисках: ЕС-5050, ЕС-5051, ЕС-5056.

Накопитель на сменных магнитных дисках ЕС-5050 предназначен для работы в качестве ВЗУ в составе моделей ЕС-1020,

ЕС-1030, ЕС-1050. Он работает совместно с устройством управления ЕС-5551 и подключается к УУ через интерфейс УУ—ВЗУ.

Накопитель работает с пакетами сменных магнитных дисков, имеющими следующие характеристики:

- число дисков в пакете — 6;
- число рабочих поверхностей — 10 (две крайние не используются);
- емкость пакета 58 млн. бит;
- число магнитных дорожек на каждой стороне диска — 200 основных и 3 запасных;
- диаметр магнитной дорожки № 000 (внешней) — 330 мм, диаметр дорожки № 202 (внутренней) — 227,5 мм;
- магнитное покрытие дисков — ферролаковое;
- метод записи — двухчастотный без возвращения к нулю;
- продольная (вдоль дорожки) плотность записи: минимальная (для дорожки № 000) — 29 бит/мм, максимальная (для дорожки № 202) — 44 бит/мм. Благодаря тому, что плотность записи информации на магнитных дорожках при движении вдоль радиуса диска от периферии к центру увеличивается, скорость записи-считывания остается постоянной (156 кбайт/с) независимо от номера дорожки, к которой производится обращение.

Рабочая частота вращения дисков 2400 ± 48 об/мин.

В накопителе используются 10 блоков магнитных головок (по одному блоку на каждую рабочую поверхность пакета дисков). В каждом блоке имеется одна универсальная головка записи-воспроизведения и одна стирающая головка. Предусмотрен автоматический вывод блоков магнитных головок за пределы пакета дисков, если нарушен режим питания или частота вращения пакета дисков уменьшилась до 1700 об/мин и менее.

Магнитные дорожки, имеющие один и тот же номер на всех десяти рабочих поверхностях пакета дисков, образуют цилиндр с тем же порядковым номером.

Доступ к требуемой информации обеспечивается в среднем за 90 мс, максимальное время доступа — 150 мс, минимальное — 30 мс.

Накопитель должен работать в закрытых помещениях при температуре окружающей среды от 10 до 35° С и атмосферном давлении от 720 до 790 мм рт. ст.

Первичное питание НМД осуществляется от трехфазной сети переменного тока напряжением 380/220В с отклонениями от номинала на +10% — —15% и частотой 50 ± 1 Гц. Для питания электронного оборудования накопителя блок питания вырабатывает стабилизированные напряжения 5; 6,3; 20; —20 и —40В и нестабилизированные 27 и 40В.

Масса накопителя не превышает 200 кг.

Структурная схема НМД ЕС-5050 приведена на рис. 11.3. Панель электроники (ПЭ) предназначена для размещения на ней типовых элементов замены (ТЭЗ), соединения их между собой

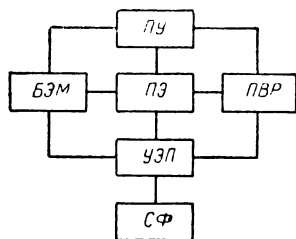


Рис. 11.3. Схема накопителя на магнитных дисках ЕС-5050

и другими блоками и узлами накопителя. В блоке электромеханики (БЭМ) размещены блоки магнитных головок, линейный двигатель, служащий для перемещения каретки с блоками магнитных головок и их установки в заданное положение, асинхронный двигатель для вращения пакета дисков. Панель внешних разъемов (ПВР) предназначена для соединения накопителя с устройством управления при работе в автоматическом режиме и с прибором контроля накопителя при работе в автономном

режиме. Устройство электропитания (УЭП) обеспечивает электронное оборудование накопителя соответствующим напряжением указанных выше потенциалов. Панель управления (ПУ) предназначена для размещения на ней элементов ручного управления и светового табло.

В накопителе ЕС-5050 логические цепи выполнены на интегральных схемах серии 155, а электронное оборудование цепей записи-воспроизведения, перемещения головок и других узлов — на дискретных элементах.

Накопитель на сменных магнитных дисках ЕС-5056 выполняет те же функции, что и накопитель ЕС-5050. Принципиальных различий между этими накопителями не имеется.

Накопитель на стационарных магнитных дисках ЕС-5051 в своем составе имеет блок электроники, блок электромеханики, насосную установку гидропривода, систему питания, вентиляционную систему. В качестве носителя информации используется блок из 32 постоянных дисков. Информация записывается на 64 рабочих поверхностях дисков. Каждая поверхность разбивается на 3 зоны, в каждой зоне имеется 128 магнитных дорожек. Каждая зона обслуживается своей магнитной головкой записи-воспроизведения.

Магнитные дорожки с одинаковым номером, расположенные в одноименных зонах всех рабочих поверхностей дисков, образуют цилиндр. Адрес необходимой магнитной дорожки может быть представлен 15-разрядным двоичным числом: 6 разрядов для выборки рабочей поверхности дисков, 2 разряда для выборки зоны и 7 разрядов для выборки дорожки (цилиндра) в зоне.

Перемещение магнитных головок в заданное положение, т. е. на заданный цилиндр, производится с помощью гидропривода, а выбор головки в данном цилиндре (выбор рабочей поверхности диска и зоны на этой поверхности) осуществляется путем электронной коммутации магнитных головок.

Накопитель ЕС-5051 может работать совместно с устройством управления ЕС-5551 только в режиме «Работа», когда выполняются операции поиска, записи и воспроизведения информации. Вместе

со стендом наладки и проверки накопитель, кроме того, может работать в режиме «Разметка», когда производится запись и воспроизведение (проверка) служебной информации: индексов, определяющих физическое начало магнитной дорожки, синхронизирующих импульсов, служащих для синхронизации при записи рабочей информации и адресов цилиндров, адресных импульсов, определяющих начало адресного сектора, и адресов цилиндров, необходимых для проверки соответствия заданного адреса цилиндра с действительным положением механизма доступа и выработки сигнала при завершении механического перемещения головок записи-воспроизведения в требуемое положение.

11.5. НАКОПИТЕЛЬ НА МАГНИТНЫХ КАРТАХ

Накопитель на магнитных картах ЕС-5071 предназначен для записи, хранения и воспроизведения больших массивов информации (до 10^9 бит). Он работает совместно с устройством управления ЕС-5551 в составе моделей ЕС-1020, ЕС-1030, ЕС-1050.

Конструктивно и схемно накопитель ЕС-5071 разделяется на два устройства: промежуточное устройство управления ЕС-5071.1 и устройство накопления ЕС-5071.2.

Промежуточное устройство управления является связующим звеном между устройством управления ЕС-5551 и устройством накопления ЕС-5071.2. Оно преобразует сигналы интерфейса в сигналы, удобные для использования устройством накопления и предназначенные для управления записью-воспроизведением информации на магнитных картах, кодирования записываемой и декодирования считываемой информации, контроля и коррекции ошибок, преобразования кодов команд.

Устройство накопления служит для хранения магнитных карт, транспортировки их к головкам записи-воспроизведения, обработки карт и передачи считанной информации к устройству ЕС-5071.1 (при воспроизведении) или от этого устройства к магнитным головкам (при записи). К одному промежуточному устройству управления может подключаться от 1 до 8 устройств накопления.

Основные функциональные части промежуточного устройства управления показаны на рис. 11.4.

Блок сопряжения с устройством управления ЕС-5551 (БСУ) выполняет функции:

— физическое подключение устройства ЕС-5071.1 к шинам

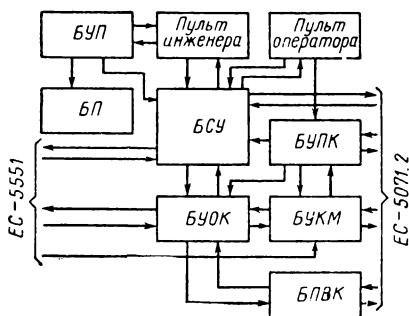


Рис. 11.4. Схема промежуточного устройства управления ЕС-5071.1

«малого» интерфейса устройства ЕС-5551 и согласование уровней сигналов, циркулирующих между ними;

— управление выбором и логическое подключение необходимого устройства накопления к УУ ЕС-5551;

— управление выбором и логическое подключение необходимого устройства накопления к устройству ЕС-5071.1 под воздействием сигналов автономного выбора, поступающих с пульта инженера;

— прием и накопление полного адреса требуемой магнитной карты по сигналам из устройства управления;

— формирование адреса текущей карты по сигналам из блока управления поиском карты и сравнение его с адресом требуемой карты;

— управление началом и окончанием поиска или возврата карты;

— выдачу адреса требуемой ячейки пенала карт;

— выдачу команд обработки карты;

— прием сигналов от устройства ЕС-5071.2, характеризующих его состояние.

Блок управления поиском карты (БУПК) управляет операцией сравнения [она происходит в блоке сопряжения с устройством управления (БСУ)] адресов требуемой и текущей карт и вырабатывает команды поиска секции, ячейки, карты и возврата карты под воздействием сигналов, образующихся в блоке сопряжения с УУ в результате совпадения адресов требуемой и текущей карт.

Блок управления картопротяжным механизмом (БУКМ) воспринимает сигналы от датчиков слежения за положением карты на барабане и команды поиска карты от БУПК и вырабатывает команды управления электромагнитами подачи карты на барабане, возврата и сброса карты, перемещения блока головок.

Блок управления обработкой карты (БУОК) обеспечивает преобразование последовательного кода информации, поступающей от устройства управления, в параллельно-последовательный код информации записи, формирование управляющих сигналов записи-воспроизведения, временного преобразования команд обработки карты (команд считывания, записи, стирания).

Блок преобразования, выравнивания и коррекции (БПВК) служит для дополнения 8-разрядного параллельно-последовательного кода записываемой информации двумя контрольными разрядами и преобразования полученного 10-разрядного кода для записи на карту фазовым методом, выдачи фазового кода записи ко всем подключенным устройствам накопления, преобразования и выравнивания во времени считанной информации, контроля выровненного 10-разрядного кода, коррекции исправимых и обнаружения неисправимых ошибок, выдачи 8-разрядного параллельно-последовательного кода считанной информации.

Пульт инженера предназначен для размещения органов ручного управления и световых индикаторов состояния накопителя и адреса текущей карты. Органы ручного управления обеспечивают включение устройства в автоматический или автономный режим работы, в режим разметки карт или в режим ручного управления, а также включение питания устройства.

На пульте оператора имеется световая сигнализация о готовности, выборе и сбое каждого из восьми подключенных устройств накопления.

Блок управления питанием (БУП) и блок питания (БП) обеспечивают подключение промежуточного устройства управления к источнику напряжения переменного трехфазного тока 380/220 В, снабжение всех блоков устройства постоянным током со стабилизированным напряжением 5, 6 и 12,6 В, включение и выключение вторичных источников питания, выдачу управляющих сигналов для последовательного питания устройств ЕС-5071.2.

В состав устройства накопления ЕС-5071.2 (рис. 11.5) входят блок автоматической картотеки (БАК), блок управления автоматической картотекой (БУАК), блок картопротяжного механизма (БКМ), блок усилителей (БУс), пульт инженера, БП, БУП, блок релейный (БР), установка пневмопитания (УПП). По заданному адресу в устройстве накопления самостоятельно выполняется операция произвольной выборки группы карт и возврат ее в картотеку. Последовательная выборка (поиск карты) и запись-воспроизведение информации осуществляются под управлением устройства ЕС-5071.1.

В блоке автоматической картотеки имеется магазин для хранения 512 групп карт, состоящий из двух секций. В секции содержится 32 пенала. С помощью пневматических дешифраторов (по одному на каждую секцию) производится произвольная выборка пенала: перемещение пенала на позицию, соответствующую адресу ячейки пенала. Извлечение группы карт из выбранного пенала и передача ее в блок картопротяжного механизма осуществляется специальным подъемником. Время, затрачиваемое на произвольную выборку группы карт, составляет 2,5 с. Управление механизмами выборки группы карт выполняется сигналами, которые формируются в блоке управления автоматической картотекой.

Последовательная выборка карт из выбранной группы, а также запись и считывание информации производится в блоке карто-

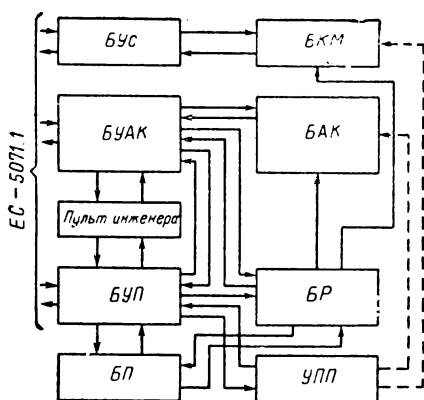


Рис. 11.5. Схема устройства накопления ЕС-5071.2

протяжного механизма, который содержит вакуумный барабан для транспортировки карт с линейной скоростью 4 м/с, механизм приема-выдачи для выдачи карт на барабан с частотой 26,5 карт/с, блок магнитных головок с 20-ю каналами, карман выдачи и карман брака для разбраковки карт.

Для усиления кодовых сигналов в блоке усилителей имеется 10 усилителей записи-воспроизведения. Кроме того, там же размещаются 10 усилителей стирания и усилители емкостных датчиков блока картопротяжного механизма. Включение электродвигателей блоков автоматической картотеки и картопротяжного механизма осуществляется сигналами, которые вырабатываются релейным блоком по командам от блока управления автоматической картотекой.

Установка пневмопитания обеспечивает блоки автоматической картотеки и картопротяжного механизма воздухом пониженного и повышенного давления (см. пунктирные линии на рис. 11.5).

В блоке питания имеются унифицированные стабилизированные источники питания. Управление включением блоков устройства ЕС-5071.2 и контроль питания осуществляется блоком управления питания.

Пульт инженера оборудован органами ручного управления, необходимыми для проведения профилактического контроля устройства накопления. Он используется также при ручной замене группы карт в картотеке.

11.6. УСТРОЙСТВО УПРАВЛЕНИЯ НАКОПИТЕЛЯМИ НА МАГНИТНЫХ ДИСКАХ И КАРТАХ

Устройство управления ЕС-5551 предназначено для управления работой накопителей на магнитных дисках ЕС-5050, ЕС-5056, ЕС-5051 и магнитных картах ЕС-5071 и подключения их к каналам ввода-вывода в составе моделей машин ЕС-1020, ЕС-1030, ЕС-1050. Устройство управления может подключаться к селекторному или мультиплексному каналу или к обоим каналам одновременно. К нему подсоединяются до 8 накопителей на дисках и картах в любом сочетании.

Выполнение команд, поступающих из каналов ввода-вывода, осуществляется с помощью управляющей программы, которая хранится в блоке памяти микрокоманд — долговременном запоминающем устройстве (ДЗУ), имеющемся в составе устройства управления (рис. 11.6). В ДЗУ хранятся также специальные тест-программы, которые используются для проверки работоспособности УУ и подключенных накопителей при работе устройства управления в автономном режиме.

В ходе обмена информацией с каналами и с ВЗУ осуществляется ее контроль: при обмене с каналами — контроль на нечетность, при обмене с ВЗУ — циклический контроль массивов данных.

Рассмотрим назначение основных функциональных частей устройства управления.

Блок центрального управления (БЦУ) служит для формирования и выдачи управляющих и синхронизирующих сигналов во все другие блоки УУ. Выработка управляющих сигналов производится в такой последовательности: формируются адреса команд управляющей программы, размещенной в ДЗУ, и осуществляется их контроль, затем адреса команд выдаются в ДЗУ, далее из ДЗУ принимаются и контролируются выбранные команды, производится дешифровка команд и выработка управляющих сигналов для всех блоков УУ. После начального пуска устройства управления блок центрального управления всегда считывает из ДЗУ команду по нулевому адресу. Адреса последующих команд формируются в соответствии с информацией, содержащейся в предыдущей команде с учетом состояния индикаторов переходов функциональных блоков УУ.

Кроме того, блок центрального управления выполняет следующие функции:

- считывание и прием команд канала, управляющей информации и данных из блока сопряжения с каналами (БСК) и их обработку в соответствии с командами управляющей программы;
- контроль данных и управляющей информации, принятой из блока сопряжения с каналами, и передачу их в блок сопряжения с накопителями (БСН) по командам управляющей программы;

- считывание и прием данных и информации о состоянии накопителей из блока сопряжения с накопителями, их контроль и передачу в блок сопряжения с каналами по командам управляющей программы;

- формирование информации о состоянии УУ и передачу ее в блок сопряжения с каналами;

- останов УУ при обнаружении ошибок в адресах команд, в самих командах и в арифметическо-логическом блоке (АЛБ);

- проверку функционирования блоков устройства управления, накопителей и их взаимодействия по тестовой программе в автономном режиме работы.

Блок памяти микрокоманд (ДЗУ) предназначен для хранения управляющей и тестовых программ УУ. Он обеспечивает считывание команд в соответствии с принятым адресом и передачу их в блок центрального управления.

Арифметическо-логический блок (АЛБ) служит для выполнения логических и арифметических операций с байтами информа-

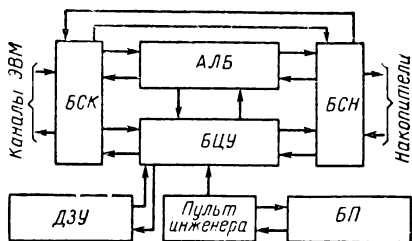


Рис. 11.6. Схема устройства управления ЕС-5551

ции, поступающей из блока сопряжения с каналами, блока сопряжения с накопителями или блока центрального управления. Тип операции определяется управляющими сигналами из блока центрального управления. Результат операции, которому присваивается контрольный код, выдается в один из блоков, связанных с АЛБ. В АЛБ по результатам выполненной операции вырабатываются сигналы для установки индикаторов переходов функциональных блоков УУ, которые (индикаторы) указываются в кодах команд управляющей программы.

Блок сопряжения с каналами (БСК) осуществляет физическое подключение УУ к шинам стандартного сопряжения селекторного и мультиплексного каналов ЭВМ с ВЗУ и УВВ.

Он обеспечивает выполнение следующих функций:

— согласование уровней сигналов стандартного сопряжения каналов с уровнями сигналов УУ;

— прием из каналов ЭВМ и контроль на нечетность адреса УУ и накопителей, подключенных к УУ; сравнение адреса УУ с адресом, принятым из канала; передачу адреса в БЦУ;

— логическое подключение УУ к каналу ЭВМ по сигналам управления из БЦУ; прием из канала и контроль на нечетность команд, служебной информации и данных и передачу их в БЦУ по сигналам БЦУ;

— прием байтов состояния УУ и данных из БЦУ и выдачу их в канал;

— логическое отключение УУ от канала;

— хранение служебной информации о состоянии накопителей, подключенных к УУ, и выдачу ее в АЛБ по сигналам из БЦУ; выдачу в канал байта состояния УУ при обращении канала к устройству управления, занятому в это время выполнением операции.

Блок сопряжения с накопителями (БСН) обеспечивает физическое подключение к УУ до восьми накопителей, а также логическое подсоединение (по сигналам из БЦУ) к УУ одного из этих накопителей.

Основные функции БСН:

— выдача в выбранный накопитель управляющих сигналов и байтов служебной информации;

— считывание байтов данных из БЦУ, контроль их на нечетность, преобразование параллельного кода байтов в последовательный код (биты) и выдача в выбранный накопитель;

— прием данных из логически подключенного накопителя, преобразование их из последовательного кода в последовательно-параллельный (каждый байт — в параллельном коде, а байты, принадлежащие одному слову, формируются последовательно) и выдача в БЦУ;

— прием адреса цилиндра выбранного накопителя на магнитных дисках и адреса магнитной головки в этом накопителе и выдача их в АЛБ по сигналам из БЦУ;

вится в соответствие 5-разрядная группа, которая заносится в выходной сдвигающий регистр блока кодирования и затем выводится в последовательной форме в соответствующий канал записи блока записи (БЗп). В канал записи информация поступает с частотой 1,5 МГц. С выходов канала записи кодовые сигналы 1 и 0 поступают в полуобмотки магнитных головок блока магнитного барабана (БМБ).

Воспроизведение записанной на барабане информации осуществляется по команде «Чтение», поступающей из УУ. Информация, считанная магнитными головками, поступает в восемь каналов воспроизведения блока воспроизведения (БВ), где производится усиление информационных сигналов и устранение временных рассогласований между сигналами разных разрядов. С выхода БВ информация подается в БКд, где осуществляется ее преобразование в двоичный код. После этого воспроизведенная информация через БСН с УУ подается в устройство управления ЕС-5533.

В накопителе имеется блок коммутации (БКм), представляющий собой электронный коммутатор магнитных головок на 100 положений. С помощью электронного коммутатора в соответствии с заданным адресом тракта производится выборка группы (тракта) из восьми магнитных головок, по одной головке на каждый из восьми информационных разрядов записываемого (воспроизводимого) байта.

Блок адресных маркеров (БМА) служит для записи и воспроизведения адресных маркеров АМ-1 и АМ-2. Маркер АМ-1 указывает начало зоны на магнитной дорожке, он выдается в УУ. Адресный маркер АМ-2 записывается перед каждым массивом информации внутри зоны магнитной дорожки и предназначен для приведения в исходное состояние схемы канала воспроизведения перед воспроизведением заданного массива информации.

Служебные сигналы, необходимые для работы всех схем накопителя, формируются в блоке выработки служебных сигналов (БВСС). Блок управления (БУ) обеспечивает включение питания накопителя и, кроме того, с его помощью осуществляется стирание информации соответствующей секцией стирающей головки, а также импульсное стирание рабочей (не служебной) информации по определенному тракту.

К одному УУ может быть подключено до восьми НМБ ЕС-5033. Выбор одного из 8 накопителей осуществляется подачей на него сигнала ВЫБОР НМБ. В ответ на этот сигнал в накопителе формируется сигнал НАКОПИТЕЛЬ РАБОТОСПОСОБЕН, если он готов к работе, или сигнал НЕИСПРАВНО в случае неготовности накопителя. Эти сигналы подаются в УУ только от выбранного накопителя. От него же в УУ подаются синхронизирующие импульсы частотой 1,2 МГц, которые при записи используются в УУ для выдачи записываемой информации.

Устройство управления накопителями на магнитных барабанах ЕС-5533 обеспечивает управление работой НМБ с побайтной записью и подключение накопителей к селекторному каналу модели ЕС-1050. Устройство ЕС-5533 может работать поочередно с четырьмя селекторными каналами.

В состав устройства управления входят (рис. 11.8): блок сопряжения с каналами (БСК), блок управления (БУ), блок контроля информации (БКИ), блок сопряжения с накопителями (БСН), пульт инженера, кроме того, блоки питания и блок управления питанием.

Основные функции блока сопряжения с каналами состоят в следующем:

- физическое подключение УУ к шинам стандартного сопряжения селекторных каналов и согласование уровней сигналов сопряжения каналов с уровнями сигналов УУ, прием сигналов сопряжения из каналов;

- прием адресов устройства управления и подключенных к нему НМБ, контроль байта адреса, принятого из канала, на четность и передача его в БУ;

- логическое подключение УУ к селекторным каналам, прием команд из канала и контроль его на четность, выдача команд в БУ;

- прием из канала байтов служебной информации и байтов данных, контроль их на четность и выдача в другие блоки устройства управления;

- выдача сигналов стандартного сопряжения в каналы, присвоение байтам служебной информации и байтам данных контрольного разряда на четность и выдача их в каналы.

В состав БСК входит блок переключения каналов, который обеспечивает выбор НМБ, резервирование накопителей по командам из каналов, освобождение зарезервированных НМБ по командам освобождения, формирование сигналов ошибок УУ и НМБ, хранение и выдачу информации о состоянии УУ и НМБ.

Блок сопряжения с накопителями предназначен для физического подключения к устройству управления восьми НМБ ЕС-5033 и логического подсоединения УУ к одному из этих накопителей. Он обеспечивает прием и выдачу в подсоединенный накопитель управляющих сигналов, служебной информации, байтов данных, а также выполнение операций логического поиска, считывания и записи.

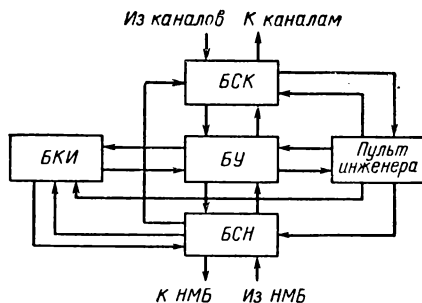


Рис. 11.8. Схема устройства управления ЕС-5533

Блок контроля информации служит для подсчета контрольных сумм данных, поступающих из селекторного канала, и выдачи их через БСН в логически подключенный накопитель, а также для подсчета контрольных сумм данных, получаемых из НБМ, и сравнения их с контрольными суммами, поступающими из накопителя.

На пульте инженера размещаются органы ручного управления и индикации, необходимые для проведения профилактического контроля устройства управления в автономном режиме и переключения его из автономного режима работы в комплексный.

Основные функции блока управления: синхронизация работы других блоков УУ, прием из БСК кодов команд и их дешифрация, выполнение команд, не требующих передачи данных, выдача на исполнение команд, требующих передачи данных, отклонение команд, нарушающих маску защиты накопителя.

УСТРОЙСТВА ВВОДА-ВЫВОДА
И ПОДГОТОВКИ ДАННЫХ

Основные устройства ввода-вывода машин Единой системы имеют одиночные (встроенные) УВУ, поэтому к каждому каналу, селекторному или мультиплексному, из-за ограниченных возможностей интерфейса можно подключить сравнительно небольшое количество УВВ. Для увеличения количества подключаемых к каналам УВУ, т. е. и УВВ, используются специальные логические ретрансляторы. Большую часть устройств ввода-вывода составляют устройства электромеханического типа, обеспечивающие ввод информации с перфокарт и перфолент, а также информации, набираемой на клавиатурах выносных пультов и пишущих машин, и вывод информации на перфоленты, перфокарты и на рулонную бумагу в алфавитно-цифровом виде или в виде графиков. В системе устройств ввода-вывода используются выносные пульты с экранами для индикации информации, а также устройства ввода и вывода алфавитно-цифровой и графической информации на электронно-лучевых трубках со световым карандашом.

Устройства ввода-вывода электромеханического типа включают в свой состав штатные механизмы (фотосчитывающие, печатающие, перфорационные и т. п.) и электронные схемы управления. Для каждого из них определен индивидуальный набор команд, определяющий все возможные режимы работы, а также наборы указателей в байте основного состояния и байте уточненного состояния; наборы команд и информацию, содержащуюся в байтах состояния, обычно относят к системным характеристикам устройств ввода-вывода.

При рассмотрении устройств ввода-вывода машин Единой системы основное внимание уделяется общим принципам их построения и работы, их системным характеристикам, особенностям функционирования при взаимодействии с каналами посредством стандартного интерфейса ввода-вывода. Для электромеханических частей устройств даются общие характеристики, основной состав и принципы взаимодействия со схемами управления.

12.1. УСТРОЙСТВА ВВОДА ИНФОРМАЦИИ С ПЕРФОКАРТ

В машинах Единой системы используются устройства ввода информации с перфокарт двух типов: ЕС-6012 и ЕС-6013. Оба эти устройства выполнены по одинаковому общему схематическому

чаются, главным образом, скоростью считывания информации, нанесенной в виде системы пробивок на стандартные 80-колонковые или 45-колонковые перфокарты. Считывание информации осуществляется фотоэлектрическим способом при подаче карт узкой стороной; кодирование информации на носителе — в коде КПК-12, на выходе устройств — в коде ДКОИ, являющимся основным для передачи информации в машинах Единой системы.

Оба устройства могут работать как в мультиплексном, так и в монопольном режиме и подключаться к мультиплексным или селекторным каналам моделей ЭВМ Единой системы через стандартный интерфейс ввода-вывода. Скорость считывания для устройства ЕС-6012 — до 500 карт/мин, для устройства ЕС-6013 — до 1200 карт/мин; емкости подающего и приемного магазинов перфокарт для ЕС-6012 — до 1000 карт, для ЕС-6013 — до 2500 карт.

Используемый для кодирования информации на перфокартах код КПК-12 обеспечивает представление любого из возможных для ЕС ЭВМ 256 символов в виде системы пробивок на одной из колонок перфокарты. Каждая колонка имеет 12 позиций по числу строк перфокарты, нумеруемых сверху вниз в таком порядке: 12, 11, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Поэтому при использовании предельной информационной возможности системы пробивок по одной колонке можно закодировать до $2^{12} = 4096$ символов.

Имеющаяся информационная избыточность используется для контроля правильности нанесения и считывания кодов; с этой целью КПК-12 построен так, что любому символу отвечает на данной колонке перфокарты комбинация, содержащая не более одной пробивки (код 1) на позициях 1—7.

Таблица кода КПК-12 приведена на рис. 12.1. Позиции 12, 11, 0 определяют зональную группу, а позиции 1, 2, 3, 4, 5, 6, 7, 8, 9 — цифровую группу кода. Все используемые символы размещены в основной части таблицы на пересечениях столбцов и строк соответствующих зональных и цифровых групп, которые определяют позиции пробивок при кодировании символов. Например, символ «:» расположен на пересечении столбца с «пустой» зональной группой и строки с цифровой группой «8—2»; этому символу отвечает комбинация на колонке перфокарты с пробивками на 2-й и 8-й позициях. Символ «б» расположен на пересечении столбца с зональной группой «12, 11, 0» и строки с цифровой группой «9—8»; этому символу отвечает комбинация на колонке перфокарты с пробивками на 12, 11, 0, 8 и 9-й позициях и т. п.

Рассматриваемые устройства обеспечивают считывание информации с перфокарт и при ее представлении в произвольном коде. В этом случае с каждой колонки считываются два 6-разрядных байта без их преобразования.

Общая схема устройства ввода информации с перфокарт приведена на рис. 12.2. В соответствии с этой схемой оно (вместе

		Зональная группа																				
		12	11	0	12	11	0	12	11	0	12	11	0	12	11	0	12	11	0			
Цифровая группа	0	&	-	0	Про дет	{		}									ц	и	я	ь	8-7	
	1	A	J	/	1	a	j	-	ы	НЗ	СУ1	Д01	Д17							Д31		9-7
	2	B	K	S	2	b	k	s	з	НТ	СУ2	Д02	С1И									9-2
	3	C	L	T	3	c	l	t	ш	КТ	СУ3	Д03	Д19									9-3
	4	D	M	U	4	d	m	u	э	Д28	Д29	Д04	Д20									9-4
	5	E	N	V	5	e	n	v	ц	ГТ	Д05	ПС	Д21									9-5
	6	F	O	W	6	f	o	w	ч	Д06	ВШ	КБ	Д22							ю		9-6
	7	G	P	X	7	g	p	x	ь	ЗБ	Д07	АР2	КП									9-7
	8	H	Q	Y	8	h	q	y	ю	Д23	АН	Д08	Д24								б	9-8
	9	I	R	Z	9	i	r	z		Д13	КН	Д09	Д25	ПРС	АР1	Д00	Д16					9-8-1
	8-2	[]	\	:	ø	к		Б	Д14	Д18	Д10	Д26								з	9-8-2
	8-3	•	α	,	#		л		Ц	ВТ	Д15	Д11	Д27	И		У	Ш					9-8-3
	8-4	<	*	%	©	Ф	М	Т	Д	ПФ	РФ	Д12	СТП	Й	Л	Ж	Э					9-8-4
	8-5	()	-	'	Г	Н			ВК	РГ	КТМ	НЕТ			Я	Щ					9-8-5
8-6	+	;	>	=		ж	Ф		РЧС	РЗ	ДА	Д30	Л		б	Ч					9-8-6	
8-7	!	^	?	"	и	п	ø	Г	ЛАТ	РЗ	ЗВ	ЗМ			Ы	ЗБ					9-8-7	

Рис. 12.1. Структура кода КПК-12

с УВУ) имеет в своем составе: механизм подачи, транспортировки и укладки перфокарт (МПК), блок управления (БУ), блок синхронизации (БС), блок восприятия и контроля информации (БВКИ), блоки сопряжения с каналом (БСК).

Механизм подачи, транспортировки и укладки перфокарт обеспечивает поочередное перемещение перфокарт от подающего магазина к приемному магазину с протяжкой их через блок фотоэлектрического считывания. По всему тракту прохождения перфокарт расположены датчики контроля, сигнализирующие о наличии карт в подающем магазине, переполнении приемного магазина, замятии карт, двойной подаче одной и той же карты, неподаче очередной карты и о положении основных деталей механизма. Механическая часть устройства рассчитана на работу в стартстопном режиме с остановками после считывания информации с одной перфокарты.

Блок восприятия и контроля информации служит для непосредственного считывания информации с перфокарт, контроля информации и ее преобразования перед выдачей информационных байт через БУ в БСК.

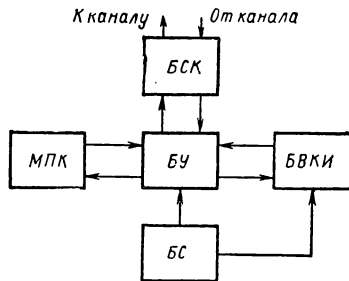


Рис. 12.2. Общая схема устройства ввода информации с перфокарт

Считывание информации осуществляется с помощью 12 фотодиодов, расположенных по прямой, перпендикулярной направлению перемещения перфокарт; засветка фотодиодов через пробивки на перфокартах производится с помощью ламп накаливания. Сигналы фотодиодов усиливаются, и по колонным кодовым комбинациям поступают в кодопреобразователь, где осуществляется контроль на несуществующую в КПК-12 комбинацию и формирование 8-разрядных комбинаций кода ДКОИ. Если выполняется команда по считыванию информации без преобразования, то контроль и преобразование не производятся. Отметим, что в БВКИ всегда осуществляется контроль на разрыв перемычек между пробивками на соседних колонках, а также контроль работы ламп подсветки, фотодиодов и схемы синхронизации.

Блок синхронизации формирует синхросигналы в моменты считывания информации с каждой колонки перфокарты и обеспечивает синхронизацию работы всех схем и механизма устройства. Источником синхросигналов служит фотодиод, в промежутке между которым и лампой подсветки при работе механизма устройства вращается непрозрачный диск с прорезями.

Блок управления устройством осуществляет включение основного двигателя МПК, слежение за прохождением перфокарт по всему тракту их перемещения, пуск и останов устройства, координирование работы БВКИ и БСК. В качестве первичной управляющей информации в БУ используются сигналы, поступающие из БСК, БС, БВКИ, а также от контрольных датчиков, установленных в механизме подачи, транспортировки и укладки перфокарт.

Блок сопряжения с каналом обеспечивает подключение устройства к интерфейсу и его взаимодействие с каналом при выполнении команд. Основные функции БСК:

- установление связи между устройством и каналом путем обмена соответствующими наборами сигналов по линиям интерфейса;

- преобразование последовательностей сигналов, поступающих из канала, а также кодов команд в сигналы, необходимые для управления работой устройства;

- хранение байт команды, основного и уточненного состояния;

- декодирование, опознание и выдача в канал своего адреса;

- формирование контрольных разрядов для байт данных, байт основного и уточненного состояния перед выдачей их на ШИН-А для передачи в канал; отметим, что в случае считывания информации с преобразованием кода КПК-12 в ДКОИ каждой колонке перфокарты отвечает один байт данных, а в случае считывания без преобразования (информация нанесена на перфокарты в произвольном коде) — два 6-разрядных, т. е. неполных, байта данных.

Устройства ввода информации с перфокарт ЕС-6012 и ЕС-6013 выполняют команды канала, представленные в табл. 12.1, где

Т а б л и ц а 12.1

Команды	Разряды байта команды							
	0	1	2	3	4	5	6	7
ЧИТАТЬ С ПРЕОБРАЗОВАНИЕМ	А	А	0	0	А	А	1	0
ЧИТАТЬ БЕЗ ПРЕОБРАЗОВАНИЯ	А	А	1	0	А	А	1	0
ПРОВЕРИТЬ ВВОД-ВЫВОД	0	0	0	0	0	0	0	0
УТОЧНИТЬ СОСТОЯНИЕ	0	0	0	0	0	1	0	0
ХОЛОСТОЙ ХОД	0	0	0	0	0	0	1	1

символом «А» обозначены разряды кода в байте команды игнорируемые, т. е. не воспринимаемые устройствами.

По команде **ЧИТАТЬ С ПРЕОБРАЗОВАНИЕМ** производится считывание информации с одной перфокарты при преобразовании кода КПК-12 в код ДКОИ. При преобразовании кода производится контроль считанной информации на недопустимую комбинацию. В случае выявления недопустимой комбинации процесс считывания и передачи информации в канал продолжается в обычном порядке, но в устройстве фиксируется признак ошибки в данных; по окончании выполнения команды в устройстве формируется запрос на передачу байта основного состояния, содержащего указатель **СБОЙ В УСТРОЙСТВЕ**.

По команде **ЧИТАТЬ БЕЗ ПРЕОБРАЗОВАНИЯ** производится считывание информации с одной перфокарты без преобразования считанного с перфокарты кода. Информация, считанная с одной колонки перфокарты, передается в виде пакета из двух байт данных: первый содержит информацию по 12, 11, 0, 1, 2 и 3-й позициям колонки, а второй — по 4, 5, 6, 7, 8 и 9-й позициям той же колонки. Любая комбинация пробивок в колонке считается допустимой.

Команда **ПРОВЕРИТЬ ВВОД-ВЫВОД** посылается в устройство для вызова в канал байта основного состояния. Если устройство выполняет другую команду, то рассматриваемая команда не оказывает никакого влияния на ход ее выполнения; в канал передается байт состояния с указателем **ЗАНЯТО**. Если же к моменту поступления команды **ПРОВЕРИТЬ ВВОД-ВЫВОД** выполнение предыдущей команды завершено и сформирован байт основного состояния, еще не принятый каналом, то устройство выдает в канал этот байт без указателя **ЗАНЯТО**.

Команда **УТОЧНИТЬ СОСТОЯНИЕ** посылается в устройство для вызова в канал байта уточненного состояния. При начальной выборке устройство отвечает на эту команду нулевым байтом состояния. В остальных случаях устройство выдает в ка-

нал сначала байт уточненного состояния, а затем — байт основного состояния с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО.

Команда ХОЛОСТОЙ ХОД используется для исключения той или другой команды из программы канала без нарушения в ней общей последовательности команд. По этой команде при начальной выборке устройство выдает в канал указатели КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО.

Байт основного состояния выдается устройством в канал в следующих случаях:

- при начальной выборке устройства;
- по окончании передачи данных (с указателем КАНАЛ КОНЧИЛ);
- по окончании полного механического цикла устройства (с указателем ВУ КОНЧИЛО);
- при переводе устройства из состояния НЕ ГОТОВО в состояние ГОТОВО;
- при выдаче в канал так называемого запомненного состояния.

Разряды 0, 1 и 2-й в байте основного состояния не используются. Наличие единицы в 3-м разряде представляет указатель ЗАНЯТО, в 4-м разряде — указатель КАНАЛ КОНЧИЛ, в 5-м разряде — указатель ВУ КОНЧИЛО, в 6-м разряде — указатель СБОЙ В УСТРОЙСТВЕ, в 7-м разряде — указатель ОСОБЫЙ СЛУЧАЙ.

Указатель ЗАНЯТО выдается только при начальной выборке. Он указывает на то, что устройство не может приступить к выполнению поступившей из канала команды, так как выполняется ранее начатая операция или устройство ожидает удовлетворения запроса на передачу байта основного состояния, содержащего указатели КАНАЛ КОНЧИЛ и (или) ВУ КОНЧИЛО. При этом считается, что операция выполняется с момента приема каналом начального байта основного состояния до момента приема им указателя ВУ КОНЧИЛО. Указатель ЗАНЯТО выдается также в ответ на команду ПРОВЕРИТЬ ВВОД-ВЫВОД, если выполнение ранее начатой операции не закончено и в устройстве не сформирован байт состояния.

Указатель КАНАЛ КОНЧИЛ формируется при окончании передачи данных между устройством и каналом. Передача данных прекращается либо по концу перфокарты, либо по инициативе канала (в случае получения нуля при счете передаваемых данных по реализуемому каналом УСК для данного устройства), а также при некоторых ситуациях (замятии или неподаче перфокарты, обнаружении сбоя в работе функциональных частей устройства).

Указатель ВУ КОНЧИЛО означает, что устройство закончило выполнение операции по вводу данных и может принять новую команду. Этот указатель формируется также при переводе устройства из состояния НЕ ГОТОВО в состояние ГОТОВО.

Указатель СБОЙ В УСТРОЙСТВЕ является общим при выявлении любых ошибок и сбоев. Он указывает на то, что в устройстве сформированы один или несколько разрядов — указателей байта уточненного состояния, дающих более подробную характеристику выявленных неисправностей.

Указатель ОСОБЫЙ СЛУЧАЙ свидетельствует о том, что было произведено считывание информации с последней перфокарты заложенного в подающий магазин устройства массива при нажатой на панели управления клавише «КОНЕЦ КАРТОТЕКИ». Этот указатель посылается в канал при начальной выборке в ответ на команду ЧТЕНИЕ; после этого устройство автоматически переключается в состояние НЕ ГОТОВО.

Байт уточненного состояния содержит информацию о причине возникновения ошибки или сбоя в работе устройства. Он выдается в канал по команде УТОЧНИТЬ СОСТОЯНИЕ. Нулевой разряд байта уточненного состояния отведен для размещения указателя КОМАНДА ОТВЕРГНУТА, 1-й разряд — для указателя ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО, 2-й разряд — для указателя ОШИБКА НА ШИН-К, 3-й разряд — для указателя СБОЙ В ОБОРУДОВАНИИ, 4-й разряд — для указателя ОШИБКА В ДАННЫХ, 5-й разряд — для указателя ПЕРЕПОЛНЕНИЕ, 6-й и 7-й разряды байта не используются.

Указатель КОМАНДА ОТВЕРГНУТА формируется в случае поступления кода команды, которую устройство не может выполнить. Если поступивший код команды имеет ошибку по четности, то формируется указатель ОШИБКА НА ШИН-К.

Наличие указателя ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО свидетельствует о неготовности устройства к работе. Этот указатель формируется в следующих случаях:

- при замятии или неподаче перфокарты;
- при выявлении сбоя в начале цикла считывания;
- в случае установления факта двойной подачи одной карты;
- после считывания информации с последней перфокарты массива и выдачи в канал байта основного состояния, содержащего указатель ОСОБЫЙ СЛУЧАЙ;
- после считывания информации с последней перфокарты массива, если клавиша «КОНЕЦ КАРТОТЕКИ» на панели управления устройством не была нажата;
- при нажатии клавиши «СТОП»;
- при переполнении приемного кармана.

После устранения неисправности и перевода устройства в состояние ГОТОВО в канал от устройства выдается запрос на передачу байта состояния, содержащего указатель ВУ КОНЧИЛО.

Указатель ОШИБКА НА ШИН-К формируется при приеме устройством кода команды, имеющего ошибку по четности. В этом случае операция в устройстве не выполняется.

Указатель СБОЙ В ОБОРУДОВАНИИ формируется тогда, когда при выполнении операции фиксируется сбой в работе схем устройства. Конкретными причинами формирования данного указателя являются:

- отказ ламп подсветки или самих фотодиодов;
- неисправность цепей синхронизации;
- разрыв перемычек между пробивками на соседних колонках перфокарты в процессе считывания информации;
- неисправность каналов считывания и синхронизации;
- проскальзывание перфокарты в транспортных валиках.

Формирование указателя СБОЙ В ОБОРУДОВАНИИ в начале цикла считывания информации с перфокарты приводит к преждевременному окончанию выполнения заданной операции, так как дальнейшее продолжение процесса считывания информации с перфокарты становится бессмысленным. В канал выдается байт состояния с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО, СБОЙ В УСТРОЙСТВЕ. Если указатель СБОЙ В ОБОРУДОВАНИИ сформирован после передачи данных, то устройство посылает в канал байт основного состояния с указателем КАНАЛ КОНЧИЛ; информация об ошибке передается в последующем байте состояния вместе с указателем ВУ КОНЧИЛО.

Указатель ОШИБКА В ДАННЫХ формируется тогда, когда при выполнении операции чтения с преобразованием считанная по некоторой колонке перфокарты комбинация единиц и нулей является недопустимой; при этом выполнение заданной операции не прекращается. Для байта данных, полученного по недопустимой комбинации, в обычном порядке формируется контрольный разряд; в обычном же порядке этот байт выдается на ШИН-А.

Указатель ПЕРЕПОЛНЕНИЕ формируется в том случае, когда канал не в состоянии удовлетворить требование устройства на передачу очередного байта данных; этот байт данных теряется, так как в устройстве по его внутреннему циклу работы автоматически осуществляется переход к считыванию информации с последующей колонки перфокарты. Формирование рассматриваемого указателя приводит к прекращению передачи данных из устройства в канал.

12.2. УСТРОЙСТВА ВЫВОДА ИНФОРМАЦИИ НА ПЕРФОКАРТЫ

Для вывода информации на перфокарты в машинах Единой системы могут использоваться устройства ЕС-7010 и ЕС-7012, отличающиеся друг от друга в основном скоростью перфорации карт. Информация наносится на перфокарты путем построичной пробивки; карты под пуансоны, связанные с пробивными электромагнитами, подаются широкой стороной. Оба устройства рас-

считаны на обработку 80-колонковых перфокарт, поэтому их пробивные механизмы включают в свой состав по 80 электромагнитов, управляющих движением пуансонов, непосредственно осуществляющих выполнение пробивок на перфокартах.

Устройства могут работать как в мультиплексном, так и в монопольном режиме и подключаться к мультиплексным или селекторным каналам моделей ЭВМ Единой системы через стандартный интерфейс ввода-вывода. Скорость перфорации для ЕС-7010 — до 100 карт/мин, для ЕС-7012 — до 250 карт/мин. Каждое устройство, помимо подающего кармана, имеет по два приемных кармана на 700 перфокарт каждый.

Информация на перфокарты наносится либо в коде КПК-12, либо в произвольном 6-разрядном коде (по две комбинации на каждой колонке). Во всех случаях информация из канала принимается побайтно. Байт кода ДКОИ на входе устройства должен преобразовываться в одну комбинацию КПК-12, которая в дальнейшем наносится на одну колонку перфокарты. При использовании произвольного 6-разрядного кода преобразование информации не производится, но система пробивок на одной колонке отвечает теперь двум байтам данных, принимаемым из канала.

При использовании для передачи информации из канала в устройство ЕС-7010 (ЕС-7012) как кода ДКОИ, так и произвольного 6-разрядного кода возникает еще необходимость «разворота» массива информации, наносимой на одной перфокарте. Действительно, из канала данные поступают в виде параллельно-последовательного кода, ориентированного по колонкам, а перфорация осуществляется по строкам перфокарты, т. е. на основе параллельно-последовательного кода, ориентированного по строкам. После преобразования на входе устройства принятый параллельно-последовательный код для одной перфокарты имеет «размер» 12×80 бит, а для нанесения информации на перфокарту необходимо использовать код 80×12 бит.

Своеобразный «разворот» кода на 90° может быть осуществлен только после поступления в устройство всех байт данных, необходимых для перфорации одной карты (80 байт для кода КПК-12 или 160 байт для произвольного 6-разрядного кода). Поэтому в составе устройства должно использоваться достаточно быстродействующее запоминающее устройство емкостью не менее 160 байт. А общий цикл нанесения информации на одну перфокарту разбивается на две основные части: прием и перекодирование информации с ее запоминанием в местном ОЗУ и собственно перфорация с предварительным «разворотом» общего параллельно-последовательного кода.

Упрощенная общая схема устройства типа ЕС-7012 приведена на рис. 12.3. В соответствии с этой схемой оно (вместе с УВУ) имеет в своем составе: блок сопряжения с каналом (БСК), схему приема и формирования управляющих сигналов (СхПФУС); узел перекодирования информации (УПИ), оперативное запоминаю-

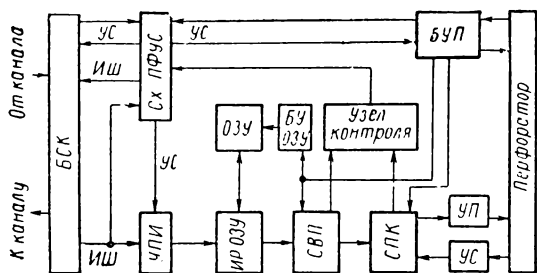


Рис. 12.3. Общая схема устройства вывода информации на перфокарты

щее устройство (ОЗУ), блок управления ОЗУ (БУ ОЗУ), информационный регистр ОЗУ (ИР ОЗУ), схему выделения позиций (СВП), узел контроля (УК), схему преобразования кодов (СПК), собственно перфоратор, блок управления перфоратором (БУП), усилители считывания (УС) и усили-

тели сигналов для пробивных электромагнитов (УП).

Основную электромеханическую часть устройства составляет перфоратор ПК-250, выполненный по обычной схеме механизмов перфорации карт с построчным пробивным устройством. Подвод перфокарт под пуансоны производится системой транспортных роликов; после осуществления пробивок карта может направляться в один из двух приемных карманов. По всему тракту перемещения перфокарт установлены датчики, реагирующие на отклонения в движении карт. Кроме того, в собственно перфораторе имеются узлы, обеспечивающие формирование синхросигналов, отражающих номер строки (позиции) перфокарты, находящейся под пуансонами, и строки, подводимой на очередном шаге под пуансоны. В состав перфоратора включено также фотосчитывающее устройство с 80 фотодиодами, обеспечивающее контрольное считывание информации, нанесенной на перфокарту. Считанные построчные коды через усилители УС и схему преобразования кодов направляются в узел контроля.

Блок управления перфоратором обеспечивает формирование необходимых синхросигналов и сигналов позиций для управления работой узлов перфоратора. Он непосредственно управляет включением и выключением основного приводного двигателя, подачей и транспортировкой карт, их перфорацией, посылкой отперфорированных карт в приемные карманы (карман брака). Блок управления осуществляет также контроль за прохождением перфокарт по всему тракту. Необходимые управляющие сигналы (УС) из БУП посылаются в схему приема и формирования управляющих сигналов, блок управления ОЗУ, схему выделения позиций и схему преобразования кодов.

Блок сопряжения с каналом обеспечивает стандартное сопряжение устройства с каналом модели ЭВМ через интерфейс ввода-вывода. Главные функции блока сводятся к следующему:

- декодирование, опознание и выдача в канал присвоенного данному устройству адреса;
- установление связи устройства с каналом;
- преобразование последовательностей сигналов, поступа-

ющих из канала, и кодов команд в наборы сигналов управления, обеспечивающих заданные режимы работы устройства;

— формирование сигналов, направляемых в канал по соответствующим линиям (шинам) интерфейса;

— хранение байт команды, основного и уточненного состояний.

Блок сопряжения с каналом для рассматриваемого устройства вывода информации на перфокарты выполняет основные функции одиночного УВУ. Он обменивается управляющими сигналами со схемой приема и формирования УС. По информационным шинам ИШ он передает байты данных в узел перекодирования информации, а код команды — в СхПФУС, от которой получает информацию о состоянии устройства. Связь БСК с каналом обеспечивается посредством всех функциональных линий интерфейса.

Схема приема и формирования управляющих сигналов является своеобразным связующим звеном между блоком сопряжения с каналом и блоком управления перфоратором. Она содержит дешифратор кодов команд, цепочки элементов, обеспечивающих формирование управляющих сигналов, регистр состояния, счетчик и узел передачи байт уточненного состояния в блок сопряжения с каналом.

Узел перекодирования информации обеспечивает преобразование 8-разрядных комбинаций ДКОИ в 12-разрядные комбинации КПК-12 с разделением каждой из последних на две 6-разрядные части. Основу узла составляет комбинационная схема типа шифратора, имеющая 8 входов и 12 выходов. Первая группа выходных вентилях, подключенных к первым шести выходам схемы преобразования, обеспечивает передачу в информационный регистр ОЗУ кодов 12, 11, 0, 1, 2, и 3 позиций для одной колонки перфокарты, а вторая группа выходных вентилях — кодов 4, 5, 6, 7, 8 и 9-й позиций для той же колонки. Два байта произвольного 6-разрядного кода, минуя схему преобразования, выдаются последовательно в информационный регистр ОЗУ через эти же группы выходных вентилях. Таким образом, информация, наносимая на любую колонку перфокарты, всегда разделяется на две части.

Оперативное запоминающее устройство служит буферным накопителем информации, наносимой на одну перфокарту. Оно имеет емкость 256 байт и построено по матричному принципу. Время обращения к ОЗУ — 4 мкс, время выборки — 1,5 мкс. Информационный регистр ОЗУ служит для размещения кодов, записываемых в ОЗУ или считываемых из него. Блок управления ОЗУ обеспечивает выполнение всех действий, связанных с записью и считыванием кодов. Регистр адреса выполнен в виде счетчика, содержимое которого увеличивается на единицу в каждом такте записи. Поэтому информация, которая должна наноситься по колонкам перфокарты, размещается в ячейках с после-

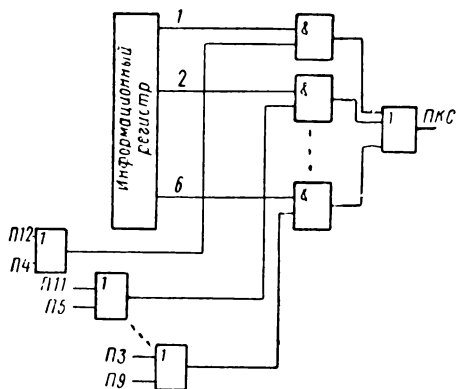


Рис. 12.4. Схема выделения позиций

довательными номерами, отвечающими номерам колонок. При этом информация для одной колонки размещается в двух смежных ячейках (нечетной и четной). В конечном итоге оказывается, что в нечетных ячейках ОЗУ содержится информация для 12, 11, 0, 1, 2 и 3-й позиций всех колонок, а в четных — информация для 4, 5, 6, 7, 8 и 9-й позиций также всех колонок перфокарты.

Схема выделения позиций обеспечивает «разворот» по колонному параллельно-последовательного кода в построчной. При этом код каждой строки (80-разрядный) на выходе схемы — последовательный. Принципы построения и работы схемы поясняются с помощью рис. 12.4.

Собственно перфорация карты начинается после записи информации в 160 ячеек ОЗУ. Первой под пуансоны подается 12-я (первая по счету) строка перфокарты, поэтому на вход схемы выделения позиций подается сигнал П12 (12-я позиция). Производится последовательное считывание содержимого нечетных ячеек ОЗУ, начиная с первой. Очевидно, что из каждой 6-разрядной комбинации с помощью первого элемента *И* будет выделяться значение разряда, отвечающего 12-й позиции соответствующей колонки. Таким образом, в ходе считывания информации из нечетных ячеек ОЗУ при действии сигнала П12 на выходе конечного элемента *ИЛИ* формируется последовательный код строки (ПКС), в данном случае отвечающей 12-й позиции. При считывании всегда осуществляется перезапись информации, поэтому она сохраняется до окончания перфорации одной карты.

После перемещения перфокарты на один шаг, т. е. на следующую позицию, на вход рассматриваемой схемы подается сигнал П11, обеспечивающий в процессе считывания информации из нечетных ячеек формирование последовательного кода строки, отвечающей 11-й позиции, т. д. Циклы считывания из нечетных ячеек ОЗУ повторяются шесть раз. После этого шесть раз информация считывается из четных ячеек, а на входные элементы *ИЛИ* последовательно подаются сигналы П4, П5, П6, П7, П8 и П9. В результате таких действий формируются последовательные коды для всех 12 строк перфокарты. При перфорации эти коды направляются в схему преобразования кодов, а при контрольном считывании с отперфорированной карты — в узел контроля.

Схема преобразования кодов служит для преобразования последовательных кодов строк в параллельные и распределения

позиционных сигналов по усилителям, связанным с пробивными электромагнитами. Ее основу составляет сдвигающий регистр с группами вентиляей. При контроле отперфорированной карты схема преобразования преобразуется параллельный код строки, поступающий с усилителей считывания, в последовательный код, направляемый в узел контроля. Этот код сравнивается с кодом, формируемым по информации, записанной в ОЗУ. В случае выявления ошибки схемы управления формируют сигналы, по которым отперфорированная с ошибкой карта направляется в карман брака.

Устройства вывода информации на перфокарты выполняют команды канала, представленные табл. 12.2.

Т а б л и ц а 12.2

Команда	Разряды байта команды							
	0	1	2	3	4	5	6	7
ЗАПИСАТЬ С ПРЕОБРАЗОВАНИЕМ И ПОДАТЬ	0	1	0	0	0	0	0	1
ЗАПИСАТЬ БЕЗ ПРЕОБРАЗОВАНИЯ И ПОДАТЬ	0	1	1	0	0	0	0	1
ЗАПИСАТЬ С ПРЕОБРАЗОВАНИЕМ, ВЫБРАТЬ КАРМАН 1 И ПОДАТЬ	0	0	0	0	0	0	0	1
ЗАПИСАТЬ БЕЗ ПРЕОБРАЗОВАНИЯ, ВЫБРАТЬ КАРМАН 1 И ПОДАТЬ	0	0	1	0	0	0	0	1
ПРОВЕРИТЬ ВВОД-ВЫВОД	0	0	0	0	0	0	0	0
УТОЧНИТЬ СОСТОЯНИЕ	0	0	0	0	0	1	0	0
ХОЛОСТОЙ ХОД	0	0	0	0	0	0	1	1

По команде ЗАПИСАТЬ С ПРЕОБРАЗОВАНИЕМ И ПОДАТЬ производится перфорация одной карты с преобразованием 8-разрядного кода ДКОИ в перфокартный код КПК-12. Во время выполнения команды информация из канала поступает в буферную память устройства. После завершения передачи информации в устройстве формируется байт основного состояния с указателем КАНАЛ КОНЧИЛ, который передается в канал. Отперфорированная карта укладывается в приемный карман 2, а очередная перфокарта подается из подающего кармана. Во время выполнения команды в устройстве ЕС-7010 производится контроль правильности выполнения пробивок на предыдущей перфокарте, а в устройстве ЕС-7012 — контроль правильности выполнения пробивок на перфорируемой карте. В случае выявления ошибки неправильно отперфорированная карта автоматически помещается в приемный карман 1, служащий в этом случае карманом брака,

и в устройстве формируется байт основного состояния с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО, СБОЙ В УСТРОЙСТВЕ.

По команде ЗАПИСАТЬ БЕЗ ПРЕОБРАЗОВАНИЯ И ПОДАТЬ выполняются действия, аналогичные действиям, выполняемым по команде записи с преобразованием; основное отличие состоит в том, что на входе устройства не осуществляется преобразование кодов. Информация, содержащаяся в двух байтах, последовательно поступающих из канала, размещается на одной колонке перфокарты.

Команда ЗАПИСАТЬ С ПРЕОБРАЗОВАНИЕМ, ВЫБРАТЬ КАРМАН 1 И ПОДАТЬ отличается от первой из рассмотренных команд только тем, что при ее выполнении отперфорированная карта помещается в приемный карман 1; карманом брака в этом случае является приемный карман 2.

Команда ЗАПИСАТЬ БЕЗ ПРЕОБРАЗОВАНИЯ, ВЫБРАТЬ КАРМАН 1 И ПОДАТЬ отличается от второй из рассмотренных команд только тем, что при ее выполнении отперфорированная карта помещается в приемный карман 1.

Команда ПРОВЕРИТЬ ВВОД-ВЫВОД посылается в устройство для вызова в канал байта основного состояния. Она не оказывает никакого влияния на ход выполнения в устройстве другой команды, если поступает до ее завершения; в этом случае устройство выдает в канал признак занятости. Если же к моменту поступления рассматриваемой команды в устройство уже сформирован байт основного состояния, еще не послышавшийся в канал, то он выдается в канал без указателя ЗАНЯТО.

Команда УТОЧНИТЬ СОСТОЯНИЕ посылается в устройство для вызова в канал байта уточненного состояния. При начальной выборке устройство отвечает на эту команду нулевым байтом состояния. После передачи байта уточненного состояния устройство выдает в канал байт основного состояния с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО.

Команда ХОЛОСТОЙ ХОД используется для исключения некоторой команды из программы канала (канальной программы) без нарушения в ней общей последовательности команд. По этой команде в процессе начальной выборки устройство выдает в канал указатели КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО.

Байт основного состояния выдается устройством в канал в следующих случаях:

- при начальной выборке устройства;
- по окончании приема данных из канала (с указателем КАНАЛ КОНЧИЛ);
- по окончании полного механического цикла устройства, т. е. после завершения перфорации одной карты (с указателем ВУ КОНЧИЛО);
- при переводе устройства из состояния НЕ ГОТОВО в состояние ГОТОВО;

— при передаче в канал так называемого запомненного состояния, т. е. байта состояния, сформированного до поступления из канала соответствующей команды.

Разряды 0, 1, 2 и 7-й в байте основного состояния не используются. Наличие единицы в третьем разряде представляет указатель ЗАНЯТО, в четвертом — указатель КАНАЛ КОНЧИЛ, в пятом — ВУ КОНЧИЛО, в шестом — СБОЙ В УСТРОЙСТВЕ.

Указатель ЗАНЯТО выдается в канал только в процессе начальной выборки. Он свидетельствует о том, что устройство не может приступить к выполнению команды, так как оно выполняет ранее начатую операцию или ожидает удовлетворения запроса на передачу байта основного состояния, содержащего указатели КАНАЛ КОНЧИЛ и (или) ВУ КОНЧИЛО. При этом считается, что операция в устройстве выполняется с момента приема каналом начального байта основного состояния до момента приема им указателя ВУ КОНЧИЛО. Если ранее начатая операция не окончена и в устройстве не хранится сформированный байт состояния, то устройство в ответ на поступившую команду ПРОВЕРИТЬ ВВОД-ВЫВОД отвечает указателем ЗАНЯТО.

Указатель КАНАЛ КОНЧИЛ устанавливается после окончания передачи данных из канала в устройство. Передача данных прекращается по инициативе канала, если в команде канала разряды счета данных становятся нулевыми, или по инициативе устройства (после приема 80 байт данных в режиме работы с преобразованием кодов, а в режиме работы без преобразования кодов — после приема 160 байт данных). Передача данных прекращается также при выявлении ошибки четности в байте данных, при замятии перфокарты и при выявлении ошибки в работе системы синхронизации устройства.

Указатель ВУ КОНЧИЛО устанавливается после завершения устройством операции по выводу информации на перфокарту; он свидетельствует о том, что устройство может принять очередную команду. Этот указатель устанавливается также при переводе устройства в состояние ГОТОВО.

Указатель СБОЙ В УСТРОЙСТВЕ свидетельствует о том, что сформированы один или несколько разрядов байта уточненного состояния, характеризующие выявленные в процессе работы устройства ошибки.

Байт уточненного состояния содержит информацию о причине возникновения ошибки или сбоя в работе устройства. Он выдается в канал по команде УТОЧНИТЬ СОСТОЯНИЕ. Нулевой разряд байта уточненного состояния используется для размещения указателя КОМАНДА ОТВЕРГНУТА, 1-й разряд — для указателя ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО, 2-й разряд — для указателя ОШИБКА НА ШИН-К, 3-й разряд — для указателя СБОЙ В ОБОРУДОВАНИИ, 4—7 разряды байта не используются.

Указатель КОМАНДА ОТВЕРГНУТА формируется в случае поступления кода команды, которую устройство не может выполнить. Если поступивший код команды имеет ошибку по четности, то формируется указатель ОШИБКА НА ШИН-К.

Указатель ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО свидетельствует о том, что устройство не готово к работе. Он формируется (устанавливается) в следующих случаях:

- при замятии или неподаче перфокарты;
- при нажатии клавиши «СТОП»;
- при обнаружении ошибки в работе системы синхронизации устройства;
- при подаче из подающего кармана последней перфокарты;
- при переполнении приемных карманов или коробки для конфетти, т. е. для кусочков перфокарты, вырезаемых из нее пуансонами при выполнении пробивок;
- при обнаружении двойной подачи перфокарт.

После перевода устройства из состояния НЕ ГОТОВО в состояние ГОТОВО оно посылает в канал запрос на передачу байта основного состояния, содержащего указатель ВУ КОНЧИЛО.

Указатель ОШИБКА НА ШИН-К показывает, что устройство приняло код команды или байт данных с ошибкой по четности. Если код команды с ошибкой по четности принят при начальной выборке, то устройство выдает в канал байт основного состояния с указателем СБОЙ В ОБОРУДОВАНИИ; команда при этом не выполняется. Если принимается байт данных с ошибкой по четности, то выполнение операции записи информации (вывод информации на перфокарты) в устройстве немедленно прекращается: устройство выдает запрос на передачу в канал байта основного состояния с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО, СБОЙ В УСТРОЙСТВЕ.

Указатель СБОЙ В ОБОРУДОВАНИИ фиксируется тогда, когда в процессе выполнения операции выявляется наличие сбоя в работе логических схем устройства. Условия формирования указателя:

- обнаружение ошибки при перфорации карты,
- выявление ошибки по четности в информации, считываемой из буферной памяти устройства.

В обоих указанных случаях перфорация карты не прекращается и доводится до конца. После этого в канал передается байт основного состояния, содержащий указатели КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО, СБОЙ В УСТРОЙСТВЕ. При обнаружении ошибки перфорации неверно отперфорированная карта автоматически укладывается в карман брака.

12.3. ПЕЧАТАЮЩИЕ УСТРОЙСТВА

Вывод информации на печать в машинах Единой системы производится с помощью алфавитно-цифровых печатающих устройств ЕС-7030 (с механизмом АЦПУ-128-5) и ЕС-7032 (с меха-

низмом АЦПУ-128-6). Оба эти устройства имеют одинаковые характеристики и выполнены по одинаковым схемам; но за счет использования более совершенных конструкций, изменения компоновки и повышения надежности деталей для ЕС-7032 увеличена средняя наработка на отказ.

Печатающие механизмы рассматриваемых устройств подобны механизмам других печатающих устройств барабанного (роторного) типа, описанных, например в [9]. Дополнительно введено в основное управление перемещением бумаги от закольцованной четырехдорожечной перфоленты, что позволяет при печати делать пропуски до 64 строк. Скорость печати обеспечивается до 600—900 строк в минуту. В каждой строке печатается до 128 символов; на каждой позиции строки может быть отпечатан любой из 83 основных символов или один служебный; набор символов охватывает русский и латинский алфавиты, цифры, знаки математических действий и специальные знаки. При печати может использоваться рулонная бумага с шириной ленты от 80 до 420 мм; если используется специальная бумага, то при печати возможно одновременное получение двух экземпляров выходного документа.

Для алфавитно-цифровых печатающих устройств ЕС-7030 и ЕС-7032 устройства управления, объединенные с ними в единую конструкцию, представляют собой сочетания блоков сопряжения с каналом и собственно устройств управления печатью, которые в машинах типа М-220, «Минск-22» относились к основному электронному оборудованию самих ЦВМ. Поэтому принципы организации печати в рассматриваемых устройствах те же, что и в устройствах типа АЦПУ-128-2. Однако печатающие устройства, используемые в машинах Единой системы, за счет некоторого усложнения их устройств управления обладают большей гибкостью при печати различных данных.

Общая схема алфавитно-цифрового печатающего устройства ЕС-7030 (ЕС-7032) приведена на рис. 12.5. На ней выделены: блок сопряжения с каналом (БСК), регистр состояния (РС), регистр уточненного состояния (РУС), дешифратор команд (ДШК), блок управления записью (БУЗ), буферный накопитель (БН), блок управления печатью (БУП), блок управления механизмом (БУМ) и печатающий механизм (ПМ).

Блок сопряжения с каналом обеспечивает все действия, выполнение которых необходимо при обмене информацией между

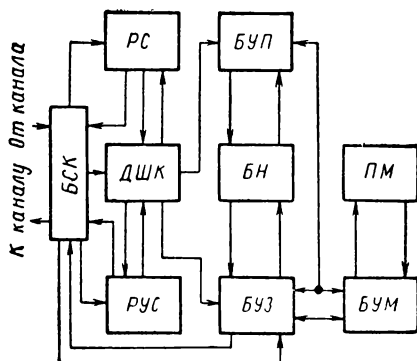


Рис. 12.5. Общая схема алфавитно-цифрового печатающего устройства

каналом и устройством через стандартный интерфейс ввода-вывода. В нем, в частности, формируются сигналы идентификации, производится контроль поступающей из канала информации по четности, формируются контрольные разряды для байт состояния, направляемых в канал, а также для байта адреса устройства. Если канал по запросу устройства не может принять от него байт состояния, то последний хранится в БСК. Информация, необходимая непосредственно для печати, из блока сопряжения направляется в БУЗ и далее в БН.

Дешифратор команд и регистры состояний как бы дополняют БСК для обеспечения системной работы устройства. В регистрах формируются и хранятся байты основного состояния (в РС) и уточненного состояния (в РУС). Дешифратор обеспечивает расшифровку байта команды и формирование начальных управляющих сигналов для БУ записью и печатью.

Блок управления печатью совместно с БН обеспечивает управление ПМ (через его БУ) при выводе на печать любой информации. Буферный накопитель представляет собой МОЗУ матричного типа емкостью 128 байт; в него записываются коды символов, печатаемых на одной строке бумажной ленты; время обращения к накопителю — 8 мкс. Блок управления печатью имеет в своем составе узлы, необходимые для управления работой БН и формирования сигналов, непосредственно управляющих работой электромагнитов печати печатающего механизма. В состав блока включен также узел транспорта бумаги, обеспечивающий перемещение бумажной ленты под пуансонами печати на заданное выполняемой командой количество интервалов (строк). Если перемещение производится с использованием закольцованной перфоленты, то узел транспорта бумаги обеспечивает сравнение кода, заданного выполняемой командой, с кодом, считываемым с перфоленты; при равенстве кодов перемещение бумажной ленты под пуансонами печати прекращается.

Блок управления записью служит для задания и поддержания двух основных режимов работы устройства: записи и печати. В режиме записи производятся прием байт данных, представляющих собой коды символов, из канала и запись их в БН. При этом прием байт данных может производиться как в мультиплексном, так и в монопольном режимах, что обеспечивает возможность подключения устройства как к мультиплексному каналу, так и к селекторному каналу. После записи информации в БН устройство переключается в режим печати; символы, коды которых размещены в БН, печатаются на одной строке бумажной ленты.

В процессе печати барабан печатающего механизма, составленный из 128 колес, на внешних ободах которых находятся литеры 84 символов, непрерывно вращается. За один оборот барабана, как правило, производится печать всех символов на

одной строке и подготовка к печати на очередной строке. При вращении барабана из БУМ в БУП посылаются сигналы, позволяющие сформировать код строки барабана, подходящей под пуансоны печати. С этим кодом последовательно сравниваются коды, записанные в БМ. Таким образом, за цикл печати на одной строке бумажной ленты производится $128 \times 84 = 10752$ обращения к БН.

После окончания цикла печати на одной строке бумажной ленты устройство готово к приему очередной порции данных из канала. Запись новых данных и перемещение бумажной ленты на очередную строку должны производиться во время свободного хода барабана печатающего механизма. Только в этом случае каждому обороту барабана будет соответствовать печать на одной строке бумажной ленты при работе устройства без остановок электродвигателя.

Печатающие устройства ЕС-7030 и ЕС-7032 рассчитаны на выполнение команд, представленных табл. 12.3, где разряды-модификаторы ММ могут принимать значения от 01 до 11 и определяют количество строк, на которое должна быть перемещена бумажная лента после печати. Разряды-модификаторы СССС принимают значения от 0001 до 1100 и определяют комбинацию на управляющей перфоленте, по которой осуществляется прогон (перемещение) бумажной ленты.

Т а б л и ц а 12.3

Команда	Разряды байта команды							
	0	1	2	3	4	5	6	7
ЗАПИСАТЬ С ПРОПУСКОМ ПОСЛЕ ПЕЧАТИ	0	0	0	М	М	0	0	1
ЗАПИСАТЬ С ПРОГОНОМ ПОСЛЕ ПЕЧАТИ	1	С	С	С	С	0	0	1
ЗАПИСАТЬ БЕЗ ПРОПУСКА	0	0	0	0	0	0	0	1
ПРОПУСК	0	0	0	М	М	0	1	1
ПРОГОН	1	С	С	С	С	0	1	1
УТОЧНИТЬ СОСТОЯНИЕ	0	0	0	0	0	1	0	0
ХОЛОСТОЙ ХОД	0	0	0	0	0	0	1	1
ПРОВЕРИТЬ ВВОД-ВЫВОД	0	0	0	0	0	0	0	0

По команде ЗАПИСАТЬ С ПРОПУСКОМ ПОСЛЕ ПЕЧАТИ печатающее устройство принимает данные из канала в БН; прием заканчивается либо по инициативе канала, либо по инициативе устройства (после заполнения всего накопителя). По окончании приема данных в канал посылается байт основного состояния с указателем КАНАЛ КОНЧИЛ, а устройство производит

печать символов на одной строке бумажной ленты и перемещение последней на число строк, заданное содержимым разрядов-модификаторов. В конце механического цикла, включающего перемещение бумажной ленты, в канал посылается байт основного состояния с указателем ВУ КОНЧИЛО. После приема этого байта каналом печатающее устройство может выполнять новую команду.

Команда ЗАПИСАТЬ С ПРОГОНОМ ПОСЛЕ ПЕЧАТИ выполняется аналогично команде записи с пропуском после печати. Отличие заключается только в том, что в механическом цикле после печати символов на одной строке перемещение (прогон) бумажной ленты производится до совпадения кода, заданного разрядами-модификаторами СССС, с одним из 12 кодов, нанесенных на перфоленту, управляющую транспортом бумаги.

Команда ЗАПИСАТЬ БЕЗ ПРОПУСКА выполняется аналогично команде записи с пропуском после печати. Отличие заключается только в том, что после печати символов, коды которых приняты в буферный накопитель, перемещение бумажной ленты не производится. Таким образом, при выполнении данной команды механический цикл оказывается сокращенным.

Команды ПРОПУСК И ПРОГОН являются командами управления с немедленным ответом. Это означает, что при получении любой из этих команд устройство отвечает каналу в последовательности сигналов начальной выборки байтом основного состояния, содержащим указатель КАНАЛ КОНЧИЛ. При выполнении команды ПРОПУСК производится только перемещение бумажной ленты на количество строк, заданное значением разрядов-модификаторов ММ. При выполнении команды ПРОГОН производится только прогон бумажной ленты при использовании управляющей перфоленты и в соответствии с кодом, содержащимся в разрядах-модификаторах СССС. После перемещения (прогона) бумажной ленты в канал посылается байт основного состояния с указателем ВУ КОНЧИЛО.

В устройстве ЕС-7032 предусмотрен особый случай выполнения команды ПРОГОН. Если код, содержащийся в разрядах-модификаторах СССС, совпадает с кодом перфоленты, уже подведенным к считывающему узлу, то прогон бумаги выполняется только в том случае, когда предыдущей командой, выполненной устройством, была команда ЗАПИСАТЬ БЕЗ ПРОПУСКА. В противном случае прогон бумаги не производится.

По команде ХОЛОСТОЙ ХОД в устройстве никаких действий не производится. Эта команда является командой с немедленным ответом. Поэтому в последовательности сигналов начальной выборки устройство отвечает на нее байтом основного состояния с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО.

Команда УТОЧНИТЬ СОСТОЯНИЕ используется для вызова в канал байта уточненного состояния. После передачи байта

уточненного состояния устройство посылает в канал байт основного состояния с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО.

По команде ПРОВЕРИТЬ ВВОД-ВЫВОД из устройства в канал посылается байт основного состояния; эта команда выполняется печатающими устройствами так же, как и устройствами, описанными выше. Необходимо отметить, что программист не может использовать код команды ПРОВЕРИТЬ ВВОД-ВЫВОД в своей канальной программе, так как эта команда является для канала внутренней, формируемой в определенных ситуациях.

Байт основного состояния передается устройством в канал в следующих случаях:

- в последовательности сигналов начальной выборки;
- по окончании приема данных;
- по окончании полного механического цикла устройства;
- при передаче в канал так называемого запомненного состояния;
- при переводе устройства из состояния НЕ ГОТОВО в состояние ГОТОВО.

Разряды 0, 1 и 2-й байта основного состояния не используются. Третий содержит указатель ЗАНЯТО, 4-й — указатель КАНАЛ КОНЧИЛ, 5-й — указатель ВУ КОНЧИЛО, 6-й — указатель СБОЙ В УСТРОЙСТВЕ, 7-й — указатель ОСОБЫЙ СЛУЧАЙ.

Указатель ЗАНЯТО выдается только при начальной выборке устройства и означает, что данное печатающее устройство не может выполнить команду вследствие того, что им выполняется ранее начатая операция или хранится условие прерывания. Отметим, что условие прерывания создается тогда, когда в устройстве сформированы указатели, но они еще не обработаны каналом.

Указатель КАНАЛ КОНЧИЛ свидетельствует о завершении передачи данных из канала в устройство, а указатель ВУ КОНЧИЛО — об окончании устройством выполнения команды и о его готовности принять новую команду.

Указатель СБОЙ В УСТРОЙСТВЕ свидетельствует о том, что сформированы значения одного или нескольких указателей для байта уточненного состояния. Практически этот указатель свидетельствует о выявлении ошибки или сбоя в работе устройства.

Указатель ОСОБЫЙ СЛУЧАЙ устанавливается тогда, когда опознается 12-й код управляющей перфоленты во время помещения бумажной ленты по команде ЗАПИСАТЬ С ПРОПУСКОМ или по команде ПРОПУСК. Он передается в канал вместе с указателем ВУ КОНЧИЛО.

Байт уточненного состояния печатающих устройств содержит информацию о сбоях и ошибках. Он выдается в канал по команде УТОЧНИТЬ СОСТОЯНИЕ. Разряды 4, 5 и 6-й байта

не используются; в 0-м разряде размещается указатель КОМАНДА ОТВЕРГНУТА, в 1-м — указатель ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО, во 2-м — указатель ОШИБКА НА ШИН-К, в 3-м — указатель СВОЙ В ОБОРУДОВАНИИ, в 7-м — указатель КАНАЛ 9.

Указатели КОМАНДА ОТВЕРГНУТА и ОШИБКА НА ШИН-К формируются так же, как и при работе устройств, описанных в пп. 12.1, 12.2.

Указатель ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО формируется при переводе устройства в состояние неготовности по одной из следующих причин:

- была нажата клавиша «СТОП» на пульте устройства;
- блок печатающего барабана не приведен в нормальное положение;
- произошел обрыв бумажной ленты или она кончилась;
- произошел отказ в системе питания;
- произошел обрыв перфоленты, управляющей транспортом бумаги;
- приводной двигатель не набрал нужного числа оборотов.

Этот указатель сбрасывается только при переводе устройства в состояние ГОТОВО.

Указатель СВОЙ В ОБОРУДОВАНИИ устанавливается по одной из двух причин: а) выявление ошибки по четности в буферном накопителе, б) поступление в устройство кода, не отвечающего ни одному из принятых символов. При поступлении такого недействительного кода устройство ЕС-7030 печатает специальный символ, а устройство ЕС-7032 делает пробел.

Указатель КАНАЛ 9 означает, что был опознан 9-й код управляющей перфоленты во время перемещения бумажной ленты по команде ЗАПИСАТЬ С ПРОПУСКОМ или по команде ПРОПУСК.

12.4. ПИШУЩАЯ МАШИНА С БЛОКОМ УПРАВЛЕНИЯ

Устройство, включающее в свой состав пишущую машину «Консул-260» и блок управления с пультом, или устройство ЕС-7070, предназначено для организации непосредственного обмена информацией между оператором и вычислительной машиной через стандартный интерфейс ввода-вывода. Пользуясь клавиатурой пишущей машины, оператор может вводить данные в ЭВМ и одновременно печатать их на бумаге; при приеме информации из ЭВМ данные печатаются на бумаге. Устройство может работать как в мультиплексном, так и в монопольном режимах и подключаться к мультиплексному или селекторному каналам моделей ЭВМ Единой системы через стандартный интерфейс ввода-вывода.

В устройстве используется печатающий механизм последовательного действия типа «Консул-260» со скоростью печати до

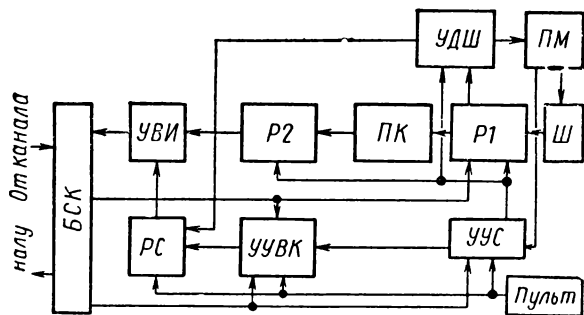


Рис. 12.6. Схема управления пишущей машиной при ее подключении к каналу

10 знаков в секунду. На строке бумажной ленты (листа бумаги) печатается до 106 знаков; набор возможных знаков включает 92 различных знака (символа). Обмен информацией с каналом осуществляется в коде ДКОИ. Любому знаку пишущей машины отвечает 8-разрядный байт, направляемый в канал, или принимаемый из канала. В распоряжении оператора кроме клавиатуры пишущей машины имеется пульт с набором кнопок (клавиш) для формирования некоторых начальных управляющих сигналов.

Общая схема рассматриваемого устройства приведена на рис. 12.6, где выделены: блок сопряжения с каналом (БСК), узел выдачи информации (УВИ), регистр состояния (РС), узел управления возвратом каретки (УУВК), узел управления синхронизацией (УУС), преобразователь кодов (ПК), два однобайтных регистра P1 и P2, узел дешифратора (УДШ), шифратор (Ш), пишущая машина (ПМ) и пульт оператора.

Блок сопряжения с каналом, как и в любых других УВУ, обеспечивает подключение устройства к интерфейсу и обмен с каналом управляющими последовательностями сигналов и информацией. Он содержит в своем составе необходимые узлы формирования управляющих сигналов, регистры, дешифратор адреса и дешифратор команд, узел контроля передаваемой и принимаемой информации по четности. В блоке сопряжения формируются некоторые сигналы, непосредственно воздействующие на механизмы пишущей машины (например, сигнал блокировки клавиатуры и т. п.)

При наборе информации на клавиатуре пишущей машины комбинатор последней - представляет каждый набираемый символ в коде КОИ-7. Шифратор преобразует комбинации, поступающие от пишущей машины, в 8-разрядные комбинации кода КОИ-8, запоминаемые в регистре P1. Регистр P1 является базовым для узла дешифратора, который обеспечивает преобразование 8-разрядных комбинаций КОИ-8 и ДКОИ в одиночные сигналы, непосредственно управляющие электромагнитами печати пишущей

машины; в УДШ, в случае необходимости, вырабатывается также сигнал смены регистра пишущей машины. Итак, при наборе информации на клавиатуре пишущей машины регистр Р1 совместно с УДШ обеспечивает ее печать; отметим, что при печати возможно получение пяти копий.

При выполнении команд считывания информации, т. е. при выдаче данных из устройства в канал, комбинации кода КОИ-8 из регистра Р1 передаются в преобразователь кодов, на выходах которого образуются комбинации кода ДКОИ. Хранение байт данных в коде ДКОИ обеспечивается регистром Р2. Из него байты данных через узел выдачи информации, содержащей группы вентилей и логических схем, передаются в БСК и далее в канал. Наличие в составе блока управления двух информационных регистров обеспечивает возможность исправления символа, неправильно набранного на клавиатуре пишущей машины. При нажатии оператором клавиши аннулирования набора в УУС и УУВК поступает сигнал, вызывающий в конечном итоге возврат каретки на одну позицию и сброс регистра Р1 в нулевое состояние; теперь можно печатать требуемый символ (вместо неправильно набранного).

Узел управления синхронизацией по сигналам от ПМ вырабатывает наборы сигналов, обеспечивающих синхронизацию работы элементов механических узлов ПМ с работой электронных узлов блока управления. Сигналы запуска УУС поступают также из блока сопряжения с каналом и пульта оператора, на котором кроме кнопок управления располагаются элементы световой и звуковой сигнализации.

Рассматриваемое устройство рассчитано на выполнение команд, приведенных в табл. 12.4, где использованы следующие обозначения; АВК — автоматический возврат каретки, ЗВК — запрет возврата каретки.

Т а б л и ц а 12.4

Команда	Разряды байта команды							
	0	1	2	3	4	5	6	7
ЗАПИСАТЬ С АВК	0	0	0	0	1	0	0	1
ЗАПИСАТЬ С ЗВК	0	0	0	0	0	0	0	1
ЧИТАТЬ	0	0	0	0	1	0	1	0
УПРАВЛЯТЬ ЗВУКОВОЙ СИГНАЛИ- ЗАЦИЕЙ	0	0	0	0	1	0	1	1
ХОЛОСТОЙ ХОД	0	0	0	0	0	0	1	1
УТОЧНИТЬ СОСТОЯНИЕ	0	0	0	0	0	1	0	0
ПРОВЕРИТЬ ВВОД-ВЫВОД	0	0	0	0	0	0	0	0

По командам записи устройство принимает данные из канала и печатает их на бумаге по одному символу. В конце сообщения команда ЗАПИСАТЬ С АВК вызывает автоматический возврат каретки и переход к последующей строке, т. е. перевод бумаги на один интервал; по команде ЗАПИСАТЬ С ЗВК эти действия не производятся.

При приеме кода недопустимого символа устройство продолжает выполнение операции записи информации, а вместо недопустимого символа печатается один из символов, имеющих на клавиатуре ПМ.

Обычно команды записи заканчиваются каналом по счетчику; в этом случае устройство формирует указатель КАНАЛ КОНЧИЛ. После возврата каретки формируется указатель ВУ КОНЧИЛО. При нормальном выполнении команды ЗАПИСАТЬ С ЗВК указатели КАНАЛ КОНЧИЛ и ВУ КОНЧИЛО формируются одновременно.

Если во время выполнения команды записи устройство переводится в состояние НЕ ГОТОВО вследствие того, что не может быть завершен механический цикл, то выполнение операции прекращается, устройство формирует указатели КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО, СВОЙ В УСТРОЙСТВЕ для байта основного состояния и указатель ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО для байта уточненного состояния. Если переход в состояние НЕ ГОТОВО произошел после передачи в канал указателя КАНАЛ КОНЧИЛ, то устройство посылает в канал запрос на-передачу байта основного состояния с указателями ВУ КОНЧИЛО и СВОЙ В УСТРОЙСТВЕ.

Команда ЧИТАТЬ, по которой производится передача информации, набираемой на клавиатуре ПМ, в канал ЭВМ, может быть вызвана только путем действий оператора на пульте устройства. Эта команда запрашивается из канала нажатием кнопок «СД» (служебное действие) и «ВН» (внимание) на пульте. Поступление из канала команды ЧИТАТЬ снимает блокировку с клавиатуры пишущей машины; на пульте включается лампочка транспоранта «ЧИТАТЬ», сигнализирующая оператору о том, что можно набирать символы на клавиатуре ПМ.

При нажатии первой клавиши байт КОИ-8, отвечающий соответствующему символу, вводится в регистр Р1; одновременно производится печать символа. При нажатии второй клавиши байт первого символа, преобразованный в код ДКОИ, вводится в регистр Р2, а байт кода КОИ-8 второго символа — в регистр Р1; второй символ печатается на бумаге, а устройство посылает в канал запрос на передачу байта первого символа. До удовлетворения запроса клавиатура ПМ блокируется. После передачи байта возможно нажатие третьей клавиши на клавиатуре ПМ и т. д. Такой порядок работы сохраняется до окончания выполнения команды.

Выполнение команды **ЧИТАТЬ** заканчивается в следующих трех случаях:

— оператор нажимает клавиши «СД» и «КТ» (конец текста); при этом в канал передается последний символ и запрос на передачу байта основного состояния с указателем **КАНАЛ КОНЧИЛ**;

— оператор нажимает клавиши «СД» и «АН» (аннулирование), что означает ошибку в сообщении и необходимость повторения операции; в канал посылается байт основного состояния с указателями **КАНАЛ КОНЧИЛ** и **ОСОБЫЙ СЛУЧАЙ**; обычно в устройство после этого повторно выдается команда **ЧИТАТЬ**, по которой снимается блокировка клавиатуры ПМ, и оператор может повторно ввести в ЭВМ сообщение;

— канал прекращает выполнение операции по счетчику.

В любом случае окончания выполнения операции чтения производится блокировка клавиатуры ПМ, выключение лампочки транспаранта «ЧИТАТЬ» и возврат каретки; в канал посылается указатель **КАНАЛ КОНЧИЛ**. После возврата каретки в исходное состояние и перевода строки на один интервал в канал посылается запрос на передачу байта основного состояния с указателем **ВУ КОНЧИЛО**.

Команда **УПРАВЛЯТЬ ЗВУКОВОЙ СИГНАЛИЗАЦИЕЙ** принимается устройством даже тогда, когда оно находится в состоянии **НЕ ГОТОВО**. По этой команде на три—пять секунд включается звуковая сигнализация на пульте оператора и в последовательности начальной выборки в канал передается байт состояния устройства с указателями **КАНАЛ КОНЧИЛ**, **ВУ КОНЧИЛО**.

Команды **УТОЧНИТЬ СОСТОЯНИЕ**, **ХОЛОСТОЙ ХОД** и **ПРОВЕРИТЬ ВВОД-ВЫВОД** имеют то же значение, что и для устройств, рассмотренных в пп. 12.1—12.3. Выполнение этих команд в рассматриваемом устройстве не имеет никаких принципиальных особенностей.

Байт основного состояния устройства с пишущей машиной выдается в канал в тех же случаях, что и байт основного состояния печатающих устройств, например. Нулевой разряд байта отведен для размещения указателя **ВНИМАНИЕ**, 3-й — указателя **ЗАНЯТО**, 4-й — указателя **КАНАЛ КОНЧИЛ**, 5-й — указателя **ВУ КОНЧИЛО**, 6-й — указателя **СБОЙ В УСТРОЙСТВЕ**, 7-й — указателя **ОСОБЫЙ СЛУЧАЙ**; первый и второй разряды байта основного состояния не используются.

Указатель **ВНИМАНИЕ** формируется тогда, когда на пульте нажата кнопка «ВН» и соблюдаются следующие условия:

— устройство не занято в последовательности сигналов начальной выборки;

— устройство не занято передачей данных;

— не выполняется цепочка команд.

Если эти условия не выполняются, то состояние, вызываемое нажатием кнопки «ВН», сохраняется до тех пор, пока в устройстве не будет сформирован указатель **ВНИМАНИЕ**.

После того, как в байте состояния устанавливается этот указатель, устройство становится занятым и не выполняет никакие новые команды (за исключением команды ПРОВЕРИТЬ ВВОД-ВЫВОД); в последовательности начальной выборки устройства выдается байт основного состояния с указателями ВНИМАНИЕ и ЗАНЯТО.

Указатель ЗАНЯТО формируется в следующих случаях:

- после завершения последовательности сигналов начальной выборки для команд чтения, записи и уточнения состояния;
- при формировании указателя ВНИМАНИЕ;
- при переводе устройства из состояния НЕ ГОТОВО в состояние ГОТОВО;

- при хранении в устройстве байта состояния;
- после приема каналом указателя ВУ КОНЧИЛО;
- при системном и селективном сбросах; этот указатель свидетельствует о занятости устройства, невозможности выполнения еще одной задаваемой операции.

Указатель КАНАЛ КОНЧИЛ показывает каналу, что устройство может быть отключено от интерфейса. Указатель ВУ КОНЧИЛО показывает, что устройство завершило выполнение операции ввода-вывода или что оно переведено из состояния НЕ ГОТОВО в состояние ГОТОВО.

Указатель СБОЙ В УСТРОЙСТВЕ устанавливается при формировании хотя бы одного указателя в байте уточненного состояния и свидетельствует о наличии сбоя или ошибки.

Указатель ОСОБЫЙ СЛУЧАЙ формируется при нажатии клавиш «СД» и «АН»; он посылается в канал всегда одновременно с указателем КАНАЛ КОНЧИЛ.

Байт уточненного состояния устройства ЕС-7070 выполняет те же функции, что и применительно к другим ранее рассмотренным устройствам. Его нулевой разряд отведен под указатель КОМАНДА ОТВЕРГНУТА, 1-й — указатель ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО, 2-й — указатель ОШИБКА НА ШИН-К, 3-й — указатель СБОЙ В ОБОРУДОВАНИИ; 4—7-й разряды байта уточненного состояния не используются.

Указатель КОМАНДА ОТВЕРГНУТА формируется при поступлении в устройство невыполнимого кода команды.

Указатель ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО свидетельствует о неготовности устройства к работе; он формируется в следующих случаях:

- нажата клавиша «НЕ ГОТОВО» на пульте оператора;
- устройство не может завершить механический цикл вследствие зависания рычага в ПМ;
- при подаче соответствующего сигнала возврат каретки не производится.

Указатель ОШИБКА НА ШИН-К формируется тогда, когда принимаемый из канала байт имеет ошибку по четности.

Указатель СБОЙ В ОБОРУДОВАНИИ свидетельствует о том, что схемами контроля выявлен сбой в работе механических или электрических узлов устройства.

12.5. ВЫНОСНЫЕ ПУЛЬТЫ С ГРУППОВЫМ УСТРОЙСТВОМ УПРАВЛЕНИЯ

Для обслуживания абонентов, располагающихся на удалении нескольких сот метров от машинного зала, в моделях ЭВМ Единой системы предусмотрено использование выносных пультов ЕС-7066, обеспечивающих ввод и вывод алфавитно-цифровой информации. Эти выносные пульты могут подключаться к машине только через групповое устройство управления ЕС-7566. Комплекс ЕС-7906, состоящий из группового устройства управления (группового УВУ) и набора связанных с ним выносных пультов, может работать как в мультиплексном, так и в монопольном режимах и подключаться к мультиплексному или селекторному каналам моделей ЭВМ Единой системы посредством стандартного интерфейса ввода-вывода.

Каждый выносной пульт конструктивно автономен и рассчитан на индивидуальное использование. Групповое УВУ также выполнено в виде отдельной конструктивной единицы, которая может быть расположена в помещении, непосредственно примыкающем к машинному залу; это определяется возможностями стандартного интерфейса ввода-вывода. Отдельной конструктивной частью комплекса ЕС-7906 является пишущая машина типа «Консул-260», выполняющая функции печатающего устройства; в случае необходимости информация, вводимая с выносных пультов в машину или выводимая на выносные пульты, может быть отпечатана на белой бумаге; вывод информации на печать требуется, например, при документировании данных, поскольку информация на экранах выносных пультов не сохраняется на длительное время.

К одному групповому УВУ ЕС-7566 можно подключить до 16 выносных пультов ЕС-7066; при этом цепей подключения всего 4. К каждой цепи подключается либо один выносной пульт (ВП), либо два ВП, либо четыре ВП; соответственно к УВУ подключается либо четыре, либо восемь, либо 16 выносных пультов. Количество подключаемых к групповому УВУ выносных пультов влияет на реальные возможности каждого ВП по вводу и выводу информации; это объясняется тем, что общие возможности УВУ меньше суммарных потенциальных возможностей 16 выносных пультов.

Общая схема комплекса ЕС-7906 при максимальном наборе выносных пультов приведена на рис. 12.7. На этой схеме в составе УВУ выделены только блок сопряжения с каналом (БСК) и блок управления (БУ). Такое разделение УВУ характеризует только независимость схем управления пишущей машиной (ПМ) и схем управления работой выносных пультов ЕС-7066.

Собственно ВП представляет собой конструкцию, устанавливаемую на любом канцелярском столе и имеющую массу около 50 кг. Экранную часть пульта составляет экран прямоугольной формы электронно-лучевой трубки 43ЛМ1И с рабочим полем 320 × 180 мм. В составе выносного пульта имеется клавиатура с 49 алфавитно-цифровыми клавишами и 20 клавишами управления; это позволяет осуществлять набор любого из 128 символов и производить необходимые действия по вводу и выводу информации, а также по исправлению ошибочно набранных символов. В корпусе выносного пульта размещаются схемы управления электронно-лучевой трубкой и схемы связи с УВУ.

Связь выносного пульта с УВУ осуществляется по специальному кабелю длиной до 500 м; вся информация передается во внутреннем коде устройства ЕС-7566. Важное место в схемах управления выносного пульта занимает знакогенератор, который преобразует код символа в последовательность сигналов, обеспечивающих векторное начертание соответствующего знака на экране электронно-лучевой трубки. Вся информация, набираемая оператором на клавиатуре, а также поступающая из группового УВУ, обязательно индицируется.

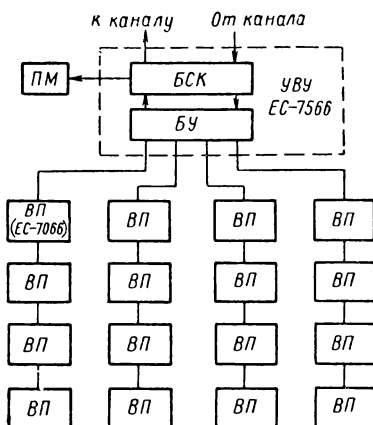


Рис. 12.7. Общая схема комплекса ЕС-7906

Технические характеристики выносного пульта

Максимальный объем информации, воспроизводимой (индицируемой) на экране, знаки	960
Количество строк	12 или 6
Количество знаков в строке	80 или 40
Частота воспроизведения информации на экране, Гц	50
Количество различных знаков, воспроизводимых на экране	128
Размеры знаков, мм:	
при размещении на экране 960 знаков	3,5 × 2,5
при размещении на экране 480 или 240 знаков	7 × 5

Количество строк на экране выносного пульта, количество знаков в строках и размеры воспроизводимых знаков определяются в конечном итоге возможностями группового УВУ ЕС-7566 и количеством подключенных к нему выносных пультов. Полностью возможности выносного пульта ЕС-7066 используются только тогда, когда к групповому УВУ подключается четыре ВП; это объясняется тем, что групповое УВУ рассчитано на суммарную емкость экранов выносных пультов, определяемую 3840 знаками.

Для обеспечения индикации знаков на экранах выносных пультов в состав группового УВУ включено запоминающее устройство с накопителем на магнитных сердечниках емкостью 4096 байт и временем обращения порядка 2 мкс. Каждому выносному пульту выделяется отдельная зона ЗУ; количество ячеек, включаемых в состав зоны, определяется числом подключаемых к УВУ выносных пультов. Использование буферной памяти выносных пультов необходимо не только для целей накопления информации при обмене ею между ВП и каналом (ЭВМ), но и для обеспечения работы схем управления ЭЛТ, информация на экранах которых должна восстанавливаться с частотой 50 Гц.

Кроме буферной памяти выносных пультов в составе группового УВУ имеется буферная память устройства печати, в качестве которого используется пишущая машина типа «Консул-260». Поскольку печать производится со скоростями до 10 знаков в секунду, то для хранения кодов печатаемых знаков оказалось достаточным использовать периодическое запоминающее устройство в виде магнитострикционной линии задержки. Буферная память устройства печати рассчитана на одновременное хранение кодов до 1000 знаков, выводимых на печать.

Внутренние и внешние связи группового устройства управления выносными пультами показаны на рис. 12.8, где выделены его основные части: кодопреобразователь (КП), буферная память выносных пультов (БП ВП) со входным и выходным регистрами (соответственно регистры Р1 и Р2), блок связи с выносными пультами (БС ВП), блок управления (БУ) и буферная память устройства печати (БП УП); в выносном пульте выделены индикатор и клавиатура.

Обмен информацией между выносными пультами и машиной может производиться как от ВП к машине, так и от машины к ВП. Первый случай есть не что иное как ввод информации с ВП; считается, что в этом случае реализуется режим чтения. Второй случай представляет вывод информации из ЭВМ на выносной пульт; считается, что в этом случае реализуется режим записи.

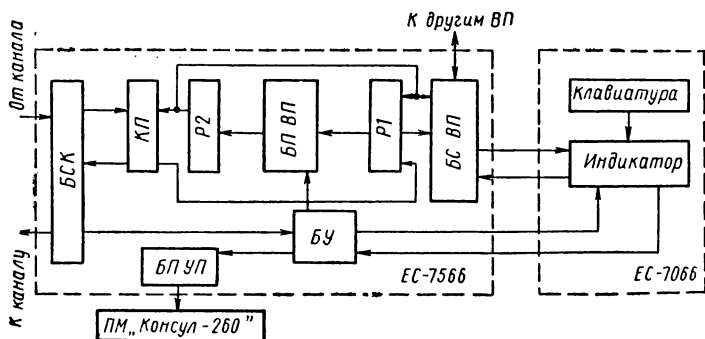


Рис. 12.8. Внутренние и внешние связи группового УВУ

В режиме ввода информации (режиме чтения) оператор набирает на клавиатуре выносного пульта необходимую информацию, которая записывается в отведенную для данного ВП зону буферной памяти и отображается на экране индикатора. В процессе набора оператор может редактировать сообщение, т. е. стирать неправильно набранные знаки и на их место записывать действительно необходимые; отображение знака на экране обеспечивается только после записи его кода в буферную память. После окончания набора оператор, работая на клавиатуре пульта, посылает в УВУ специальное сообщение, по которому организуется подключение УВУ к каналу и передача в ЭВМ набранной на пульте информации с осуществлением необходимой ее перекодировки.

Если возникает необходимость документирования сообщений, то путем нажатия специальной функциональной клавиши на клавиатуре ВП оператор обеспечивает передачу ранее набранной информации в буферную память устройства печати; далее осуществляется вывод информации на печать посредством пишущей машины «Консул-260».

В режиме вывода информации из ЭВМ (режиме записи) необходимый блок данных по запросу оператора ВП выдается из ЗУ машины и заносится в зону буферной памяти, выделенной для данного ВП, или в БП УП. В первом случае при циклическом опросе буферной памяти выносных пультов на экранах ВП индицируется полученная для них информация; во втором случае информация, полученная из машины, только печатается и на индикацию не выводится.

12.6. ЛОГИЧЕСКИЙ РЕТРАНСЛЯТОР

Стандартный интерфейс ввода-вывода, разработанный для ЭВМ Единой системы, реально обеспечивает подключение к каналу только восьми УВУ различных типов. Для расширения возможностей интерфейса ввода-вывода используется специальный логический ретранслятор, или устройство ЕС-4080. Это устройство может работать как в мультиплексном, так и в монопольном режиме и подключаться к мультиплексному или селекторному каналу.

Рассматриваемое устройство включается в основной интерфейс ввода-вывода на месте УВУ с наивысшим приоритетом. Оно образует дополнительный, или вспомогательный, интерфейс, как это показано на рис. 12.9. В процессе работы устройство ЕС-4080 ретранслирует все сигналы интерфейса во вспомогательный интерфейс без каких бы то ни было преобразований; оно не управляется со стороны канала и не требует никакого математического обеспечения. Все это определяет простоту его использования при необходимости увеличения количества УВУ на одном интерфейсе ввода-вывода. Отметим, что устройство ЕС-4080 может использоваться также для увеличения длины линий интерфейса до 100—120 м.

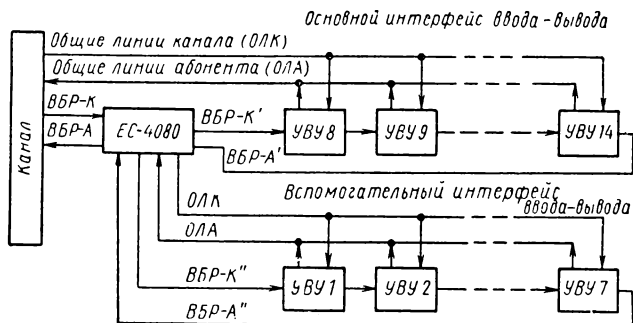


Рис. 12.9. Расширение возможностей интерфейса ввода-вывода при использовании логического ретранслятора

Если логический ретранслятор ЕС-4080 подключается к стандартному интерфейсу ввода-вывода на месте УВУ с наивысшим приоритетом, то основной и вспомогательный интерфейсы можно нагрузить пятнадцатью УВУ; при этом вспомогательный интерфейс отвечает всем требованиям, предъявляемым к стандартному интерфейсу ввода-вывода. К образованному вспомогательному интерфейсу можно подключить до восьми УВУ. Количество УВУ, подключаемых к вспомогательному интерфейсу, определяется тем, что общее число УВУ, установленных в последовательной цепи сигналов ВБР-К, ВБР-А с учетом ЕС-4080 не должно быть больше восьми.

При использовании устройства ЕС-4080 обеспечивается:

- работа канала с устройствами управления внешних устройств (УВУ), подключенными к основному или вспомогательному интерфейсу, без предъявления дополнительных требований к физической реализации интерфейса ввода-вывода;

- подключение УВУ различных типов с любыми адресами (в пределах 256) к вспомогательному (дополнительному) интерфейсу;

- представление высшего приоритета устройствам, подключенным к вспомогательному интерфейсу; устройства, подключаемые к основному интерфейсу при наличии и использовании вспомогательного интерфейса всегда имеют более низкий приоритет по обслуживанию со стороны канала.

Величина временной задержки, вносимой устройством ЕС-4080 для сигнала ВБР-К, отвечает времени прохождения этого сигнала через схемы одного УВУ. При отключении устройства от источника питания прекращается работа только вспомогательного интерфейса, т. е. УВУ, подключенные к этому интерфейсу, становятся недоступными для канала; основной интерфейс ввода-вывода в этом случае обеспечивает нормальную работу подключенных к нему УВУ с каналом, т. е. отключение ЕС-4080 от источника питания не влияет на работу основного интерфейса ввода-вывода.

Схема, поясняющая подключение логического ретранслятора к каналу и его основной состав, приведена на рис. 12.10. Его основу составляют усилители интерфейсных сигналов и группы логических элементов; на рисунке выделены только те логические элементы, которые необходимы для пояснения работы рассматриваемого устройства в различных возможных ситуациях. Под шинами 1 и 3 понимаются линии канала (это ШИН-К, УПР-К, БЛК-К, ИНФ-К, РАБ-К, РВБ-К, ИЗМ-К, СМС-К, АДР-К); под шинами 2 и 4 понимаются общие линии абонента (это ШИН-А, РАБ-А, ИЗМ-А, АДР-А, УПР-А, ИНФ-А).

Логический ретранслятор ЕС-4080 подключается к линиям стандартного интерфейса ввода-вывода аналогично УВУ. Все связи с каналом (за исключением линии по передаче сигнала ВБР-К, обеспечивающего установление приоритета УВУ по их обслуживанию со стороны канала) осуществляются по общим шинам, так что каждый сигнал, поступающий из канала, доступен любому УВУ, подключенному к основному или вспомогательному интерфейсу. Наличие фактически двух интерфейсов при использовании устройства ЕС-4080 не противоречит одному из главных условий обеспечения стандартного сопряжения УВУ с каналами: логической связи с каналом в любой момент времени только одного УВУ.

Выбор устройства управления внешним устройством, или УВУ, со стороны канала всегда производится сигналом ВБР-К, который последовательно проходит через все устройства управления, подключенные как к основному, так и к вспомогательному интерфейсу ввода-вывода. Это дает возможность всем УВУ в каждой из двух групп (относящихся к основному и вспомогательному интерфейсам) поочередно реагировать на сигналы, поступающие из канала. Приоритетность УВУ по группам устанавливается только при наличии заявок на обслуживание со стороны УВУ, подключенных к вспомогательному интерфейсу.

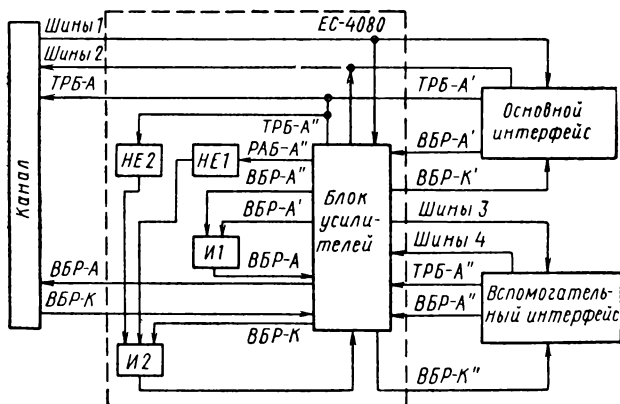


Рис. 12.10. Организация логического ретранслятора

Для обеспечения работы канала с двумя интерфейсами ввода-вывода логический ретранслятор управляет коммутацией сигналов ВБР-К и ВБР-А в соответствии с особенностями всех возможных на практике ситуаций. Таких ситуаций (случаев) четыре.

Рассмотрим особенности работы ЕС-4080 в каждой из них.

1. Канал выдает сигнал ВБР-К; ни одно из УВУ, подключенных к вспомогательному интерфейсу, не выдало до этого запрос на обслуживание, т. е. на линиях ТРБ-А" и РАБ-А" сигналы отсутствуют. В этом случае сигнал ВБР-К, поступающий из канала, проходит через схемы блока усилителей, схему совпадений И2, через разветвляющуюся цепочку усилителей и одновременно выдается в основной и вспомогательный интерфейсы. Логическую связь с каналом устанавливает то УВУ, подключенное к основному или вспомогательному интерфейсу, которое опознает свой адрес.

2. Канал выдает сигнал ВБР-К, но уже действуют сигналы ТРБ-А" или РАБ-А", свидетельствующие о наличии запроса на обслуживание УВУ, подключенного к вспомогательному интерфейсу. В этом случае сигнал ВБР-К, поступивший из канала в блок усилителей, в виде сигнала ВБР-К" проходит только во вспомогательный интерфейс, так как схема совпадений И2 оказывается закрытой по одному из входов сигналом, поступившим с выхода НЕ1 или НЕ2. Логическая связь с каналом устанавливается одним из УВУ, подключенным к вспомогательному интерфейсу.

Таким образом, УВУ, подключенные к вспомогательному интерфейсу, имеют высший приоритет по отношению к УВУ, подключенным к основному интерфейсу.

3. Канал выдает сигнал ВБР-К, который ретранслируется в оба интерфейса, и, пройдя по кольцевым линиям интерфейсов, возвращается в устройство ЕС-4080 в виде сигналов ВБР-А' и ВБР-А". Эти сигналы, пройдя через блок усилителей, поступают на схему совпадений И1. Сигнал ВБР-А, образующийся на выходе схемы И1, через блок усилителей поступает в канал; это означает, что ни одно из УВУ не опознало свой адрес и логически не подключилось к каналу через интерфейс для обмена информацией.

4. Устройства управления, подключенные как к основному, так и к вспомогательному интерфейсу сформировали запросы на обслуживание; действуют одновременно сигналы ТРБ-А' и ТРБ-А". В этом случае канал начинает обслуживать УВУ, начиная с устройств, подключенных к вспомогательному интерфейсу. Это объясняется тем, что сигнал ВБР-К не пройдет через схему И2, так как она закрыта сигналом, поступающим с выхода НЕ2; он ретранслируется только во вспомогательный интерфейс. Таким образом, и в этом случае УВУ, подключенные к вспомогательному интерфейсу, имеют высший приоритет по отношению к УВУ, подключенным к основному интерфейсу ввода-вывода.

12.7. УСТРОЙСТВА ПОДГОТОВКИ ДАННЫХ

Ввод информации в машины Единой системы осуществляется в основном с перфокарт и перфолент. Для нанесения кодов на эти носители в виде систем пробивок используются устройства подготовки данных на перфокартах и устройства подготовки данных на перфолентах.

Устройство подготовки данных на перфокартах. Нанесение информации на стандартные 80-колонковые перфокарты в виде системы пробивок по перфокартному коду КПК-12 производится с помощью устройства ЕС-9012. Это устройство обеспечивает по-колонную набивку кодов 82 различных символов со скоростью до 25 колонок в секунду в режиме реперфорации; при ручном наборе скорость набивки кодов определяется навыками работы оператора; нанесение информации на перфокарты контролируется путем сравнения набиваемых кодов с исходными. Таким образом ЕС-9012 обеспечивает реализацию функций перфоратора с клавишным устройством и контрольно-считывающего устройства.

В состав ЕС-9012 входят: блок клавиатуры, перфоратор, механизм транспорта перфокарт, блок считывания информации, блок контроля, блок управления и некоторые другие узлы. Блок клавиатуры содержит набор клавиш и связанные с ними шифраторы кодов; если нажата некоторая клавиша, то при подаче на вход шифратора исполнительного сигнала на его выходах образуется код соответствующего символа; этот код подается в перфоратор, а также в блок контроля.

Собственно перфоратор выполнен по обычной схеме с пробивными электромагнитами, управляющими движением пуансонов. Дополнительно в состав перфоратора включены датчики, формирующие код в соответствии с движением пуансонов, т. е. код, действительно набиваемый на перфокарте. Этот код подается в блок контроля для сравнения с исходным кодом, подаваемым из блока клавиатуры при ручном наборе или из блока считывания при реперфорации (дублировании) карт.

Блок считывания включает в свой состав набор фотодиодов, усилителей и источник света. При перемещении перфокарты под фотодиодами обеспечивается считывание кодов, направляемых в перфоратор и блок контроля, основу которого составляют регистры и поразрядная схема сравнения. При несовпадении исходного кода и кода, поступающего от датчиков перфоратора, хотя бы по одной позиции формируется сигнал, по которому блок управления обеспечивает останов устройства и сигнализирует оператору о наличии неисправности.

Механизм транспорта перфокарт осуществляет перемещение носителей исходной информации от подающего кармана до одного из двух приемных карманов. Перемещение перфокарт производится в соответствии с заданным режимом работы устройства. Каждый из карманов устройства имеет емкость до 500 карт, причем во вто-

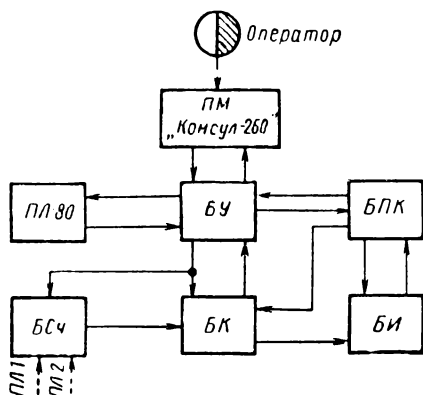


Рис. 12.11. Схема устройства подготовки данных на перфокартах

подготовленных лент, а также реперфорацию лент. Общий состав устройства показан на схеме (рис. 12.11).

Набор символов, коды которых набиваются на перфоленге, осуществляется на клавиатуре пишущей машины (ПМ) «Консул-260». Набираемая информация обычно печатается на бланке; возможна также печать по информации, поступающей из блока считывания (БСч). Набивка кодов производится посредством типового ленточного перфоратора ПЛ-80, связанного с источниками информации, т. е. машиной «Консул-260» и блоком считывания, через общий для всего устройства блок управления.

Блок считывания включает в свой состав два лентопротяжных механизма и соответственно две схемы фотосчитывания информации с перфоленга (ПЛ1 и ПЛ2). При работе с двумя перфоленгами обеспечивается их синхронная протяжка под фотодиодными узлами. Информация, считываемая с перфоленга, направляется в блок контроля; при реперфорации лент необходимая информация из блока считывания в ПЛ-80 направляется через блоки контроля (БК) и управления (БУ).

Блок контроля служит для приема кодов символов от БСч и блока преобразования кодов (БПК), сравнения кодов, формирования сигналов о совпадении и несовпадении кодов, а также для передачи кодов через блок управления в ленточный перфоратор ПЛ-80. С блоками контроля и преобразования кодов связан блок индикации (БИ), обеспечивающий световую индикацию содержания различных регистров устройства, световую и звуковую индикацию сигналов ошибок. Схемы блока преобразования кодов служат для перехода от кода пишущей машины «Консул-260» к кодам обмена, принятым для машин Единой системы и используемым при нанесении информации на перфоленгу.

Блок управления устройства формирует управляющие сигналы и организует работу всех остальных частей устройства в различ-

ной приемный карман перфокарты укладываются в обратном порядке. На обратной стороне перфокарты перед ее укладкой в приемный карман печатаются порядковый номер карты и личная метка оператора.

Устройство подготовки данных на перфокартах. Нанесение информации на перфоленгу шириной 25,4 мм в виде системы пробивок по кодам обмена, принятым для ЕС ЭВМ, производится с помощью устройства ЕС-9020. Это устройство обеспечивает также контроль информации при ее считывании с уже

ных режимах, задаваемых с пульта оператора. В свой состав он включает узлы управления пишущей машиной и ленточным перфоратором, а также отдельные схемы выработки управляющих сигналов.

Основные режимы работы устройства:

— режим подготовки данных; набираемая на клавиатуре пишущей машины информация наносится в виде системы пробивок на перфоленду и печатается на бланке;

— режим реперфорации; по данным, считаемым с ПЛ1 или ПЛ2 в блоке считывания, производится перфорация ленты посредством механизма ПЛ-80;

— режим сравнения; производится сравнение кодов, считываемых с ПЛ1 и ПЛ2; при несовпадении кодов хотя бы по одной позиции формируется сигнал ошибки;

— режим сравнения с клавиатурой; данные, считываемые с ПЛ1 или ПЛ2 сравниваются с данными, набираемыми на клавиатуре пишущей машины;

— режим сравнения с реперфорацией; сравниваются коды, считываемые с ПЛ1 и ПЛ2, и при их совпадении они наносятся на третью перфоленду с помощью механизма ПЛ-80;

— режим распечатки; информация, считываемая с ПЛ1 или ПЛ2, печатается на бланке;

— режим сравнения с распечаткой; сравниваются коды, считываемые с ПЛ1 и ПЛ2, осуществляется реперфорация третьей ленты и печать данных на бланке.

Скорость обработки информации в режимах, связанных с работой пишущей машины — до 10 знаков в секунду; в остальных режимах — до 50 знаков в секунду.

МУЛЬТИПЛЕКСНЫЕ И СЕЛЕКТОРНЫЕ КАНАЛЫ

13.1. ОРГАНИЗАЦИЯ ПЕРЕДАЧ ИНФОРМАЦИИ

Во всех машинах Единой системы передача информации любого вида между оперативной (основной) памятью (ОП) и внешними устройствами (ВУ) осуществляется по схеме: оперативная память — канал — интерфейс — управление внешними устройствами — внешние устройства; очевидно, что информация может направляться как от ОП к ВУ, так и от ВУ к ОП. Центральный процессор (ЦП) организует обмен информацией между ОП и ВУ путем выдачи в канал соответствующих команд. Для обеспечения нормального протекания процесса решения задач центральный процессор получает информацию от каналов только в виде отдельных слов, характеризующих состояния каналов и ВУ.

Канал — это устройство, или комплекс аппаратуры, специально предназначенное для выполнения операций ввода и вывода информации и обеспечивающее все необходимые двухсторонние связи между ОП, ЦП и множеством используемых внешних устройств при стандартных формах всех передаваемых данных. Обычно данные между каналом и ОП передаются пословно, а между каналом и ВУ (ВУ) — побайтно. При большом наборе внешних устройств их обслуживание организуется с помощью нескольких каналов как однотипных, так и разнотипных; типы каналов в основном определяются режимами работы, непосредственно связанными с режимами передачи данных.

В машинах Единой системы используются селекторные и мультиплексные каналы. Средства любого из этих каналов, необходимые для выполнения отдельной операции ввода-вывода, т. е. для связи с одним ВУ, называются подканалом. Практически подканал представляет собой часть памяти канала, которая используется для размещения информации, необходимой для работы с данным ВУ. Очевидно, что возможность работы канала в монопольном (групповом) или мультиплексном режиме по передаче данных определяется количеством подканалов.

Селекторный канал имеет только один подканал и может работать только в групповом режиме. Он не может одновременно обслуживать несколько ВУ и поэтому участвует только в одной операции по передаче данных. Посредством селекторных каналов осу-

ществляется передача данных между ОП и достаточно быстродействующими устройствами типа внешних ЗУ на магнитных лентах, барабанах и дисках. Если селекторный канал занят обслуживанием некоторого устройства, то поступающая в это время новая команда на ввод или вывод начинает выполняться только после завершения исполнения текущей команды.

Мультиплексный канал имеет множество подканалов, определяемое емкостью его оперативной памяти, и может работать, как в мультиплексном, так и в групповом режиме; в любой момент времени он может переключаться с одного режима на другой. При мультиплексном режиме работы он может обеспечить одновременное (параллельное) выполнение по одной операции ввода-вывода для каждого подканала. При групповом режиме работает один подканал, который монополизировывает основные средства канала; остальные подканалы бездействуют. В состав мультиплексного канала могут включаться отдельные селекторные подканалы, как это сделано, например в ЕС-1050. Собственно мультиплексный канал обычно обеспечивает передачу данных между ОП и медленно действующими ВУ типа печатающих устройств, перфораторов, устройств считывания информации с перфокарт и перфолент и т. п. Возможность параллельного обслуживания нескольких ВУ с низким быстродействием обусловлена тем, что электронное оборудование канала способно за один цикл работы таких ВУ воспринять и реализовать заявки на действия от многих устройств, включая формирование и передачу слов и байт.

Программное обеспечение работы каналов всех моделей ЕС ЭВМ предусматривает максимальное сокращение обмена информацией между каналами и центральными процессорами. С этой целью организуется иерархическая система программных средств для реализации операций ввода-вывода, причем основная их часть реализуется непосредственно в канале и УВУ.

Первичная командная информация, определяющая номер канала и внешнего устройства, которые должны участвовать в выполнении операции ввода-вывода, вид операции, область оперативной памяти для записи или выборки слов, а также количество байт информационных слов, подлежащих передаче из ОП в данное ВУ или из ВУ в ОП, представляется в виде иерархии трех основных командных слов:

КВВ → АСК → УСК,

где КВВ — это команда ввода-вывода, включаемая в основную программу, которая выполняется процессором;

АСК — адресное слово канала;

УСК — управляющее слово канала.

Отметим, что для лучшего разделения команд процессора и команд канала первые обычно называют инструкциями, а вторые —

просто командами. Всего используется четыре инструкции ввода-вывода;

- НАЧАТЬ ВВОД-ВЫВОД;
- ПРОВЕРИТЬ ВВОД-ВЫВОД;
- ПРОВЕРИТЬ КАНАЛ;
- ОСТАНОВИТЬ ВВОД-ВЫВОД.

Все инструкции ввода-вывода имеют формат SI. При их реализации непосредственный операнд не используется, а адрес канала и адрес внешнего устройства образуются путем сложения базы со смещением (см. п. 8.3). Непосредственно в канал передаются: двухразрядный код инструкции или по отдельным шинам сигналы, отвечающие этим инструкциям (для каналов моделей ЕС-1050 и ЕС-1060), восьмиразрядный адрес внешнего устройства и адрес канала или по отдельной радиальной линии сигнал выборки данного канала (для каналов моделей ЕС-1050 и ЕС-1060). Считается, что любая шина связывает ЦП со всеми каналами, а радиальная линия — только с одним каналом.

Для всех моделей ЭВМ Единой системы принято, что код инструкции ПРОВЕРИТЬ КАНАЛ или соответствующий сигнал в каналы, как правило, не передается, так как эта инструкция обычно выполняется в центральном процессоре. В то же время любая из четырех инструкций ввода-вывода является привилегированной, т. е. выполняется тогда, когда процессор находится в состоянии СУПЕРВИЗОР. Инструкции ПРОВЕРИТЬ ВВОД-ВЫВОД и ОСТАНОВИТЬ ВВОД-ВЫВОД выполняют функции опроса состояния подканала или прекращения операции в подканале. Операция ввода-вывода начинается в канале и в заданном внешнем устройстве только по инструкции НАЧАТЬ ВВОД-ВЫВОД.

Адресное слово канала располагается в фиксированной ячейке основной оперативной памяти (например, в ячейке «72»). Оно определяет адрес следующего слова в иерархии командных слов, представляя собой как бы вспомогательное командное слово, жестко связанное с инструкцией НАЧАТЬ ВВОД-ВЫВОД; ввод АСК в фиксированную ячейку памяти осуществляется управляющей программой перед началом выполнения операции ввода-вывода.

Формат АСК приведен на рис. 13.1. Первый полубайт, т. е. разряды [0—3], представляет собой ключ защиты (КЗ); если средства защиты памяти не используются, то этот полубайт должен содержать нули. Второй полубайт, т. е. разряды [4—7], всегда содержит нули. Остальные разряды представляют адрес УСК; практически это адрес первого байта УСК в основной оперативной памяти, имеющий в трех последних разрядах нули. Выборка АСК из ОП осуществляется каналом по получении кода инструкции НАЧАТЬ ВВОД-ВЫВОД.

Управляющее слово канала представляет собой своеобразную команду, выполняемую каналом и заданным ВУ. Для выполнения

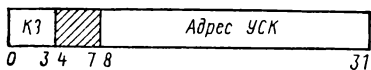


Рис. 13.1. Адресное слово канала

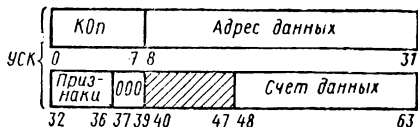


Рис. 13.2. Управляющее слово канала

одной операции ввода-вывода может быть задана последовательность УСК, которые располагаются в соседних или несмежных ячейках основной оперативной памяти. Любое УСК имеет формат двойного слова, как это показано на рис. 13.2.

Первый байт УСК, т. е. разряды [0—7] представляет код операции (КОп), выполняемой каналом и ВУ. Следующие три байта, т. е. разряды [8—31], определяют начальный адрес считываемых из ОП или записываемых в ОП данных; два младших разряда адреса данных указывают номер байта первого обрабатываемого, т. е. считываемого или записываемого, слова. В разрядах [32—36] размещаются признаки, или флажки, разряды [40—47] не используются, а в разрядах [48—63] указывается количество байт данных, подлежащих передаче в ОП или из ОП. Практически группы разрядов [8—31] и [48—63] определяют область основной оперативной памяти, из которой считываются или в которую записываются данные при выполнении данного УСК, так как запись или считывание производятся всегда в смежные или из смежных ячеек памяти.

Признаковая часть УСК используется следующим образом. Наличие единицы в 32-м разряде указывает на цепочку данных (ЦД); в этом случае после завершения передачи данных по исполняемому УСК канал автоматически выбирает следующее УСК из соседней ячейки ОП и продолжает операцию, заданную предыдущим УСК, но при использовании новой области памяти. Если признак цепочки равен нулю, то это означает, что реализуемое УСК является последним в операции ввода-вывода.

В 33-м разряде размещается признак цепочки команд (ЦК). Если признак ЦД равен нулю, то единичное значение ЦК задает цепочку команд. Это означает, что после реализации данного УСК осуществляется переход к УСК, содержащему новый код операции. Цепочка команд отсутствует, если признак ЦК имеет нулевое значение или если признак ЦД равен единице. Отметим, что последовательность выполняемых УСК называют программой канала, или канальной программой.

В 34-м разряде размещается признак блокировки сигнала неправильной длины (БНД), который формируется в канале при несоответствии количества фактически передаваемых между ОП и ВУ байт количеству байт, заданному выполняемым УСК. Сигнал блокируется, если значение рассматриваемого признака равно единице.

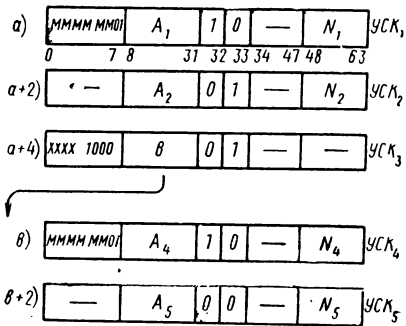


Рис. 13.3. Канальная программа

единицы, а разряды [37—39] в данном УСК содержат нули, то осуществляется прерывание канальной программы.

Управляющая информация, содержащаяся в байте кода операции УСК, т. е. УСК [0—7], задает ту операцию, которая выполняется каналом и заданным внешним устройством. Расшифровке значений УСК [0—7] отвечают данные, приведенные в табл. 10.1 и 10.2. Характер операций, выполняемых каналом и ВУ, рассмотрен в гл. 10 и 12. Отметим только, что выполнение операции перехода производится непосредственно в канале без обращения к ВУ. При выполнении этой операции осуществляется переход к УСК, расположенному в любом месте памяти, произвольном по отношению к адресу предыдущего УСК. Адрес нового УСК указывается в разрядах [8—31] текущего УСК. Рассматриваемая операция никаких действий по вводу-выводу не начинает и служит только для обеспечения связи между УСК, находящимися в несмежных ячейках оперативной памяти.

Пример канальной программы, обеспечивающей передачу информации из различных областей оперативной памяти в ВУ; заданную инструкцией НАЧАТЬ ВВОД-ВЫВОД, приведен на рис. 13.3. Начальным УСК в программе является УСК₁, выбираемое по адресу «а», заданному в АСК; это же УСК начинает цепочку управляющих слов. УСК₁ и УСК₂ составляют цепочку данных, так как в 32-м разряде УСК₁ содержится единица. При выполнении операции записи (код ММММ ММ01) из области памяти, начинающейся ячейкой A₁, в заданное ВУ передается группа из N₁ байт; после этого осуществляется передача N₂ байт из ячеек A₂, A₂ + 1 и т. д. Переход к следующей группе УСК осуществляется по УСК₃, содержащему код операции перехода. В этом же УСК₃ содержится адрес «б», по которому выбирается УСК₄. Последнее УСК, т. е. УСК₅, имеет нулевые признаки цепочек, поэтому по выполнении этого УСК заканчивается операция записи, заданная инструкцией НАЧАТЬ ВВОД-ВЫВОД.

В 35-м разряде размещается признак блокировки записи в оперативную память (БЗП). Если этот признак имеет значение единицы, то запись данных в ОП не производится. Если же он равен нулю, то обмен информацией между ВУ и ОП производится обычным порядком.

В 36-м разряде размещается признак программно-управляемого прерывания (ПУП). Если этот признак имеет значение

Из изложенного следует, что операции ввода-вывода заканчиваются автоматически, если просмотрена вся цепочка управляющих слов; при этом просмотр заканчивается тогда, когда при реализации последнего

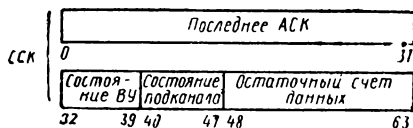


Рис. 13.4. Слово состояния канала

УСК счет данных становится нулевым, т. е. УСК [48—63] = 0*. Для принудительного прекращения операции ввода-вывода используется инструкция ОСТАНОВИТЬ ВВОД-ВЫВОД. По ней производится отключение заданного ВУ от канала, а в канале прекращается выполнение прерываемой операции.

Инструкции ПРОВЕРИТЬ ВВОД-ВЫВОД и ПРОВЕРИТЬ КАНАЛ служат для выявления действительного состояния канала, подканала и ВУ. При выполнении этих инструкций в канале формируется слово состояния канала (ССК) или двухразрядный код состояния. ССК записывается в оперативную память при выполнении инструкций НАЧАТЬ ВВОД-ВЫВОД и ОСТАНОВИТЬ ВВОД-ВЫВОД. Код состояния передается в центральный процессор при выполнении всех инструкций ввода-вывода.

Формат ССК приведен на рис. 13.4. Первая часть слова состояния канала содержит фактически АСК, указывающее адрес последнего из выполненных каналом УСК. Далее располагаются байты состояния адресуемого внешнего устройства (ВУ) и соответствующего подканала (канала), а также остаточный счет данных. Отдельные разряды в байтах состояния отвечают соответствующим указателям (см. п. 10.4). Остаточный счет данных показывает, сколько еще байт данных не передано при реализации последнего УСК.

Код состояния, выдаваемый из канала в процессор при выполнении инструкции ПРОВЕРИТЬ ВВОД-ВЫВОД, идентифицирует следующее:

- 00 — адресуемое устройство доступно и готово к работе;
- 01 — слово состояния канала (ССК) сформировано;
- 10 — канал или подканал заняты;
- 11 — адресуемое устройство не работает (выключено).

При выполнении инструкции ПРОВЕРИТЬ КАНАЛ опрос подканала не производится, а код состояния идентифицирует следующее:

- 00 — канал доступен;
- 01 — канал хранит условие прерывания;
- 10 — подканалы работают в монопольном режиме (для МК);
- 11 — канал выключен (код состояния формируется процессором).

* Здесь и далее обозначение [] = 0 указывает, что в соответствующих разрядах содержатся нули.

Код состояния, выдаваемый из канала в процессор, обычно используется командами условного перехода, следующими в программе процессора после команд (инструкций) ПРОВЕРИТЬ ВВОД-ВЫВОД и ПРОВЕРИТЬ КАНАЛ.

13.2. МУЛЬТИПЛЕКСНЫЙ КАНАЛ ОДНОРОДНОЙ СТРУКТУРЫ

Под мультиплексным каналом однородной структуры понимается канал, предназначенный для параллельного обслуживания медленнодействующих внешних устройств в режиме разделения времени. Внешние запоминающие устройства к такому каналу не подключаются, поэтому он не имеет селекторных подканалов. Все мультиплексные подканалы считаются равноценными и равноправными; для каждого из них в памяти мультиплексного канала отведено по четыре полноразрядных ячейки. Мультиплексный канал однородной структуры включается, например, в состав машины ЕС-1030; в этом случае он имеет 256 подканалов, т. е. к нему возможно подключение до 256 медленнодействующих внешних устройств; пропускная способность канала до 40 кбайт/с при мультиплексном режиме работы.

Необходимость использования в составе мультиплексного канала собственной оперативной памяти диктуется характером обслуживания внешних устройств. В предельном случае за один сеанс связи с данным ВУ канал передает или принимает только один байт информации; после этого он переключается на работу с другим внешним устройством. Так как обмен информацией с ОП производится пословно, то собираемое из байт информационное слово или разделяемое на байты слово должно храниться в канале до очередного сеанса связи с соответствующим ВУ. Кроме того, должна сохраняться вся управляющая информация, относящаяся к этому ВУ и обязательно включающая АСК и УСК. Управляющая информация и данные, т. е. информационное слово, составляют управляющее слово устройства (УСУ), представляющее собой по формату четыре 32-разрядных слова без учета контрольных разрядов.

Поскольку рассматриваемый мультиплексный канал может обслуживать до 256 внешних устройств (или меньшее количество ВУ с номерами от 0 до 255), то он имеет 256 подканалов. Соответственно емкость памяти мультиплексного канала (ПМК) равна 1024 словам; каждое слово с учетом контрольных разрядов является 36-разрядным. Время обращения к ПМК составляет 2 мкс и в значительной степени определяет пропускную способность канала.

Общая схема рассматриваемого мультиплексного канала приведена на рис. 13.5 с выделением основных регистров, схем формирования сигналов, блока управления каналом, схемы обращения к ОП, а также памяти канала. Для этой схемы приняты сле-

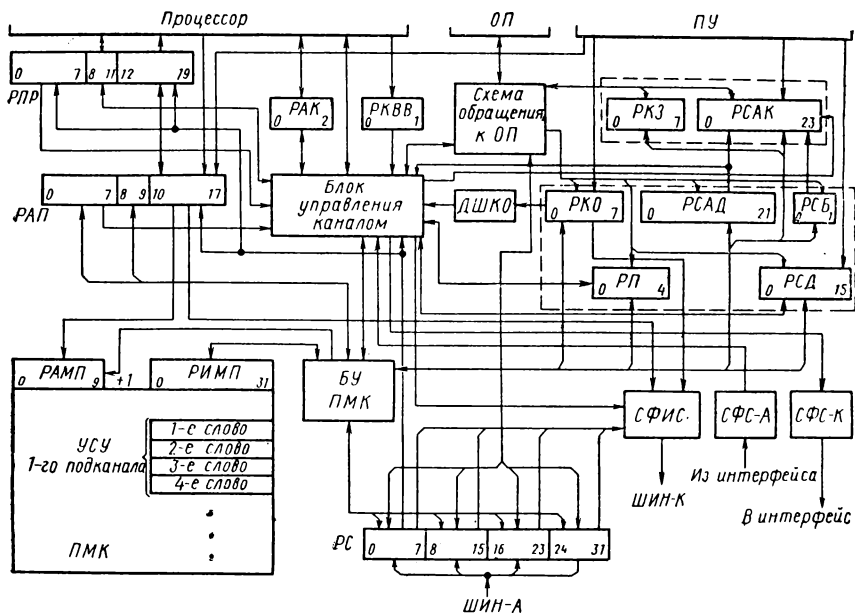


Рис. 13.5. Общая схема мультиплексного канала

дующие обозначения: РКЗ — регистр ключа защиты; RSAK — регистр адреса команд (регистр-счетчик адресов команд), RKO — регистр кода операций, RSAK — регистр адреса данных (регистр-счетчик адресов данных), RCB — регистр счета байт, RP — регистр признаков (указателей, флажков), RCD — регистр счета данных, ДШКО — дешифратор кода операций, РКВВ — регистр команд ввода-вывода, РАК — регистр адреса канала, RPP — регистр прерываний, RAP — регистр активного подканала, РС — регистр связи с интерфейсом, РАМП — регистр адреса мультиплексной памяти, РИМП — регистр информации (регистр числа) мультиплексной памяти, СФИС — схема формирования информационных сигналов, СФС-А — схема формирования интерфейсных сигналов абонента, СФС-К — схема формирования интерфейсных сигналов канала.

Две группы регистров, выделенные на рис. 13.5 пунктирными линиями, служат для хранения и модификации частей АСК и УСК активного подканала, т. е. того, который отвечает взаимодействующему с каналом на данном временном интервале внешнему устройству. Ввод адресного слова канала, выбираемого из ячейки «72» оперативной памяти, осуществляется в РКЗ и RSAK. По содержанию RSAK производится выборка УСК, отвечающего исполняемой инструкции НАЧАТЬ ВВОД-ВЫВОД, в RKO, RSAK, RCB, RCB, RP и RCD. Так как возможно выполнение цепочки команд, то после выборки начального УСК содержимое RSAK увеличивается

на 8 единиц, поскольку УСК имеет формат двойного слова. В случае выполнения операции перехода в РСАК вводится содержимое РСАД и РСБ, что обеспечивает задание новой последовательности команд.

При выполнении заданной операции записи или считывания, а также при выполнении операции перехода содержимое РКЗ не изменяется. Ключ защиты из РКЗ выдается в оперативную память машины при каждом обращении к ней со стороны канала как для выборки УСК, так и для записи или считывания данных. В конце сеанса связи с ВУ содержимое РКЗ и РСАК помещается в ПМК, а в начале очередного сеанса связи с этим же ВУ вновь возвращается в регистры. Отметим, что содержимое РСАК может быть изменено при работе оператора на пульте управления машины; это необходимо, например, при загрузке начальной программы.

Управляющее слово канала для данной операции ввода-вывода первоначально размещается одновременно в РКО, РСАД, РСБ, РП и РСД. Содержимое этих регистров в процессе работы канала отвечает УСК (начальному или модифицированному) активного подканала. В РКО размещается код операции, который при начальной выборке через схему формирования информационных сигналов и ШИН-К передается в адресуемое внешнее устройство. С РКО связан дешифратор на четыре выхода, которые отвечают операциям записи, считывания, считывания в обратном направлении и перехода; сигналы с выхода дешифратора кода операций поступают в блок управления, координирующий работу всех узлов и блоков канала.

Начальное содержимое РСАД и РСБ при выполнении операции отвечает адресу данных исполняемого УСК. Так как обмен данными с ОП производится четырехбайтными словами, то после передачи или приема одного слова содержимое РСАД увеличивается на единицу (запись, прямое считывание) или уменьшается на единицу (обратное считывание); два младших разряда адреса всегда равны нулю. Содержимое РСАД не изменяется только тогда, когда признак блокировки записи, содержащийся в разряде [35] УСК, имеет значение единицы.

Содержимое РСБ отвечает номеру обрабатываемого байта в данном информационном слове. После обработки каждого байта, т. е. после приема его из интерфейса или передачи его в ВУ, содержимое РСБ увеличивается на единицу (запись, прямое считывание) или уменьшается на единицу (обратное считывание). Если при начальной установке РСБ его содержимое не равно 00 (11 при обратном считывании), то это означает, что передача данных начинается не с границы полного слова. В этом случае при операции считывания, например, по начальному адресу из ОП выбирается слово данных, размещаемое в РС, и в нем заменяются байты в соответствии с содержимым РСБ. Затем это слово отсылается в ОП по прежнему адресу. Далее операция выполняется в обычном порядке.

Аналогичные действия производятся также тогда, когда последнее считываемое или записываемое слово является неполным.

Регистр признаков (РП) хранит признаковую часть исполняемого УСК и выдает значения признаков в блок управления каналом.

Регистр счета данных (РСД) служит для размещения и модификации исходного значения числа передаваемых байтов при выполнении заданной операции. Содержимое РСД уменьшается на единицу при каждой передаче через интерфейс одного байта данных. Состояние РСД = 0 свидетельствует об окончании выполнения операции по данному УСК. В этом случае обеспечивается переход к следующему УСК по каналной программе или к завершающим действиям по реализуемой инструкции НАЧАТЬ ВВОД-ВЫВОД. Содержимое РСД, как и содержимое РКО, РСАД, РСБ, РП, при отключении соответствующего ВУ от канала передается для хранения в ПМК.

В регистр команд ввода-вывода (РКВВ) из процессора вводятся двухразрядные коды, отвечающие выполняемым каналом инструкциям:

00 — НАЧАТЬ ВВОД-ВЫВОД;

01 — ПРОВЕРИТЬ ВВОД-ВЫВОД;

10 — ОСТАНОВИТЬ ВВОД-ВЫВОД.

Эти коды как бы настраивают блок управления каналом на выполнение определенных действий. Одновременно с кодом команды (инструкции) канал принимает и исполнительный адрес внешнего устройства.

Регистр адреса канала содержит код номера рассматриваемого мультиплексного канала, равный нулю. Этот нулевой код (адрес) всегда сравнивается с кодом, поступающим от процессора или ПУ при обращении к каналу. При прерываниях по вводу-выводу содержимое РАК записывается в РССП [21—23].

Регистр прерываний (буфер прерываний), т. е. РПР, служит для хранения состояния и адреса того ВУ, которому в данный момент времени дан высший приоритет в запросах на прерывание. Кроме того, он имеет ряд разрядов для хранения признаков. Байт состояния ВУ хранится в РПР [0—7], а его адрес — в РПР [12—19], ввод этой информации производится из интерфейса через РС [0—7]. Содержимое признаковой части указывает на следующее:

РПР [8] = 1, когда канал полностью заполнил буфер прерываний;

РПР [9] = 1 — канал выполняет заданную операцию ввода-вывода;

РПР [10] = 1 — признак разрешения связи канала с процессором для получения команды ввода-вывода или для выполнения прерывания ввода-вывода;

РПР [11] = 1 — канал неисправен вследствие отказа в его аппаратуре.

Построение регистра активного подканала, или РАП, аналогично построению РПР: две однобайтные части и признаковая часть. В РАП [0—7] хранится байт состояния, а в РАП [10—17] — адрес активного подканала, т. е. соответствующего внешнего устройства. Устройство, которое связалось с каналом для передачи информации, помещает свой адрес в РАП [10—17] и монополизирует оборудование канала, заполняя регистры своим УСУ, вызываемым из ПМК по адресу РАП [10—17]. Содержимое РАП [8—9] характеризует состояние подканала.

Память мультиплексного канала, как отмечалось выше, рассчитана на одновременное хранение 256 управляющих слов внешних устройств (соответствующих подканалов). Каждое УСУ занимает 4 соседних ячейки памяти, так как имеет формат счетверенного слова. Первое слово, входящее в состав УСУ, выбирается по адресу, задаваемому содержимым РАМП [0—7] = РАП [10—17] при РАМП [8—9] = 0. Остальные слова, составляющие УСУ, выбираются за счет увеличения в каждом такте на единицу содержимого РАМП [8—9]; очевидно, что таких дополнительных тактов при выборе УСУ — три.

Записываемые в ПМК слова, а также слова, считываемые из ПМК, размещаются предварительно в регистре информации, связанном с блоком управления (БУ ПМК) посредством групп входных и выходных вентилях. По передаче и приему информационных слов БУ ПМК связан, кроме регистра информации мультиплексной памяти, с регистрами хранения и модификации АСК и УСК, а также с РС, обеспечивающим связь с интерфейсом.

Распределение участка ПМК, включающего в свой состав четыре последовательных ячейки, для размещения управляющего слова устройства, иллюстрируется данными, приведенными в табл. 13.1.

Т а б л и ц а 13.1

Размещение УСУ

Части УСУ (ячейки ПМК)	Разряды	Содержимое
1	[0—7]	РКЗ [0—7] — ключ защиты ОП машины
2	[8—31]	РСАК [0—23] — адрес УСК в ОП
	[0—4]	РП [0—4] — признаки в УСК
3	[5]	Не используется
	[6—7]	РАП [8—9] — режим подканала
	[8—15]	РАП [0—7] — состояние подканала (БУ)
	[16—31]	РСД [0—15] — счет байт данных
4	[0—7]	РКО [0—7] — код операции
	[8—29]	РСАД [0—21] — адрес данных в ОП
4	[30—31]	РСБ [0—1] — счет байт данных
	[0—31]	РС [0—31] — данные (слово данных)

Регистр связи с интерфейсом имеет 32 информационных разряда (контрольные разряды для него, как и для остальных регистров канала, на рис. 13.5 не показаны), функционально он разделен на четыре однобайтные части РС [0—7], РС [8—15], РС [16—23], РС [24—31]. Каждая часть имеет группы входных и выходных вентиляей; через группы входных вентиляей осуществляется прием байт данных из ШИН-А и их распределение по частям регистра; через группы выходных вентиляей осуществляется выдача байт данных через СФИС в ШИН-К. Часть регистра РС [0—7] имеет дополнительные группы выходных вентиляей для выдачи адресов ВУ и байт их состояний в РПР, РАП и блок управления каналом. Посредством полноразрядных групп входных и выходных вентиляей регистр РС связан с ПМК через ее блок управления и со схемой обращения к ОП. Эти группы вентиляей служат для приема и выдачи слов данных (информационных слов).

Схемы формирования информационных и интерфейсных сигналов обеспечивают усиление и формирование сигналов, поступающих из линий интерфейса или подаваемых в линии интерфейса. Аналогичная схема, не показанная на рис. 13.5, имеется и в цепи ШИН-К — регистр связи с интерфейсом.

Режимы работы (состояния) канала. Прерывания. Для рассматриваемого мультиплексного канала характерны четыре режима работы.

1. Канал в нерабочем состоянии (РПР [11] = 1); он не может выполнять операции ввода-вывода из-за отказа в аппаратуре.

2. Канал в рабочем состоянии, но никакой операцией не занят и не взаимодействует с процессором (РПР [9—11] = 0). В таком состоянии канал начинает опрос ВУ с целью выборки устройства, требующего обслуживания со стороны канала для обработки данных или выдачи байта состояния, что отвечает требованию на прерывание от ввода-вывода.

3. Канал принимает команду ввода-вывода из процессора. Считается, что он находится в рабочем состоянии, но еще не занят выполнением операции ввода-вывода (РПР [9, 11] = 0).

4. Канал обслуживает ввод-выводное прерывание; состояние его — рабочее (РПР [11] = 0). Мультиплексный канал выполняет прерывание, когда РПР [8] = 1 и процессор предоставил каналу возможность реализации прерывания. При этом он записывает сформированное блоком управления ССК в фиксированную ячейку ОП «64» или «68». Если прерывание вызвано наличием признака программно-управляемого прерывания (РП [4] = 1), то дополнительно в РССП процессора записываются адрес устройства и адрес канала.

Канал формирует требование на обслуживание прерывания со стороны процессора по требованию от ВУ. Это происходит в тех случаях, когда после выдачи своего адреса ВУ формирует сигнал идентификации УПР-А, свидетельствующий о выдаче на ШИН-А

байта состояния ВУ. Байт состояния размещается в РПР [0—7], адрес ВУ — в РПР [12—19], а РПР [8] придается значение единицы, что свидетельствует о наличии запроса на прерывание.

Выполнение команд ввода-вывода. На любую команду ввода-вывода канал отвечает либо записью ССК в фиксированную ячейку «64», либо выдачей в процессор двухразрядного кода состояния. Данные табл. 13.2 показывают, как идентифицируются коды состояния в зависимости от вида выполняемой команды.

Т а б л и ц а 13.2

Расшифровка кодов состояний

Коды состояний	Идентификация кодов состояний при командах		
	НАЧАТЬ ВВОД-ВЫВОД	ПРОВЕРИТЬ ВВОД-ВЫВОД	ОСТАНОВИТЬ ВВОД-ВЫВОД
00	Операция начата и канал продолжает ее выполнение	Канал готов к работе	Канал или подканал не работает
01 10	ССК записано Канал или подканал занят	ССК сформировано Канал или подканал занят	ССК записано Монопольная передача информации закончена
11	Устройство не работает	Устройство не работает	Устройство не работает

По команде (инструкции) НАЧАТЬ ВВОД-ВЫВОД канал вызывает из фиксированной ячейки «72» адресное слово канала, а по АСК — управляющее слово канала; при этом адрес внешнего устройства уже принят из процессора в РАП [10—17]. Далее осуществляется выборка ВУ и передача ему кода операции из РКО [0—7]. Если ВУ отвечает нулевым байтом состояния, то в процессор направляется код состояния 00, осуществляется запоминание УСУ данного устройства в ПМК, и устройство отключается от канала до начала очередного сеанса связи. Если устройство отвергает команду, то канал организует запись части ССК в фиксированную ячейку ОП.

По команде ПРОВЕРИТЬ ВВОД-ВЫВОД осуществляется выборка из ПМК управляющего слова устройства, адрес которого задан выполняемой командой. Если адресуемое устройство заполнило буфер прерываний и находится в состоянии окончания выполнения операции, то производится запись полного ССК, а в канале и подканале (устройстве) сбрасываются состояния ожидания прерывания. В противном случае осуществляется выборка устройства с целью получения от него байта состояния.

Команда ОСТАНОВИТЬ ВВОД-ВЫВОД относится только к адресуемому устройству, которое по этой команде прекращает выполнение операции. Если устройство монополюно занимает оборудование канала, то для реализации рассматриваемой команды его адрес не требуется.

Обработка данных при выполнении операции записи. При реализации команды НАЧАТЬ ВВОД-ВЫВОД до записи УСУ в ПМК производится выборка из ОП первого слова данных по адресу, находящемуся в РСАД. Когда устройство вновь связывается с каналом, производится вызов УСУ из ПМК с размещением слова данных в РС и начинается обработка байта, задаваемого содержимым РСБ. Эта обработка сводится к выделению требуемого байта и передаче его на ШИН-К.

Обработка байт данных производится по сигналам ТРБ-А, поступающим от устройства. При опросе устройство выдает свой адрес на ШИН-А, сопровождая его сигналами АДР-А и РАБ-А. Сигналом УПР-К канал сообщает ВУ о приеме его адреса; по этому адресу из ПМК выбирается соответствующее УСУ. Передача байта данных осуществляется по получении из ВУ сигнала ИНФ-А.

Канал обращается к ОП для выборки очередного слова данных во время обработки последнего байта в текущем слове данных. После передачи этого байта во внешнее устройство РС освобождается и в нем размещается новое слово данных. Далее УСУ, включающее в свой состав это новое слово данных, запоминается в ПМК, где хранится до очередного сеанса связи с внешним устройством. Передача каждого байта через интерфейс уменьшает содержимое РСД на единицу и увеличивает содержимое РСБ на единицу. После каждого обращения к ОП содержимое РСАД увеличивается на единицу.

В конце выполнения операции, когда устройство связывается с каналом для приема последнего байта, выделяемого из последнего слова данных, производится анализ состояния РП [0], содержащего признак цепочки данных. При РП [0] = 1, т. е. при наличии цепочки данных, осуществляется выборка из ОП очередного УСК по адресу, задаваемому содержимым РСАК. Это УСК размещается в РКО, РСАД, РСБ, РП и РСД; по новому содержимому РСАД из ОП выбирается первое слово данных и размещается в РС. Устройство отключается от канала, а в канале производится передача нового УСУ в ПМК для хранения.

Если выбранное по цепочке команд УСК содержит код операции перехода, то производится передача содержимого РСАД и РСБ в РСАК. Тем самым в регистре адреса команд устанавливается значение адреса, определяющего новую последовательность УСК. По новому адресу производится выборка УСК из ОП и размещение его по соответствующим регистрам канала.

Обработка данных при выполнении операции считывания. Начальная выборка устройства при выполнении операции считывания производится так же, как и при операции записи. При этом, если после выборки УСК содержимое РСБ равно нулю, то в состав первого УСУ, запоминаемого в ПМК, слово данных не включается. Если же содержимое РСБ не равно нулю (или 11 при операции обратного считывания), то по содержимому РСАД выбирается

слово данных из ОП и включается в состав первого УСУ; содержимое РСАД при этом не изменяется. Последний случай отвечает ситуации, когда часть байт некоторого слова заменяется байтами, получаемыми при считывании от ВУ.

Пусть РСБ = 10 при операции прямого считывания, а выбранное из ОП слово данных представляется четырьмя байтами $V_0V_1V_2V_3$. При очередном сеансе связи с ВУ это слово помещается в РС. Принимаемый от ВУ байт данных V'_2 заменяет байт V_2 в исходном слове, которое принимает вид $V_0V_1V'_2V_3$. Содержимое РСБ увеличивается на единицу, т. е. $РСБ := РСБ + 1$; содержимое РСД уменьшается на единицу; в ПМК запоминается модифицированное УСУ. Во время очередного сеанса связи производится замена третьего байта в слове данных, принимающем окончательный вид $V_0V_1V'_2V'_3$. После увеличения на единицу содержимое РСБ становится равным нулю; содержимое РСД уменьшается на единицу. Сформированное в РС слово направляется для записи в ОП по исходному содержимому РСАД; после этого содержимое РСАД увеличивается на единицу.

Далее операция считывания производится обычным порядком, т. е. в РС осуществляется группирование принимаемых из ВУ байт в слова, которые записываются в ОП по адресам, представляющим собой содержимое РСАД, увеличиваемое на единицу после каждого обращения к ОП. При выполнении операции обратного считывания содержимое РСАД после каждого обращения к ОП уменьшается на единицу; содержимое РСБ также модифицируется за счет уменьшения его на единицу при обработке каждого байта. Содержимое РСД всегда модифицируется за счет его уменьшения на единицу после передачи через интерфейс одного байта данных.

При обработке последнего байта в последнем слове производится анализ остаточного счета байт. Если последнее слово содержит не четыре байта, то оно формируется подобно первому, когда исходное состояние РСБ не равно нулю. Слово данных вызывается из ОП; соответствующие байты этого слова заменяются байтами, поступающими от ВУ; после этого слово записывается в ОП по прежнему адресу.

После завершения действий по выполнению операции считывания по данному УСК производится анализ признаков цепочек данных и команд. Если эти признаки равны нулю, то операция заканчивается; если же имеет место цепочка данных или цепочка команд, то производятся действия по выборке очередного УСК и его реализации.

13.3. СЕЛЕКТОРНЫЕ КАНАЛЫ

Селекторные каналы любой машины Единой системы служат для подключения и обслуживания внешних запоминающих устройств и работают всегда в монопольном режиме, имея в своем составе по одному подканалу. Их конкретный состав и характери-

стики связаны с соответствующими показателями машин, частью которых они являются. Для старших моделей характерно выделение селекторных каналов как сугубо автономных устройств высокой производительности; в младших моделях селекторные каналы используют частично аппаратуру процессоров, имея относительно невысокую пропускную способность. В любом случае общая структура селекторного канала строится по классическому принципу, описанному, например в [12].

Селекторные каналы машины ЕС-1020 функционально и конструктивно включены в состав процессора; основу каждого из двух каналов составляют буферные регистры, узлы преобразования УСК и узлы управления. При одновременной работе каналов первый из них имеет повышенную пропускную способность, поэтому к нему подключаются накопители на магнитных дисках; ко второму селекторному каналу подключаются накопители на магнитных лентах. Максимальная скорость обмена в каждом канале — до 200 кбайт/с.

Селекторные каналы машины ЕС-1030 по своей общей структуре близки к мультиплексному каналу (без мультиплексной памяти), описанному в п. 13.2. Работают они только в монопольном режиме, обеспечивая скорость обмена (скорость передачи данных) до 600—900 кбайт/с.

Наиболее сложными являются селекторные каналы машины ЕС-1050. Они полностью автономны и обеспечивают высокие скорости передачи данных. При этом к каждому селекторному каналу ЕС-4035 допускается подключение одного адаптера канал—канал ЕС-4060, через который он может быть связан с селекторным каналом этой же или другой машины.

Назначение и общая структура селекторного канала ЕС-4035. Селекторный канал ЕС-4035 предназначен для организации обмена информацией между основной памятью и внешними запоминающими устройствами в монопольном режиме передачи. Этот канал представляет собой автономное устройство, включаемое в состав машины ЕС-1050. Канал связан с ЦП, ОП и интерфейсом в соответствии с общими принципами построения машин Единой системы. Обмен информацией с ОП осуществляется двойными словами (72 разряда, из которых 8 — контрольные), а с внешними устройствами — побайтно (9 разрядов, включая один контрольный). Канал обеспечивает максимальную скорость передачи данных до $1,25 \cdot 10^6$ байт/с.

Обмен информацией между каналом и ЦП осуществляется по адресным шинам и шинам передачи управляющих сигналов и кодов признаков. Для передачи адреса ВУ из ЦП в канал используется 9 шин; передача управляющих сигналов и кодов признаков из ЦП в канал производится по девяти шинам, включая четыре шины для передачи сигналов-признаков инструкций ввода-вывода. Для передачи адреса ВУ из канала в ЦП используется

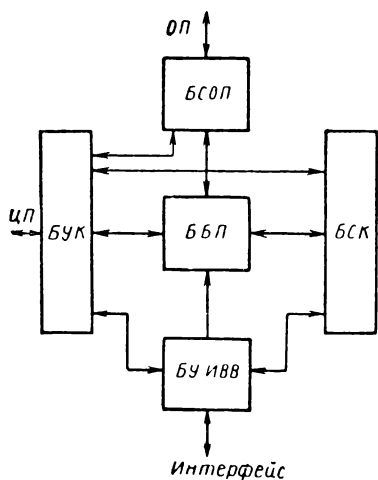


Рис. 13.6. Общая схема селекторного канала

- блок сопряжения с оперативной памятью (БСОП);
- блок буферных регистров, или буферной памяти (ББП);
- блок управления интерфейсом ввода-вывода (БУ ИВВ);
- блок состояния канала и условий прерывания (БСК);
- блок управления каналом (БУК).

Блок БСОП обеспечивает обмен информацией с ОП как при передаче данных, так и при передаче управляющей информации. В ББП осуществляется промежуточное хранение передаваемых данных и преобразование их форматов. Блок БУ ИВВ служит для управления работой интерфейса ввода-вывода при передаче информации.

Блок БСК предназначен для выработки, фиксации и обработки всех возможных условий состояния канала и реализации совместно с БУК и БСОП прерываний. Блок БУК обеспечивает централизованное управление каналом при любых режимах работы; он непосредственно связан с ЦП и со всеми остальными блоками канала. Блоки БСОП, ББП и БУ ИВВ в общем случае образуют магистраль передачи данных, как это показано на схеме канала (рис. 13.6).

Блок сопряжения с оперативной памятью представляет собой совокупность регистров с входными и выходными логическими схемами, а также схем управления и контроля. Он связан с ОП информационными и адресными шинами, а также шинами передачи сигналов признаков и запросов, которые в дальнейшем именуется управляющими сигналами (УС). При каждом обращении к ОП в БСОП формируется запросное слово, включающее в свой состав адрес, ключ защиты памяти, признак режима работы, маркеры и данные, если осуществляется запись. Запросное слово со-

9 шин; передача запросов и кодов признаков из канала в ЦП осуществляется по шести шинам. При обмене информацией между каналом и ОП дополнительно используются информационные шины. Данные из ОП в канал поступают по 72 информационным шинам; для передачи признаков используются восемь шин. Данные из канала в ОП передаются по 72 информационным шинам, адреса — по 24 адресным шинам; для передачи маркеров байтов используется 9 шин, кода защиты памяти — 5 шин, сигналов признаков и запросов — 5 шин.

Селекторный канал включает в свой состав пять основных блоков (рис. 13.6):

проводится специальным сигналом, по которому блок управления ОП удовлетворяет запрос канала при первом же свободном цикле памяти. После удовлетворения запроса блок управления ОП посылает соответствующий сигнал в БСОП.

Основное место в составе БСОП занимает регистр данных (РД) (рис. 13.7). Он является 72-разрядным триггерным регистром (64 информационных разряда и 8 контрольных). Прием информационных двойных слов производится из ОП, а также из ББП канала; кроме того, при формировании слова состояния канала в РД из регистра ключа защиты (РКЗ) вводится код ключа защиты памяти, из БУК — адрес следующего УСК и остаточный счет данных, а из БСК — байты состояния канала и устройства. Выдача двойных слов из РД осуществляется в ОП, а также в ББП и БУК канала; кроме того, код ключа защиты выдается в регистр РКЗ, а адрес УСК — в БУК.

Регистр адреса (РА) служит для формирования стандартных адресов для выборки АСК или записи ССК, а также для передачи в ОП адресов данных, поступающих из БУК. Формирование и передача адресов осуществляются под действием сигналов, поступающих из узла управления; адреса в регистре РА не хранятся; поэтому он выполнен только на логических элементах. Регистр имеет 21 основной разряд и 3 контрольных разряда.

Регистр маркеров (РМ) получает информацию из ББП, а выдает ее в оперативную память в составе запросного слова. Каждый из восьми основных разрядов регистра отвечает одному байту информации, содержащейся в регистре РД. Если данный основной разряд РМ содержит единицу, то соответствующий ему байт записывается в ОП; в противном случае запись не осуществляется. Регистр маркеров выполнен на триггерах и кроме основных имеет один контрольный разряд.

Четырехразрядный триггерный регистр ключа защиты служит для хранения кода ключа защиты в течение всего времени работы канала по выполнению одной инструкции НАЧАТЬ ВВОД-ВЫВОД. При обращении к памяти код ключа защиты передается в ОП в составе запросного слова. При выборке адресного слова канала и записи ССК, т. е. при обращении к стандартным ячейкам памяти, в РКЗ формируется нулевой код защиты. Прием кода в РКЗ производится из разрядов [0—3] РД; в эти же разряды РД осуществляется передача кода ключа защиты при формировании ССК.

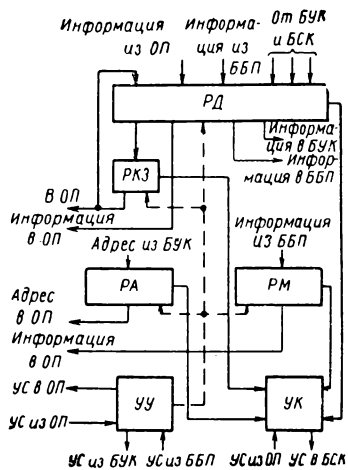


Рис. 13.7. Блок сопряжения с оперативной памятью

Узел управления (УУ) по внешним управляющим сигналам, поступающим из блока управления ОП и БУК, формирует «внутренние» управляющие сигналы, координирующие работу всех остальных узлов блока (пунктирные линии на рис. 13.7) в режиме обмена информацией с оперативной памятью. Отдельные управляющие сигналы посылаются УУ в блок управления ОП.

В узле контроля (УК) осуществляется контроль на четность всей обрабатываемой в рассматриваемом блоке информации. Кроме того, УК воспринимает управляющие сигналы из ОП, сигнализирующие о наличии сбоев при обращении к памяти, а также формирует сигналы о наличии сбоев, направляемые в БСК. Сигналы в БСК выдаются по результатам контроля на четность обрабатываемых данных, управляющей информации, кода защиты памяти и контроля канальной программы (по формату АСК).

Блок буферной памяти представляет собой совокупность регистровых и логических узлов, а также схем управления. Он занимает промежуточное положение между блоком интерфейса и блоком сопряжения с оперативной памятью, осуществляя, главным образом, объединение последовательности байт в двойные слова при передаче информации из ВУ в ОП и разбиение слов на байты при передаче информации из ОП в ВУ. В обоих возможных режимах работы ББП определяет конец обмена информацией с ВУ, что фиксируется формированием сигналов, посылаемых в блоки БУК и БСК. Кроме того, через входной и выходной регистры блока передается служебная информация. Структура блока приведена на рис. 13.8, где показаны его основные узлы, а также основные внутренние и внешние связи.

Входной регистр РВ служит для приема информации от внешнего устройства (через УВУ и блок интерфейса канала) и выдачи ее в буферный регистр РБ1, а также в БУК и БСК. Его основу составляют триггеры с входными и выходными логическими схемами. Регистр РВ имеет восемь информационных разрядов и один контрольный разряд. Выходной регистр ВР аналогичен по построению регистру РВ. Он принимает информацию из буферного регистра РБ2 и передает ее в блок интерфейса для дальнейшей передачи в выбранное внешнее устройство.

Основу рассматриваемого блока составляют два 72-разрядных буферных регистра РБ1 и РБ2. При передаче информации из ОП в ВУ в РБ1 поступают двойные слова, включая контрольные разряды, из БСОП канала. При передаче информации из ВУ в ОП в этом регистре формируется двойное слово по байтам, поступающим из входного регистра; распределение байтов по группам разрядов регистра производится с помощью восьми групп входных вентилей, управляемых сигналами, поступающими из узла управления. Второй буферный регистр обеспечивает хранение и преобразование двойных слов, поступающих из РБ1. Выдача двойных слов из РБ2 осуществляется через БСОП в оперативную память. При

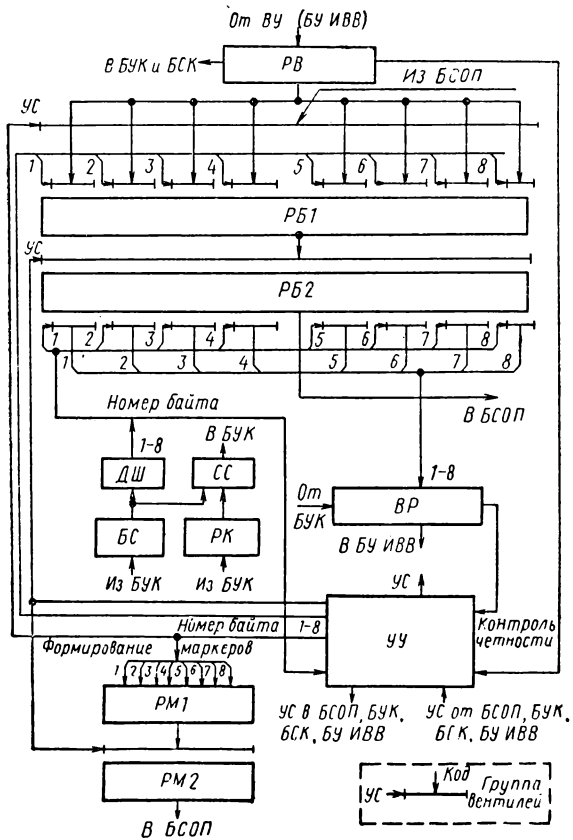


Рис. 13.8. Блок буферной памяти

обмене типа «память — внешнее устройство» двойное слово, размещенное в РБ2, с помощью восьми групп выходных вентилях, управляемых сигналами, образующимися на выходах дешифратора ДШ, разделяется на байты, последовательно передаваемые в выходной регистр.

Регистр формирования маркеров РМ1 имеет восемь информационных и один контрольный разряд; он служит для образования кода маркеров в режиме обмена «внешнее устройство. — память». Формирование маркеров производится управляющими сигналами, распределяющими байты данных по группам разрядов регистра РБ1. При передаче данных из РБ1 в РБ2 код маркеров передается из регистра РМ1 в регистр РМ2 и далее в БСОП.

Буферный счетчик (БС) служит для размещения трех младших разрядов адреса данных, поступающих из БУК и представляющих собой номер байта в двойном слове. Содержимое счетчика изменяется по сигналам от узла управления. Расшифровка содержи-

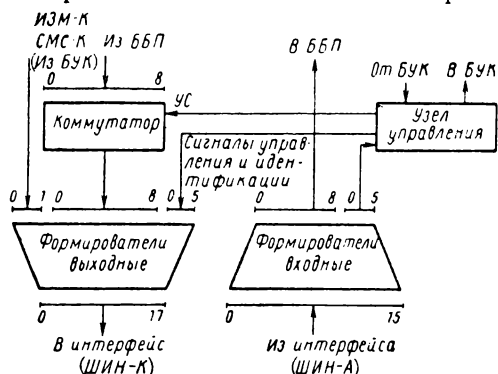
мого счетчика производится посредством дешифратора ДШ, который на своих выходах формирует сигналы, управляющие работой выходных вентилях регистра РБ2; эти же сигналы используются в узле управления при формировании УС, управляющих работой входных вентилях регистра РБ.

В регистре конца (РК) размещается содержимое трех младших разрядов счетчика байтов блока управления каналом, определяющее количество байтов, подлежащих передаче в последнем информационном слове. Схема сравнения (СС) сравнивает содержимое РК и буферного счетчика (БС). В случае совпадения кодов СС формирует сигнал об окончании обмена информацией с внешним устройством.

Узел управления рассматриваемого блока по внешним управляющим сигналам формирует все управляющие сигналы, необходимые для функционирования всех остальных узлов блока в различных режимах работы, а также для связи с другими блоками канала. Схемы узла управления обеспечивают также контроль на четность информации, обрабатываемой в блоке буферной памяти.

Блок управления интерфейсом состоит из коммутатора информации, узла управления и формирователей, как это показано на его структурной схеме (рис. 13.9). Узел управления по сигналам, полученным из блока УК, формирует сигналы, управляющие выдачей информации из коммутатора в выходные формирователи, а также сигналы, передаваемые в УВУ по шинам интерфейса. В этом узле формируются также сигналы, посылаемые в блок УК, включая сигналы о сбоях в интерфейсе; основой при формировании этих сигналов служит информация, поступающая из интерфейса. Коммутатор представляет собой совокупность девятиразрядных групп вентилях, управляемых сигналами, поступающими из узла управления.

Блок состояния канала представляет собой совокупность регистров и логических схем. Он принимает информацию о состоянии



канала и внешних устройств, анализирует ее и сигнализирует блоку управления каналом о возможности продолжения работы или необходимости ее прекращения и прерывания программы. Блок БСК, кроме того, формирует и хранит информацию о состоянии канала и ВУ, необходимую для составления слова состояния канала.

Рис. 13.9. Блок управления интерфейсом

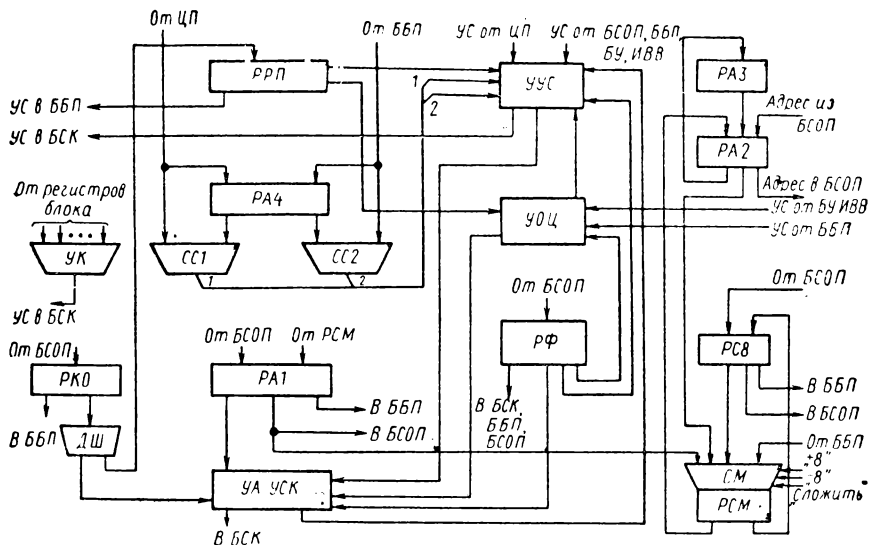


Рис. 13.10. Блок управления каналом

Блок управления каналом представляет собой совокупность запоминающих, операционных узлов и узлов формирования управляющих сигналов. Он обеспечивает организацию связи с центральным процессором, принимая от него коды и преобразуя их в наборы управляющих сигналов, и организует запросы на прерывания; осуществляет общее управление блоками канала в различных режимах работы; осуществляет контроль передачи управляющей информации и правильность ее задания. Упрощенная схема блока приведена на рис. 13.10.

Регистр RA1 служит для хранения адреса данных и имеет 24 информационных и 3 контрольных разряда. Адрес данных, указывающий номер ячейки ОП, откуда выбирается или куда заносится информация, принимается из регистра информации блока БСОП после выборки очередного УСК. Предусмотрен также ввод чисел в RA1 из регистра сумматора, где адрес данных изменяется на ± 8 единиц после каждого обращения в память по записи или считыванию данных, которые передаются двойными словами. При обращении к ОП содержимое RA1 выдается в БСОП, а также на сумматор для изменения на ± 8 единиц. Младшие разряды адреса данных, определяющие номер байта двойного слова, с которого начинается обмен «ОП \rightarrow ВУ» или «ВУ \rightarrow ОП», из регистра RA1 передаются на буферный счетчик блока ББП и в сумматор СМ для модификации счета в регистре РСВ.

Первый регистр адреса УСК, или регистр RA2, служит для хранения адреса УСК, имея 21 информационный и 3 контрольных разряда. Адрес УСК вводится в регистр RA2 из блока БСОП. Воз-

можно также поступление кода адреса из регистров РА3 и РСМ. Адрес УСК из РА2 передается в РА3 при выполнении цепочки данных в операции записи, в блок БСОП — при обращении в оперативную память по выборке нового УСК, а также при формировании полного ССК, в сумматор СМ — для получения адреса нового УСК.

Второй регистр адреса УСК, или регистр РА3, предназначен для промежуточного хранения адреса УСК и имеет ту же разрядность, что и регистр РА2. Прием адреса УСК в регистр РА3 из регистра РА2 производится по цепочке данных, когда канал обращается в ОП за новым УСК. Если при передаче информации, связанной с этим действием, произойдет сбой, то для определения, на каком УСК произошел сбой, код адреса УСК передается из РА3 в РА2. Таким образом, возникновение сбойных ситуаций не приводит к искажению или потере адреса последнего по выполнению управляющего слова канала.

Регистр адреса устройства, или регистр РА4, предназначен для хранения адреса устройства, которое либо участвует в операции, либо связано с условиями, приведшими канал в состояние ожидания прерывания. Он имеет восемь информационных разрядов и один контрольный разряд. Его выходы связаны с входами двух схем сравнения СС1 и СС2. Схема сравнения СС1 выдает сигнал в узел управления блоком при совпадении адреса, находящегося в регистре РА4, с адресом, поступающим из центрального процессора; работа СС1 происходит в режиме ожидания прерывания и при выполнении инструкции ПРОВЕРИТЬ ВВОД-ВЫВОД. Схема СС2 формирует на своем выходе сигнал при несовпадении адреса, переданного в интерфейс и находящегося в регистре РА4, с адресом, поступающим из интерфейса через блок ББП; этот сигнал в конечном итоге подается в блок БСК, как сигнал «ПРОВЕРИТЬ УПРАВЛЕНИЕ ИНТЕРФЕЙСОМ».

Регистр кода операции (РКО) имеет $(8+1)$ разрядов и служит для размещения кода операции, задаваемой выполняемым УСК; код принимается из блока БСОП и через блоки ББП и БУ ИВВ передается в выбираемое УВУ. Четыре младших разряда кода подаются также на дешифратор ДШ, который связан с регистром режима передачи РРП и узлом анализа УСК. Единичное состояние разрядов РРП определяет режимы:

- в 0-м разряде — считывание;
- в 1-м разряде — запись;
- во 2-м разряде — обратное считывание (считывание при движении носителя в обратном направлении).

Регистр флажков (РФ) имеет $(8+1)$ разрядов и служит для хранения пятиразрядного кода флажков (признаков), содержащегося в выполняемом УСК, а также значений 37—39-го разрядов этого управляющего слова; при правильном формате УСК его 37—39-й разряды должны содержать нули, и это обстоятельство, в конечном итоге, проверяется в узле анализа УСК.

Регистр РС8 имеет $(16 + 2)$ разрядов и служит для хранения числа, определяющего количество переданных байтов информации. Он связан с сумматором СМ, с помощью которого осуществляется уменьшение (увеличение) на 8 единиц содержимого РС8 при передачах двойных слов, а также его модификации за счет прибавления трех младших разрядов из регистра РА1. Содержимое трех младших разрядов регистра РС8 может передаваться в блок ББП; при записи полного слова состояния канала все содержимое регистра РС8 передается в блок БСОП. Сумматор СМ представляет собой 24-разрядную суммирующую схему, 21 младший разряд которой выполнен в виде счетчика. Результаты действий, выполняемых в сумматоре, хранятся в регистре РСМ.

Узел управляющих сигналов (УУС) служит для формирования различных управляющих сигналов в процессе работы канала; исходной информацией для него являются признаки инструкций, поступающие из центрального процессора, а также признаки режимов работы. Узел УУС работает совместно с узлом организации цепочек УОЦ, обеспечивая организацию цепочек команд и цепочек данных.

Узел анализа УСК (УА УСК) служит для проверки правильности задания управляющей информации, содержащейся в очередном УСК и формирования соответствующих сигналов контроля и управления. При обнаружении неправильного формата УСК узел вырабатывает сигнал ПРОВЕРИТЬ ПРОГРАММУ, направляемый в БСК.

Контроль передач кодов в блоке управления каналом производится по общим принципам контроля на четность. С этой целью используется отдельный узел контроля (УК), показанный в левой части схемы рис. 13.10.

Формирование последовательностей управляющих сигналов производится в блоке управления всегда в соответствии с общим алгоритмом работы канала, определяющим действия при выполнении различных команд процессора с учетом возможных состояний канала.

Общая последовательность действий при выполнении каналом команд процессора. В процессе работы канал может находиться в одном из четырех возможных состояний. Состояние *С* означает, что канал свободен и готов выполнять любую команду процессора, или инструкцию. Состояние *СУ* есть состояние ожидания прерывания, когда условие прерывания сформировано в каком-либо внешнем устройстве. В этом состоянии канал может выполнять инструкции, если они не относятся к устройству, сформировавшему условия прерывания. Состояние *СК* есть состояние ожидания прерывания, когда условие прерывания сформировано в самом канале, например, вследствие сбоя в одном из его блоков. Состояние *Р* означает, что канал работает, т. е. занят выпол-

нением заданной инструкции. Состояния канала отражаются состояниями соответствующих триггеров его блока управления.

При выполнении инструкции ПРОВЕРИТЬ КАНАЛ состояние канала не изменяется; производится только выдача кода состояния, после чего канал отключается от ЦП. При выполнении инструкции ПРОВЕРИТЬ ВВОД-ВЫВОД состояние канала не изменяется, если он работает; выдается только код 10. Если начальное состояние канала при выполнении этой инструкции есть СК и адресуемое ВУ свободно, то осуществляется запись ССК в основную память, а канал переходит в состояние С. При состояниях С и СУ, если нет сбоев, осуществляется запись ССК в ОП, а канал остается в доступном состоянии; отключение от процессора производится. При выполнении инструкции ОСТАНОВИТЬ ВВОД-ВЫВОД канал из состояния Р переводится в состояние СК при выдаче в ЦП кода 10, из состояний С и СУ — в состояние С при выдаче в ЦП кода 00; во всех случаях канал отключается от процессора. Если задана инструкция НАЧАТЬ ВВОД-ВЫВОД и канал находится в состоянии Р или состоянии СК, то все действия канала сводятся к передаче кода состояния 10 и отключению от процессора. Последовательность действий по входу в операцию начинается только при состояниях канала С и СУ.

Все действия по выполнению инструкций производятся в следующей последовательности: вход в операцию, обмен информацией, завершение операции. В общем алгоритме работы канала выделяется также последовательность действий по учету прерываний в соответствии с сигналами, получаемыми от центрального процессора.

Вход в операцию. Работа канала начинается по сигналу ОБРАЩЕНИЕ К КАНАЛУ, поступающему из процессора. Канал осуществляет прием инструкции, анализирует состояние блоков и, в зависимости от типа инструкции либо выполняет ее сразу (например, ОСТАНОВИТЬ ВВОД-ВЫВОД), либо формирует код состояния и отключается от процессора, либо принимает адрес устройства и переходит к установлению с ним связи.

Процесс установления связи с ВУ заключается в возбуждении шины ВБР-К интерфейса и передаче адреса устройства в ШИН-К. В знак подтверждения установления с ним связи устройство в ШИН-А выдает свой адрес. Если этого не происходит, то канал формирует код состояния, ССК и отключением от процессора прекращает выполнение операции. В случае, когда ответный адрес ВУ совпадает с заданным и принимается без сбоев, канал продолжает выполнение операции.

Для продолжения выполнения инструкции НАЧАТЬ ВВОД-ВЫВОД кроме установления связи с устройством необходима еще правильная выборка адресного и управляющего слов канала. В этом случае во внешнее устройство посылается код первой команды. Если ВУ принимает команду к исполнению, т. е. отве-

чает каналу нулевым байтом состояния, то канал формирует и посылает в ЦП код состояния 00, отключается от процессора и переходит к непосредственному выполнению заданной в УСК операции.

На этапе входа в операцию обеспечиваются также действия по организации цепочек данных и команд. Так, при цепочке команд по соответствующему сигналу от блока БСК производится выборка очередного УСК и его обработка, а также возобновляется связь с устройством, работавшим по предыдущему УСК. Аналогичные действия производятся при реализации цепочки данных.

Обмен информацией. Собственно обмен информацией между ОП и ВУ начинается сразу же после установления связи с устройством и выборки очередного УСК. В режиме записи информации во внешнее устройство производится выборка из ОП двойного слова и размещение его в регистре РБ1 блока ББП. При его передаче из РБ1 в РБ2 производится изменение содержимого счетчика байт на 8 единиц. Если после этого содержимое счетчика байт не равно нулю, то обмен информацией продолжается. Из РБ2 информация побайтно передается в интерфейс, при этом содержимое БС блока ББП с передачей каждого байта уменьшается на единицу. Когда содержимое БС становится равным нулю из РБ1 в РБ2 передается очередное двойное слово и т. д. При передаче в ВУ последнего слова вырабатывается сигнал окончания обмена как только содержимое БС станет равным содержимому РК блока буферной памяти.

В режиме считывания информации из внешнего устройства байты слов из интерфейса поступают во входной регистр блока ББП и передаются в соответствующие группы разрядов буферного регистра РБ1 по значению содержимого БС. После заполнения РБ1 двойное слово передается в РБ2 и далее в БСОП для его записи в память. Если принят последний байт последнего слова, что отмечается достижением равенства содержимого БС и регистра конца при приеме последнего слова, то вырабатывается сигнал ОБМЕН С ВУ ЗАКОНЧЕН. При обработке цепочки данных после приема последнего слова сбрасываются условия этого слова, и канал готов к приему очередного байта по новому УСК. Если же цепочка отсутствует, то после приема и передачи последнего слова вырабатывается сигнал УСК ВЫПОЛНЕНО.

Завершение операции. Операция в канале завершается в следующих случаях: при возникновении сбоев в канале, влекущих за собой прекращение работы канала, при приеме байта состояния устройства, сигнализирующего об окончании обмена, а также при формировании сигналов об окончании обмена в самом канале. Сбои, как правило, приводят к немедленному окончанию операции или процесса обмена и переключению канала в режим ожидания прерывания при выдаче в ЦП запроса на прерывание.

В случае окончания передачи информации в ВУ, а также при возникновении необычных условий в УВУ и ВУ, ведущих к прекращению обмена, устройство через интерфейс выдает в канал

байт своего состояния. В зависимости от результатов анализа этого байта в блоке БСК канал либо продолжает выполнение цепочки команд, либо переключается в режим ожидания прерывания. При поступлении в канал от устройства сигнала ТРБ-А канал также может принять байт состояния от ВУ; при этом канал переводится в состояние СУ и выдает в центральный процессор запрос на прерывание.

Прерывания. Последовательность действий при прерываниях начинается с получением от ЦП сигнала УЧЕТ ПРЕРЫВАНИЯ; канал при этом может находиться в различных состояниях. Если канал находится в рабочем состоянии, то прерывание возможно только при наличии соответствующего признака в составе выполняемого УСК. В этом случае без нарушения работы канала по обмену информацией производятся формирование и передача в память ССК, а также отключение от процессора. Если канал находится в состоянии СК, то процесс прерывания заключается в формировании ССК, выдаче его в память и отключении от процессора. Если канал находится в состоянии СУ, то при прерывании устанавливается связь с ВУ, считывается байт его состояния, после чего повторяется процедура формирования и выдачи ССК, а также отключения канала от центрального процессора.

13.4. МУЛЬТИПЛЕКСНЫЙ КАНАЛ С СЕЛЕКТОРНЫМИ ПОДКАНАЛАМИ

Назначение и общая структура мультиплексного канала ЕС-4012. Мультиплексный канал ЕС-4012 предназначен для организации обмена информацией между основной оперативной памятью и большим количеством внешних устройств малой и средней производительности, работающих параллельно и независимо друг от друга. Он представляет собой автономное устройство, включаемое в состав машины ЕС-1050. Канал связан с ЦП, ОП и интерфейсом в соответствии с общими принципами построения машин Единой системы. Обмен информацией с ОП осуществляется двойными словами, а с внешними устройствами — побайтно.

Особенностью рассматриваемого мультиплексного канала является то, что кроме 192 подканалов обычных, т. е. мультиплексных, он может иметь до четырех селекторных подканалов, выполняемых в виде модульных дополнений к основному каналу со своими блоками интерфейса. Каждый селекторный подканал может обслуживать до 16 внешних устройств. Поэтому канал ЕС-4012 позволяет обслуживать 256 внешних устройств. Если в состав ЕС-4012 включаются два селекторных подканала, то канал имеет 224 мультиплексных подканала (МПК).

Пропускная способность V мультиплексного канала зависит от загрузки селекторных подканалов и их числа. При этом считается, что максимальная пропускная способность селекторного подканала (СПК) равна 180 кбайт/с. Некоторые данные по зави-

Таблица 13.3

Пропускная способность канала

Загрузка подканалов, кбайт/с					V_i кбайт/с
СПК-1	СПК-2	СПК-3	СПК-4	МПК	
0	0	0	0	110	110
180	0	0	0	87	267
180	180	0	0	65	425
180	180	180	0	42	582
180	180	180	180	30	770

Таблица 13.4

Адресация ВУ

Наименование подканалов	Адреса ВУ
Мультиплексный	00—BF
1-й селекторный	C0—CF
2-й »	D0—DF
3-й »	E0—EF
4-й »	F0—FF

симости пропускной способности канала от загрузки селекторных подканалов приведены в табл. 13.3. Очевидно, что при включении в состав мультиплексного канала только двух СПК его максимальная пропускная способность равна 425 кбайт/с.

Адресация внешних устройств, подключаемых к мультиплексному каналу, зависит от количества используемых селекторных подканалов. В табл. 13.4 дано распределение адресов ВУ по подканалам для случая четырех СПК; адреса записаны в шестнадцатиричной системе счисления.

Если МК не укомплектован частью селекторных подканалов, то отвечающие им адреса передаются ВУ, подключаемым к мультиплексному подканалу.

Рассматриваемый МК, как и селекторный канал ЕС-4035, управляется всеми четырьмя инструкциями ввода-вывода, а программа его работы может состояться из команд записи, считывания, считывания в обратном направлении, управления, уточнения состояния и перехода в канале. Общая схема канала приведена на рис. 13.11. Канал имеет в своем составе следующие основные части:

- блок главного канала (БГК);
- блок мультиплексного подканала (БМПК);
- селекторные подканалы (СПК);
- запоминающее устройство ЕС-3415, или память мультиплексного канала (ПМК);
- блок магистральных усилителей (БМУ).

Особенностью данного МК является выделение в его составе блока главного канала, выполняющего функции управления работой всех подканалов и связи с центральным ядром системы, т. е. с процессором и оперативной памятью. В то же время связи канала с ОП и ЦП аналогичны связям селекторного канала ЕС-4035; обмен информацией с ОП осуществляется двойными словами, с внешними устройствами — побайтно и т. д.

Главный канал предназначен для приема управляющей информации, т. е. признаков инструкций ввода-вывода, АСК, УСК, запросов на обслуживание, формирования управляющих слов устройств (УСУ), передачи информации от оперативной памяти

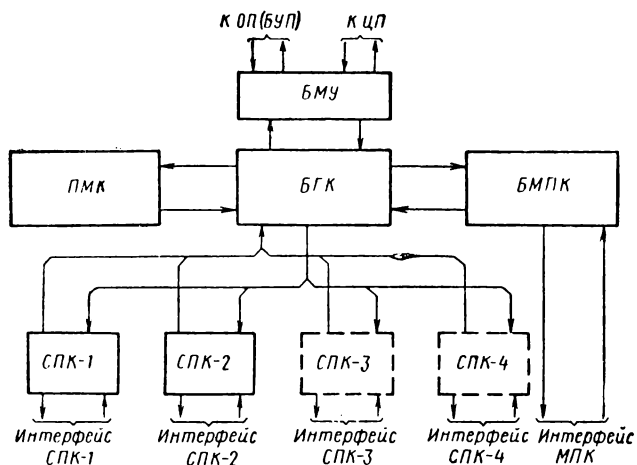


Рис. 13.11. Мультиплексный канал с селекторными подканалами

к ВУ через подканалы и от ВУ в оперативную память, обслуживая запросы на прерывания и установления очередности обслуживания подканалов и ЦП. Обслуживание подканалов и центрального процессора производится в соответствии с приоритетами запросов:

- 1-й приоритет — 1-й селекторный подканал;
- 2-й приоритет — 2-й селекторный подканал;
- 3-й приоритет — 3-й селекторный подканал;
- 4-й приоритет — 4-й селекторный подканал;
- 5-й приоритет — мультиплексный подканал;
- 6-й приоритет — центральный процессор.

Для выполнения указанных функций главный канал включает в свой состав регистры для хранения данных и управляющей информации, сумматор для изменения адресов и счета данных, регистры связи с мультиплексной памятью, узел управления, логические и другие схемы. По своей схеме главный канал аналогичен управляющей части мультиплексного канала однородной структуры, но дополнительно включает схему приоритетов запросов. В рассматриваемом МК главный канал обеспечивает формирование четырех УСУ, имеющих формат двойных слов; содержимое этих УСУ аналогично содержимому УСУ для МК, рассмотренного в п. 13.2.

Блок мультиплексного подканала предназначен для управления обменом информацией между главным каналом и внешними устройствами, подключенными к интерфейсу МПК. Он содержит ряд 9-разрядных регистров (8 информационных + 1 контрольный разряд), служащих для промежуточного хранения байт данных, команд и адресов ВУ, пересылаемых из главного канала в ВУ и обратно.

Селекторный подканал — это своеобразное модульное дополнение к мультиплексному каналу со своим интерфейсом ввода-вывода, предназначенное для управления обменом информацией между главным каналом и внешними устройствами, работающими в монопольном режиме при скоростях передачи информации до 180 кбайт/с. Он содержит ряд регистров и схем, позволяющих объединять байты в двойные слова, разделять двойные слова на байты и направлять данные либо в главный канал, либо в интерфейс.

Память мультиплексного канала, или устройство ЕС-3415, представляет собой оперативное запоминающее устройство на ферритовых сердечниках типа 2D емкостью 1024 72-разрядных слова; полный цикл работы памяти около 1 мкс; время выборки слова $T_{\text{выб}} \leq 0,5$ мкс. Емкость мультиплексной памяти обеспечивает одновременное хранение управляющих слов для всех подключаемых к мультиплексному каналу внешних устройств.

Блок магистральных усилителей предназначен для приема и выдачи информации, передаваемой по общим шинам между каналом и процессором, а также каналом и блоком управления оперативной памятью машины. Он содержит в своем составе необходимое количество усилителей — приемников и усилителей — передатчиков.

Работа мультиплексного канала. Работа канала по выполнению инструкции ввода-вывода начинается с приема из ЦП признака инструкции и адреса внешнего устройства. Затем из процессора поступает сигнал по линии выборки канала. По этому сигналу определяется состояние канала; если он работает в монопольном режиме, неработоспособен или выполняется инструкция ПРОВЕРИТЬ КАНАЛ, то в ЦП выдается соответствующий код состояния, и выполнение инструкции на этом заканчивается. Если канал свободен или находится в состоянии ожидания прерывания и заданная ЦП инструкция не является инструкцией ПРОВЕРИТЬ КАНАЛ, то возбуждается запрос на обслуживание ЦП главным каналом.

Обслуживание запросов ЦП выполняется главным каналом только при отсутствии других запросов. В результате этого обслуживания проверяется состояние подканала; если подканал занят, то главный канал выдает в ЦП соответствующий код состояния и заканчивает на этом выполнение инструкции. Если же подканал свободен и выполняется инструкция НАЧАТЬ ВВОД-ВЫВОД, то в МПК или один из СПК выдаются адрес ВУ и указание о проведении начальной выборки этого устройства; одновременно в ОП выдается запрос на выборку АСК.

Главный канал организует последовательную выборку управляющих слов, т. е. АСК и УСК. Одновременно от ВУ поступает либо байт состояния, либо его ответный адрес. В первом случае главный канал формирует код состояния и отправляет его в ЦП,

заканчивая тем самым исполнение инструкции. Во втором случае главным каналом сравниваются адреса устройств. При несовпадении адресов формируются код состояния и слово состояния канала; отсылкой ССК в оперативную память для этого случая заканчивается исполнение инструкции ввода-вывода.

При совпадении адресов главный канал через соответствующий подканал передает в ВУ код операции. В ответ из ВУ должен поступить байт состояния. Отличие этого байта от нуля указывает на невозможность выполнения операции; в ЦП посылается код состояния 01. Если байт состояния является нулевым, то в ЦП посылается код состояния 00; канал и устройство готовы к выполнению операции записи или считывания.

Обслуживание запросов на передачу информации зависит от того, какой подканал работает. Если поступил запрос от ВУ, подключенного к мультиплексному подканалу, на запись информации, то главный канал после учета этого запроса выдает один байт данных устройству, изменяет счет данных и адрес в УСУ; после этого осуществляется проверка на конец слова и на конец массива. Если операция записи не закончена, то обслуживание заканчивается; очередной байт будет передан в ВУ только по следующему запросу. Если закончена запись слова, то из главного канала в ОП посылается запрос на выборку следующего слова; ~~это слово в составе УСУ данного ВУ направляется для хранения~~ в мультиплексную память; канал ждет поступления следующего запроса от ВУ. Если закончена запись всего массива, то операция заканчивается при отсутствии цепочки данных; при наличии этой цепочки производится выборка нового УСК и по нему слов нового массива данных. Работа канала при выполнении операции считывания данных протекает аналогично случаю записи; основное отличие заключается в направлении передачи информации.

При работе канала с ВУ, подключенного к селекторному подканалу, главный канал управляет в основном обменом двойными словами с ОП. Разделение слов на байты и объединение байт в слова производятся непосредственно в селекторном подканале.

Запросы на прерывание работы процессора обрабатываются в рассматриваемом мультиплексном канале подобно аналогичным действиям в селекторном канале (см. п. 13.3). Отличием является то, что запрос на прерывание, после его фиксации в канале, отсылается для хранения в ВУ, запрашивающее прерывание; это связано с работой системы приоритетов главного канала.

ОПЕРАЦИОННЫЕ СИСТЕМЫ

14.1. СТРУКТУРА ДИСКОВОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ И НАЗНАЧЕНИЕ ОТДЕЛЬНЫХ ЕЕ КОМПОНЕНТОВ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Дисковая операционная система (ДОС ЕС) предназначена для повышения эффективности функционирования ЕС ЭВМ (за счет увеличения объема вычислительной работы, выполняемой машиной в единицу времени), автоматизация процесса создания программ, сокращения периода времени между постановкой задачи на решение и получением результатов; ДОС облегчает работу всем, кто имеет отношение к использованию ЕС ЭВМ: программистам, операторам, обслуживающему персоналу.

Применение ДОС дает возможность программисту:

- разбивать задачу на части, записывать каждую часть на наиболее подходящем языке и затем объединять их в готовую к выполнению программу;
- автоматизировать процесс отладки программ;
- помещать отлаженную программу в системную библиотеку и получать доступ к этой программе;
- при необходимости делить большую программу на сегменты, которые могут выполняться автономно и одновременно;
- выполнять программу непосредственно после ее трансляции;
- использовать стандартные процедуры ввода-вывода для работы с данными.

Оператор, благодаря ДОС, получает больше возможности по управлению вычислительным процессом, а обслуживающему персоналу представляется информация, необходимая для постоянного контроля за состоянием машины и самой операционной системы.

Прежде чем рассматривать структуру ДОС ЕС, остановимся на основных понятиях и определениях, которые в дальнейшем будут широко использованы.

Задание — основная независимая единица работы ЭВМ, задаваемая извне программистом, оператором (вообще: потребителем машинного времени, абонентом). Задания не зависят друг от друга и могут выполняться одновременно. Каждое задание обычно

описывается с помощью нескольких управляющих операторов, определяющих, например, имя задания, начало и конец задания, название программы, которая должна быть выполнена. Задания могут быть либо одиночными, либо сгруппированными в один пакет, который образует входной поток пакетированных заданий.

Шаги задания — это части задания, на которые по усмотрению программиста может разбиваться задание. Например, задание на работу, предусматривающую трансляцию, редактирование и выполнение программы, состоит из следующих шагов: выполнение требуемого транслятора, выполнение программы РЕДАКТОР, выполнение самой программы. Шаги, составляющие одно задание, выполняются последовательно, причем выполнение очередного шага зависит от того, насколько успешно реализован один или несколько предыдущих шагов. Простейшее задание состоит из одного шага.

Управление заданиями — совокупность процедур, включающая прием заданий, их контроль, подготовку запрашиваемых программ к выполнению, запуск этих программ и автоматический переход к очередному заданию. Управление заданиями осуществляется управляющей программой операционной системы.

Задача — это работа, которая должна быть выполнена программой, указанной в шаге задания. Для решения задачи операционная система должна выделить ей необходимые ресурсы: время процессора, определенную емкость оперативной и внешней памяти, устройства ввода-вывода и т. д. Задача может решаться в однопрограммном или в мультипрограммном режиме. В первом случае машина решает только одну задачу и все ее ресурсы находятся в распоряжении этой задачи. Во втором случае машина одновременно решает несколько (в ДОС ЕС до трех) независимых задач и ее ресурсы с помощью операционной системы распределяются между программами этих задач. Такой режим работы называется мультипрограммированием с фиксированным числом задач (программ).

Раздел — часть области основной оперативной памяти, предоставляемой проблемным (рабочим) программам потребителей (абонентов). Мультипрограммирование с фиксированным числом задач (программ) иначе называется мультипрограммированием с фиксированным числом разделов. Таких разделов может быть один, два или три, причем каждый из них отводится для одной проблемной программы. Размеры разделов определяются при создании ДОС для данной машины и при необходимости могут быть изменены оператором. Если область оперативной памяти, выделенная для проблемных программ, состоит из одного раздела, машина может работать только в однопрограммном режиме. Разделы имеют названия: фоновый, первый раздел переднего плана F1, второй раздел переднего плана F2.

Фоновая программа — это проблемная программа, которая при ее выполнении размещается в фоновом разделе. Фоновые прог-

раммы составляются для фоновых задач. Проблемные программы, размещенные в разделах F2 и F1, называются соответственно программой второго раздела переднего плана и программой первого раздела переднего плана.

Запись (или: логическая запись) — совокупность единиц информации. Форматы записей, организация их в физические блоки, способы хранения и поиска отдельных записей определяются разработчиками ДОС ЕС. Сама же обработка, поиск, хранение, доставка записей из ВЗУ и ОЗУ осуществляется программами операционной системы.

Файл — совокупность записей, объединенных по некоторому общему признаку (или признакам). Файлы данных размещаются в ВЗУ.

Том — емкость стандартного блока внешней памяти. Например, это емкость одной катушки магнитной ленты или емкость одного пакета магнитных дисков и т. д.

Многофайловый том — это том, содержащий несколько файлов информации.

Многотомный файл — это файл, располагающийся в нескольких томах.

Метка файла — совокупность характеристик файла (имя файла, его границы, формат и др.), позволяющих операционной системе его опознать. Аналогичное понятие (метка тома) имеется применительно к тому.

Метод логических устройств — это метод, обеспечивающий независимость программных компонентов ДОС ЕС и проблемных программ от конкретных физических адресов внешних устройств (ВЗУ, УВВ). Устанавливается стандартный набор символических имен, принятых для обозначения логических устройств, которые программист использует в исходной программе во всех случаях, когда необходимо обратиться к внешнему устройству. Следовательно, в программе указывается логическое устройство, а не конкретный физический адрес требуемого внешнего устройства. Перед выполнением программы на машине программист должен назначить каждому логическому устройству конкретный физический адрес внешнего устройства. Количество логических устройств может не совпадать с количеством физических внешних устройств, подключенных к машине. Символическое имя, используемое для обозначения логического устройства, имеет вид SYSXXX, где XXX может принимать определенные буквенные или числовые значения (в последнем случае от 000 до максимального количества логических устройств, обслуживаемых операционной системой).

Системные логические устройства — это логические устройства, которые используются операционной системой для собственных надобностей. Для них символические имена зафиксированы постоянно и каждому из них ставится в соответствие конкретный тип физических внешних устройств.

Основными системными логическими устройствами являются: SYSRES — запоминающее устройство на магнитных дисках, где во время работы машины постоянно размещается ДОС ЕС (резиденция системы);

SYSRDR — устройство, которое используется управляющей программой ДОС ЕС для ввода управляющих операторов задания;

SYSIPT — устройство, используемое компонентами ДСС ЕС для ввода данных (системный ввод);

SYSPCH — устройство, используемое компонентами ДОС ЕС для вывода данных на карты (системный перфоратор);

SYSLST — устройство, которое используется компонентами ДОС ЕС для вывода данных на печать (системная печать);

SYSLOG — устройство связи с оператором.

Кроме этих системных логических устройств, необходимых для работы управляющей программы операционной системы, имеются системные логические устройства, используемые другими компонентами ДОС ЕС.

Операционная система ДСС ЕС дает возможность программисту использовать языки программирования, ориентированные на различные классы задач. К ним относятся: ФОРТРАН — алгоритмический язык для научно-технических задач с преобладанием вычислений, РПГ — язык программирования для задач обработки данных и ПЛ/1 — универсальный язык программирования. Все это — проблемно-ориентированные языки, не учитывающие параметры и характеристики конкретных машин. Кроме того, применяется машинно-ориентированный язык АССЕМБЛЕР, который позволяет использовать все возможности машины и вместе с тем освобождает программиста от выполнения таких операций, как запоминание машинных кодов команд, вычисление действительных адресов памяти и т. п.

В ДОС ЕС программа, составленная на любом из указанных языков программирования, транслируется в так называемый объектный модуль.

Объектный модуль — это программный модуль в промежуточном, общем для всех трансляторов системы, формате. В объектном модуле специфика исходного языка программирования теряется. Однако это еще не машинная программа, для непосредственного выполнения на машине объектный модуль не годится. Для этого объектный модуль должен пройти этап редактирования, после которого получается программная фаза (абсолютный модуль).

Абсолютный модуль — это программный объект, представленный на машинном языке и не подлежащий дроблению при вызове его в основную оперативную память для выполнения.

Все программы в зависимости от того, в какой стадии подготовки они находятся (исходный модуль, представленный на одном из указанных алгоритмических языков, объектный модуль, абсолютный модуль, или фаза), могут храниться в соответствующей библиотеке. В ДОС ЕС имеются библиотеки трех типов: библи-

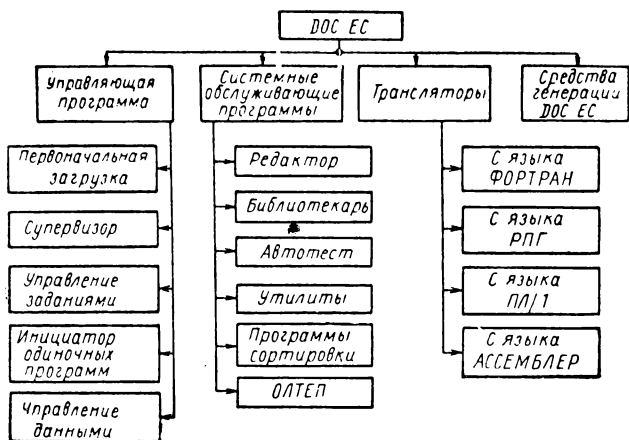


Рис. 14.1. Структура дисковой операционной системы ЕС ЭВМ

отека исходных модулей, библиотека объектных модулей и библиотека абсолютных модулей. В библиотеке абсолютных модулей содержатся в виде фаз программные компоненты самой операционной системы, а также готовые к выполнению программы абонентов, поэтому для функционирования ДОС ЕС наличие этой библиотеки обязательно (две другие библиотеки могут отсутствовать). Любая программа поступает в оперативную память для выполнения только из библиотеки абсолютных модулей.

Конкретная операционная система представляет собой *резидентный файл*, который содержит либо только библиотеку абсолютных модулей, либо эту библиотеку в любом сочетании с остальными двумя библиотеками. Резидентный файл обычно занимает весь пакет магнитных дисков, отведенный под резиденцию системы.

В ДОС ЕС могут существовать как системные, так и личные (принадлежащие конкретным абонентам) библиотеки объектных и исходных модулей. Существование личной библиотеки абсолютных модулей не допускается, так как для этого потребовалась бы слишком большая емкость запоминающих устройств.

Рассмотрим теперь структуру ДОС ЕС (рис. 14.1). Основными программными компонентами дисковой операционной системы являются [20]:

- управляющая программа;
- системные обслуживающие программы;
- трансляторы для языков АССЕМБЛЕР, ФОРТРАН, РПГ, ПЛ/1;
- средства генерации операционной системы.

Управляющая программа обеспечивает автоматическое выполнение программ (как проблемных, так и программ самой операционной системы) на машине, т. е. она обеспечивает автоматичес-

кий переход от выполнения одной программы к выполнению другой в рамках одного задания или в пределах различных заданий.

Управляющая программа функционально разделяется на следующие части:

— программа ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА, обеспечивающая подготовку ДОС ЕС к работе;

— СУПЕРВИЗОР, осуществляющий управление всем вычислительным процессом на машине;

— программа УПРАВЛЕНИЕ ЗАДАНИЯМИ, обеспечивающая подготовку ДОС ЕС для выполнения пакета заданий;

— ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ, с помощью которого осуществляется подготовка одиночных программ к выполнению в мультипрограммном режиме;

— УПРАВЛЕНИЕ ДАННЫМИ (система управления вводом-выводом), включающая программы для управления процессом обмена данными между основной оперативной памятью и внешними устройствами.

В состав системных обслуживающих программ входят:

— РЕДАКТОР, обеспечивающий получение программ, готовых к выполнению на машине;

— БИБЛИОТЕКАРЬ, выполняющий функции корректировки и обслуживания библиотек операционной системы;

— АВТОТЕСТ, или отладочная программа, назначение которой состоит в оказании помощи программисту при отладке программ;

— УТИЛИТЫ — это набор программ, обеспечивающих выполнение таких часто повторяющихся вспомогательных операций, сущность которых не зависит от специфики машины и формата данных индивидуального абонента. К этим операциям относятся перемещение файлов с одного внешнего накопителя информации на другой, копирование файлов, печать и перфорация файлов и др.;

— программы сортировки, с помощью которых осуществляется:

а) сортировка файлов, состоящих из неупорядоченных записей;

б) объединение нескольких файлов с упорядоченными записями в один упорядоченный последовательно организованный файл;

— программа ОЛТЕП, предназначенная для проверки правильности функционирования устройств ввода-вывода.

ДОС имеет модульную структуру, позволяющую потребителям приспособлять ее к конкретной конфигурации технических устройств ЕС ЭВМ. Следовательно, в зависимости от имеющегося в распоряжении потребителя набора и структуры технических средств ЕС ЭВМ он по своему желанию может включать или не включать ту или иную компоненту операционной системы. Процесс создания конкретной ДОС, соответствующей составу оборудования машины и области ее применения, называется *генерацией*

операционной системы. Для реализации этого процесса необходимы средства генерации, представляющие собой совокупность соответствующих программ и правил.

Конкретная ДОС, полученная в результате генерации операционной системы, постоянно размещается на магнитных дисках. Пакет магнитных дисков, на которых хранятся программы ДОС, называется *резиденцией системы.*

14.2 УПРАВЛЯЮЩАЯ ПРОГРАММА ДОС ЕС. ПРОГРАММЫ ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА И СУПЕРВИЗОР

Основная роль в автоматизации процесса выполнения программ на машине принадлежит управляющей программе, компоненты которой включаются в работу для выполнения своих функций на соответствующих этапах этого процесса.

Функционирование ДОС начинается с выполнения процедуры первоначальной загрузки, назначение которой заключается в том, чтобы привести машину и ее операционную систему в состояние готовности к работе. Выполнение этой процедуры начинается с подготовительных действий оператора, который:

— устанавливает пакет дисков с записанным в нем резидентным файлом в соответствующий накопитель на магнитных дисках (НМД);

— набирает адрес этого НМД на переключателях загрузки, расположенных на пульте оператора;

— нажимает кнопку «Загрузка».

При этом автоматически в основную оперативную память (ООП) из резиденции системы вводится программа ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА, которая прежде всего производит перепись в основную память ядра СУПЕРВИЗОРА. Ядром СУПЕРВИЗОРА называется часть СУПЕРВИЗОРА, в ходе вычислительного процесса постоянно присутствующая в основной памяти. После загрузки ядра СУПЕРВИЗОРА могут выполняться и другие функции программы ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА, из которых главная — указание операционной системе конкретной конфигурации устройств ввода-вывода (УВВ) данной ЭВМ.

Оператор имеет возможность оперативно влиять на состав УВВ, с которыми будет оперировать ДОС. Для этого он с помощью пультовой пишущей машинки дает директивы, т. е. указания, касающиеся добавления в систему новых устройств или исключения тех из них, которые не будут использоваться в течение длительного времени. Эти директивы воспринимаются программой ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА и записываются в соответствующую таблицу, которая хранится в основной памяти. Они не вносятся в резидентный файл, поэтому сохраняют свою силу только до выполнения новой процедуры первоначальной загрузки.

Программа ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА заканчивает свою работу передачей управления СУПЕРВИЗОРУ, а он, в свою очередь, вызывает в основную память программу УПРАВЛЕНИЕ ЗАДАНИЯМИ, которая настраивается на прием первого задания в фоновом разделе.

СУПЕРВИЗОР представляет собой набор программ (фаз), обеспечивающих контроль выполнения проблемных программ на всех этапах, начиная с момента ввода задания на их выполнение и до получения результатов работы. СУПЕРВИЗОР постоянно хранится в библиотеке абсолютных модулей в резиденции операционной системы. Ядро СУПЕРВИЗОРА постоянно хранится в специальной области ООП, которая называется *областью управляющей программы*. Остальные фазы СУПЕРВИЗОРА, называемые *транзитами*, по мере необходимости вызываются в ООП ядром СУПЕРВИЗОРА для выполнения. Постоянного присутствия транзитов в основной памяти не требуется. Для записи транзитов в ООП выделяется так называемая *транзитная область управляющей программы*, которая является частью области управляющей программы. Оставшаяся часть ООП образует область проблемных программ.

В области управляющей программы выделяется участок, называемый *областью связи*, где содержится информация, которая в ходе вычислительного процесса может потребоваться СУПЕРВИЗОРУ и проблемным программам. Это — дата выполнения проблемной программы, имя этой программы, адрес начала и конца области проблемных программ и т. д.

СУПЕРВИЗОР включается в работу по сигналам прерывания и обеспечивает выполнение следующих функций:

- распознавание и обработка прерываний;
- планирование работы каналов;
- организация связи с оператором;
- вызов программных фаз из библиотеки абсолютных модулей;
- обработка сбоев внешних устройств, программных и машинных сбоев;
- выполнение процедур, связанных с окончанием выполнения задания;
- обеспечение мультипрограммного режима работы, при котором одновременно выполняются до трех проблемных программ;
- обслуживание таймера (электронных часов);
- создание контрольных точек.

В ЕС ЭВМ существует несколько типов прерываний: обращение к СУПЕРВИЗОРУ, программный сбой, машинный сбой, прерывание ввода-вывода, внешнее прерывание. Все прерывания принимаются СУПЕРВИЗОРОМ.

Получив управление после прерывания, СУПЕРВИЗОР запоминает полное состояние прерванной программы, что необходимо для обеспечения возможности продолжить ее выполнение после обработки прерывания. Состояние прерванной программы запо-

минается в специальной области основной памяти, называемой *областью сохранения*. В режиме мультипрограммной работы каждый раздел ООП имеет свою область сохранения.

После установления типа прерывания и запоминания состояния прерванной программы управление передается определенной подпрограмме СУПЕРВИЗОРА, которая производит обработку этого прерывания. Обработка заключается в уточнении причины прерывания и выполнении определенных действий. После обработки прерывания СУПЕРВИЗОР определяет программу, которая может быть запущена, и передает ей управление.

Прерывание «ОБРАЩЕНИЕ К СУПЕРВИЗОРУ» возникает в случае, когда в выполняемой программе встречается команда «ОБРАЩЕНИЕ К СУПЕРВИЗОРУ». На языке АССЕМБЛЕР код этой команды SVS. В общем случае эта команда используется в программе для того, чтобы запросить выполнение функций, обеспечиваемых операционной системой, например, таких, как выполнение операции ввода-вывода, загрузка очередной фазы программы, открытие или закрытие файла и др. Указание, какая из функций ДОС требуется программе, содержится в самой команде SVS.

В случае появления сигнала ПРОГРАММНЫЙ СБОЙ СУПЕРВИЗОР обеспечивает выбор одного из следующих вариантов продолжения:

- передача управления для обработки программного сбоя подпрограмме выполняемой проблемной программы. Этот вариант реализуется в случае, когда с помощью соответствующей макрокоманды СУПЕРВИЗОРУ заранее (до появления сбоя) сообщается о наличии в проблемной программе подпрограммы обработки сбоев;
- разгрузка памяти (освобождение раздела памяти от выполняемого задания) и прекращение выполнения задания;
- прекращение выполнения задания.

Особенность прерываний по программному сбою заключается в том, что некоторые из этих прерываний могут запрещаться проблемной программой с помощью команды УСТАНОВИТЬ МАСКУ ПРОГРАММЫ. При наличии такой команды прерывание, вызванное программным сбоем, игнорируется. Появление сигнала МАШИННЫЙ СБОЙ, возникающего в результате неисправности машины, приводит операционную систему в состояние ожидания, все прерывания запрещаются. Продолжение работы операционной системы невозможно, и для ее восстановления необходима процедура первоначальной загрузки.

Прерывание ввода-вывода может быть вызвано следующими причинами:

- освобождением канала (при этом появляется сигнал КАНАЛ КОНЧИЛ);
- освобождением устройства управления УВВ (сигнал УУ КОНЧИЛО);
- освобождением УВВ (сигнал УВВ КОНЧИЛО);

— нажатием кнопки «ВНИМАНИЕ» (сигнал ВНИМАНИЕ);
— выполнением канальной команды ввода-вывода (сигнал ПРОГРАММНО-УПРАВЛЯЕМОЕ ПРЕРЫВАНИЕ).

Обработка прерываний по указанным сигналам ввода-вывода производится планировщиком каналов, являющимся одной из составных частей СУПЕРВИЗОРА.

Причинами внешних прерываний являются появление внешних сигналов, сигналов от таймера, сигналов от кнопки «ПРЕРЫВАНИЕ» на пульте оператора. Прерывания, вызванные внешними сигналами, игнорируются, т. е. после распознавания такого сигнала СУПЕРВИЗОР возвращает управление прерванной программе. Обработка внешних прерываний, вызванных другими сигналами, рассматривается ниже.

Вся работа с внешними устройствами проходит через программу, которая называется *планировщиком каналов*. В функции этой программы входит планирование запросов на выполнение операций ввода-вывода (установление очередности обращения к внешним устройствам и каналам), запуск операций ввода-вывода, обработка всех прерываний ввода-вывода, возникающих при нормальном завершении операции, при обнаружении ошибки, достижении конца файла и т. п., обработка сбоев ввода-вывода, контроль корректности работы с устройствами прямого доступа (с накопителями на магнитных дисках).

В планировщике каналов содержится информация о том, какие внешние устройства должны быть задействованы, какая нагрузка имеется на каждое устройство, в каком состоянии находятся устройства и каналы в каждый момент времени. Эта информация необходима для обеспечения эффективной загрузки внешних устройств и каналов.

Рассмотрим подробнее порядок выполнения своих функций планировщиком каналов, предварительно отметив функции проблемной программы при выполнении операций ввода-вывода. Функции эти заключаются в следующем:

- подготовка канальной программы;
- формирование блока управления данными. В этом блоке СУПЕРВИЗОРУ (планировщику каналов) передается необходимая для выполнения операции информация;
- выдача СУПЕРВИЗОРУ запроса на выполнение операции ввода-вывода;
- обработка информации о результатах выполнения операции, помещаемой СУПЕРВИЗОРОМ в блок управления данными по окончании операции ввода-вывода.

Программирование ввода-вывода осуществляется обычно на так называемом *логическом уровне*, когда программист может не знать специфику работы внешних устройств. Он заботится лишь о логической структуре своих данных. Все функции по обмену данными между внешними накопителями и основной памятью выполняются операционной системой. Логический уровень про-

граммирования ввода-вывода реализуется совокупностью программ, называемых логической системой управления вводом-выводом, или планировщиком каналов.

Одной из функций планировщика каналов, как уже указывалось, является управление очередью запросов к внешним устройствам (УВВ и ВЗУ). При запросе со стороны проблемной программы на операцию ввода-вывода в сформированном до этого блоке управления данными СУПЕРВИЗОРУ сообщается, на каком логическом устройстве должна выполняться эта операция. Задача СУПЕРВИЗОРА — выяснить, какое физическое устройство соответствует указанному логическому устройству. Для этого в СУПЕРВИЗОРЕ имеются таблицы, определяющие однозначное соответствие между логическими и физическими устройствами. Используя эти таблицы, СУПЕРВИЗОР определяет адрес требуемого физического внешнего устройства и помещает запрос на выполнение операции ввода-вывода в очередь запросов к этому устройству.

После постановки запросов в очередь СУПЕРВИЗОР анализирует одну из своих таблиц, в которой указаны характеристики и состояние выбранного физического устройства. Если устройство свободно, СУПЕРВИЗОР формирует адресное слово канала и выдает команду НАЧАТЬ ВВОД-ВЫВОД. В противном случае управление возвращается проблемной программе для ее продолжения, и ввод-вывод выполняется после того, как все предыдущие запросы, ранее поставленные в очередь, будут удовлетворены.

Обработка прерываний ввода-вывода заключается в анализе информации, помещенной в слове состояния канала (CSW), формируемом аппаратно в момент прерывания ввода-вывода. В этом слове отражена причина прерывания и состояние внешнего оборудования — канала, устройства управления УВВ или ВЗУ, внешнего устройства. При появлении одного из сигналов КАНАЛ КОНЧИЛ, УУ КОНЧИЛО или УВВ КОНЧИЛО, если он не сопровождается установкой в слове CSW битов, указывающих на наличие сбойной ситуации, СУПЕРВИЗОР фиксирует завершение операции ввода-вывода и исключает из очереди запрос на эту операцию. Затем осуществляется переход к удовлетворению следующего запроса. Если же операция ввода-вывода не смогла нормально завершиться из-за наличия сбоя, управление передается подпрограмме обработки сбоев. Прерывание, вызванное сигналом ВНИМАНИЕ, обрабатывается лишь в случае, если этот сигнал поступил с пультовой пишущей машинки. Обработка такого прерывания сводится к установлению связи оператора с подпрограммой СУПЕРВИЗОРА, которая предназначена для приема и обработки директив оператора. Программно-управляемые прерывания используются программами, организующими работу с устройствами телеобработки.

В ДОС ЕС для некоторых системных логических устройств допускается назначение физических устройств прямого доступа

(НМД). При этом на системном логическом устройстве используется только последовательный метод организации файла, который характеризуется тем, что адрес данных на диске от обращения к обращению непрерывно возрастает. СУПЕРВИЗОР обеспечивает контроль адресов данных, указанных в командах обращения к диску. При нарушении непрерывности возрастания адресов выполнение проблемной программы прекращается. В этом и заключается функция контроля корректности работы с устройствами прямого доступа.

В случае появления сбоев при выполнении операций ввода-вывода действия, предпринимаемые СУПЕРВИЗОРОМ, зависят от вида сбоя. Так, сбой оборудования канала и сбой интерфейса приводят к переводу машины в состояние ожидания, выход из которого осуществляется только через процедуру первоначальной загрузки. Последствием сбоев, вызванных программными ошибками и нарушением защиты памяти, является прекращение выполнения проблемной программы.

При других видах сбоев реакция СУПЕРВИЗОРА зависит от специфики внешнего устройства, в котором произошел сбой, от момента сбоя, состояния канала, устройства управления ВЗУ или УВВ и внешнего устройства в этот момент, от режима обработки сбоев, установленного проблемной программой. Эта реакция может быть такой:

- попытка избавиться от сбоя. В любом случае избавление от сбоя заключается в повторном выполнении либо всей канальной программы, либо только ее части, начинающейся с команды, при выполнении которой произошел сбой;

- игнорирование сбоя, если он не приводит к искажению информации и не влияет на ход выполнения программы;

- возврат управления проблемной программе для принятия необходимых мер по устранению последствий сбоя;

- прекращение выполнения задания, если СУПЕРВИЗОР не может предпринять ни одной из указанных мер.

Принятие решения по тому или иному виду сбоя производится СУПЕРВИЗОРОМ на основании анализа информации, содержащейся в блоке управления данными и в слове состояния канала, а также информации, полученной по команде уточнения состояния внешнего устройства.

Одна из функций СУПЕРВИЗОРА — вызов программных фаз из библиотеки абсолютных модулей в основную память. В этой библиотеке могут храниться программы двух типов:

- *абсолютные* — записываются только в ту область основной памяти, для которой они были отредактированы программой РЕДАКТОР. Ядро СУПЕРВИЗОРА, большинство системных обслуживающих программ и проблемных программ являются абсолютными;

- *самоперемещающиеся* — могут записываться для выполнения в требуемую область основной памяти даже в том случае, если

они были отредактированы для другой области. Транзиты управляющей программы являются самоперемещающимися.

Загрузка в основную память обоих типов готовых к выполнению фаз, независимо от их принадлежности к фазам управляющей программы, обслуживающих программ или проблемных программ, осуществляется подпрограммой ядра СУПЕРВИЗОРА, которая называется *системным загрузчиком*.

После окончания выполнения проблемной программы управление передается СУПЕРВИЗОРУ для реализации процедур *завершения шага задания* или всего задания. Управление передается как в случае нормального завершения проблемной программы, так и в случае ненормального завершения, обусловленного программным сбоем, неисправимым сбоем ввода-вывода, требованием оператора о прекращении выполнения программы, сообщением самой программы о невозможности продолжения ее выполнения. Выполнив процедуры завершения, СУПЕРВИЗОР передает управление программе УПРАВЛЕНИЕ ЗАДАНИЯМИ для приема следующего шага задания или нового задания. При ненормальном завершении программы прекращается выполнение всего задания, выводится сообщение о причине прекращения выполнения задания, передается управление программе УПРАВЛЕНИЕ ЗАДАНИЯМИ для приема следующего задания.

Если проблемная программа выполняется в течение длительного времени, СУПЕРВИЗОР дает возможность в процессе ее выполнения периодически создавать *контрольные точки*. Контрольные точки создаются по требованию программиста с помощью специальных подпрограмм СУПЕРВИЗОРА, хранящихся в библиотеке абсолютных модулей. Контрольная точка содержит информацию о состоянии программы и системы в момент ее создания. Эта информация используется для запуска прерванной по какой-либо причине проблемной программы с некоторой промежуточной точки, а не с самого начала. Это позволяет избежать непроизводительных потерь времени на повторение той части программы, которая ранее была выполнена правильно.

ДОС обеспечивает двустороннюю *связь с оператором*, предоставляя возможность оператору, с одной стороны, получать сведения о состоянии системы в процессе ее функционирования и, с другой стороны, управлять работой системы, выдавая ей директивные указания. Эта связь заключается в обмене информацией между программами операционной системы и оператором, причем для этого используется пультовая пишущая машинка. Все сообщения оператору система выдает на пишущую машинку и с машинки принимает ответы оператора и директивы.

Сообщения, поступающие от системы к оператору, имеют одинаковую структуру. Они носят либо чисто информационный характер, либо представляют собой запрос на выполнение оператором некоторых действий.

Для программ операционной системы оператор вводит информацию либо в виде директивных указаний, либо в виде ответов на сообщения этих программ. Директивные указания выдаются оператором в тех случаях, когда необходимо прекратить выполнение задания, начать, приостановить или запустить заново пакетную обработку в любом из разделов основной памяти, запустить одиночную программу в любом разделе переднего плана, назначить логическим устройствам физические внешние устройства, приостановить вычислительный процесс после выполнения задания или шага задания, напечатать таблицу распределения основной памяти, перераспределить область проблемных программ между разделами и т. д.

Директивные указания могут передаваться как управляющей программе ДОС, так и тем из обслуживающих и проблемных программ, которые имеют специальные подпрограммы для принятия и обработки таких указаний. Директивные указания, воспринимаемые и обрабатываемые управляющей программой, подразделяются на директивы программы ВНИМАНИЕ, директивы программы ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ и директивы программы УПРАВЛЕНИЕ ЗАДАНИЯМИ.

Прежде чем ввести директиву в систему, оператор должен установить связь с соответствующей программой. Связь с программой ВНИМАНИЕ устанавливается путем нажатия кнопки «ВНИМАНИЕ» на пульте пишущей машинки, что вызывает прерывание ввода-вывода, после обработки которого СУПЕРВИЗОР вводит в транзитную область программу ВНИМАНИЕ. Эта программа настраивается на прием и обработку директив оператора. Связь с программой ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ устанавливается с помощью директивы START, которая принимается к исполнению после нажатия кнопки «ВНИМАНИЕ». Наконец, связь с программой УПРАВЛЕНИЕ ЗАДАНИЯМИ устанавливается сначала автоматически (после процедуры первоначальной загрузки), а впоследствии — с помощью соответствующих директив или управляющих операторов.

В составе операционной системы имеется таймер, используя который СУПЕРВИЗОР обеспечивает выполнение следующих функций:

- слежение за временем дня, которое проблемная программа может запросить в любой момент своего выполнения;
- печать времени начала и окончания выполнения задания;
- установление интервала времени, по истечении которого должно быть прервано выполнение проблемной программы и передано управление по заданному в этой программе адресу;
- установление интервала времени, в течение которого программе не разрешено выполнение.

Следует отметить, что в каждый конкретный промежуток времени СУПЕРВИЗОР может обеспечивать выполнение функции управления интервалом времени только для одной из числа одновременно выполняемых проблемных программ.

Важной функцией СУПЕРВИЗОРА является *обеспечение мультипрограммного режима работы*. Одновременно может выполняться до трех независимых проблемных программ, каждая из которых находится в своем разделе основной памяти. Принятый режим мультипрограммирования характеризуется тем, что все физические внешние устройства (УВВ и ВЗУ), подключенные к машине, распределяются между разделами. Это распределение осуществляется оператором и при желании может быть изменено. Одно и то же физическое внешнее устройство не может одновременно работать с несколькими разделами. Исключение составляют ВЗУ на магнитных дисках и пишущая машинка: машинка и различные участки одного и того же пакета дисков могут использоваться программами разных разделов.

В режиме мультипрограммирования для каждого из разделов может существовать отдельный входной поток заданий. Наиболее эффективное использование ЭВМ достигается в случае, когда во всех разделах выполняются пакетированные задания, так как при этом простой оборудования сводятся к минимуму.

Между программами, выполняющимися в различных разделах, устанавливается приоритет. Самый высокий приоритет имеет программа первого раздела переднего плана, самый низкий — программа в фоновом разделе (фоновая программа).

Операционная система обеспечивает защиту памяти по разделам. Для каждого раздела основной памяти и области управляющей программой устанавливается свой ключ защиты памяти. В ходе вычислительного процесса СУПЕРВИЗОР осуществляет контроль за тем, чтобы каждая программа обращалась к памяти только своего раздела. Если это правило нарушается, вырабатывается сигнал прерывания ЗАЩИТА ПАМЯТИ, который обрабатывается СУПЕРВИЗОРОМ так же, как программный сбой. При работе в мультипрограммном режиме наличие защиты памяти обязательно.

Вся информация, которую получает СУПЕРВИЗОР от других программ и от оператора, размещается в специальных таблицах. Это таблицы логических и физических устройств, таблица состояний программ, таблица состояний системных файлов на дисках, таблица очередей к внешним устройствам и др. СУПЕРВИЗОР выполняет свои функции, используя содержимое этих таблиц. Одна из них, называемая областью связи, предназначена для совместного использования СУПЕРВИЗОРОМ и проблемными программами.

14.3. УПРАВЛЯЮЩАЯ ПРОГРАММА ДОС ЕС. ПРОГРАММЫ УПРАВЛЕНИЕ ЗАДАНИЯМИ И ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ

Основная задача программы УПРАВЛЕНИЕ ЗАДАНИЯМИ — управление потоком пакетированных заданий, т. е. прием входного потока заданий и подготовка системы к их выполнению.

Основные функции программы УПРАВЛЕНИЕ ЗАДАНИЯМИ заключаются в следующем:

- подготовка программы к выполнению;
- назначение логическим устройствам физических устройств;
- установка режима работы системы;
- редактирование и заполнение информации о метках файлов;
- запись информации в область связи;
- подготовка к запуску программ с контрольной точки.

Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ включается в работу каждый раз, когда необходимо подготовить систему к выполнению очередного задания или очередного шага задания. Она вызывается СУПЕРВИЗОРОМ из библиотеки абсолютных модулей (из резиденции системы) и записывается в тот раздел области проблемных программ основной памяти, в котором должно выполняться подготавливаемое задание или шаг задания. Вызов программы УПРАВЛЕНИЕ ЗАДАНИЯМИ в основную память производится после нормального завершения задания (шага задания), после обработки ненормального завершения программы, по требованию оператора начать пакетную обработку в разделе переднего плана основной памяти, после процедуры первоначальной загрузки.

Выполнение программы УПРАВЛЕНИЕ ЗАДАНИЯМИ заканчивается передачей управления СУПЕРВИЗОРУ, который вызывает затребованную программу из библиотеки абсолютных модулей в тот раздел памяти, в котором до этого находилась программа УПРАВЛЕНИЕ ЗАДАНИЯМИ. Следовательно, программа УПРАВЛЕНИЕ ЗАДАНИЯМИ из основной памяти удаляется и весь раздел памяти предоставляется в распоряжение проблемной программы.

Для выполнения своих функций программа УПРАВЛЕНИЕ ЗАДАНИЯМИ использует информацию, выдаваемую программистом в виде управляющих операторов, информацию, поступающую от оператора в виде директив УПРАВЛЕНИЯ ЗАДАНИЯМИ, а также информацию, содержащуюся в таблицах СУПЕРВИЗОРА. Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ может корректировать содержимое этих таблиц и записывать в них новую информацию.

Рассмотрим подробнее функции программы УПРАВЛЕНИЕ ЗАДАНИЯМИ.

Подготовка программы к выполнению. Существо этой процедуры заключается в следующем. Просматривается оглавление библиотеки абсолютных модулей и из него выписывается в специальную таблицу информация о расположении в библиотеке всех фаз затребованной в задании программы. В этой таблице фиксируется таким образом оглавление фаз программы, которое помещается на диск резидентного пакета. В дальнейшем таблица с оглавлением фаз затребованной программы будет использоваться СУПЕРВИЗОРОМ для ускорения поиска очередной фазы во время

выполнения проблемной программы. После построения оглавления фаз программа УПРАВЛЕНИЕ ЗАДАНИЯМИ очищает раздел основной памяти, в котором будет выполняться затребованная программа, и передает управление СУПЕРВИЗОРУ. СУПЕРВИЗОР вызывает в освобожденный от информации раздел памяти первую фазу программы и запускает ее. Вызов в основную память всех остальных фаз осуществляется также СУПЕРВИЗОРОМ по требованию выполняемой программы.

Назначение логическим устройствам физических устройств. ДОС не только обеспечивает независимость программ от конкретных физических адресов внешних устройств, но и поддержание и использование установленного соответствия между логическими и физическими устройствами в процессе работы.

Различаются три типа назначений логическим устройствам физических устройств: стандартное, постоянное и временное.

Стандартным называется назначение, которое производится во время генерации системы. Оно осуществляется только для фонового раздела основной памяти и характеризуется тем, что принятое соответствие между логическими и физическим устройствами вступает в силу с момента выполнения процедуры первоначальной загрузки и сохраняется до тех пор, пока не будет заменено другим соответствием, установленным с помощью постоянного назначения. Стандартное назначение восстанавливается во время очередной процедуры первоначальной загрузки; это его существенная особенность.

Постоянным называется назначение, устанавливаемое оператором с помощью директив назначения, если в директиве отсутствует специальное указание о том, что назначение должно быть временным. Постоянное назначение отменяет стандартное и остается в силе, сохраняясь от задания к заданию, до тех пор, пока не будет сделано другое постоянное назначение или пока не будет выполнена процедура первоначальной загрузки. Постоянное назначение может быть заменено временным только на период выполнения одного задания, после чего оно восстанавливается.

Временным называется назначение, которое устанавливается оператором с помощью директив назначения только на время выполнения одного задания, отменяя на это время соответствие между логическими и физическими устройствами, установленное до этого постоянным или стандартным назначением. После окончания выполнения задания все временные назначения для логических устройств автоматически отменяются и в силу вступают стандартные или постоянные назначения. Временное назначение может устанавливаться также программистом с помощью управляющих операторов УПРАВЛЕНИЯ ЗАДАНИЯМИ. Временные назначения дают возможность оператору оперативно вмешиваться в вычислительный процесс в тех случаях, когда по какой-либо причине необходимо одно физическое внешнее устройство заменить другим.

Для разделов переднего плана стандартные назначения не производятся. Для программ, размещаемых в этих разделах, устанавливаются, как правило, временные назначения. Эти назначения становятся постоянными и действие их аналогично действию постоянных назначений для программ фонового раздела, если оператор указывает операционной системе, что в данном разделе переднего плана назначения, установленные для конкретного задания, должны быть сохранены и после выполнения этого задания.

Все назначения логических устройств реализуются программой УПРАВЛЕНИЕ ЗАДАНИЯМИ. На основании информации, содержащейся в директивах и управляющих операторах, она производит соответствующие назначения, отменяет временные назначения по операторам отмены назначения и по окончании задания, отменяет любые назначения по директивам отмены назначения и восстанавливает постоянные или стандартные назначения после отмены временного. Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ контролирует также правильность указанных программистом и оператором назначений, т. е. следит, например, за тем, чтобы одно и то же физическое внешнее устройство не было назначено одновременно разным разделам основной памяти, чтобы для системного логического устройства было назначено физическое устройство из числа допустимых типов.

Установка режима работы системы. ДОС может предоставить программисту целый ряд услуг, совокупность которых и создает определенный режим использования операционной системы, или режим работы системы. Он устанавливается как режим работы только для системных обслуживающих программ. Так как эти программы выполняются только в фоновом разделе, то о режиме работы системы имеет смысл говорить только для программ фонового раздела.

Режим работы системы задается либо при генерации ДОС (это стандартный режим), либо устанавливается по требованию программиста для его задания. Укажем некоторые из режимов работы системы, устанавливаемых по директивам и управляющим операторам УПРАВЛЕНИЯ ЗАДАНИЯМИ;

— режим печати управляющих операторов и директив, вводимых с перфокарт программами УПРАВЛЕНИЕ ЗАДАНИЯМИ или ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ, на пультовой пишущей машинке;

— режим распечатки областей основной памяти, используемых управляющей программой и разделом, занимаемым проблемной программой, после ненормального завершения этой проблемной программы;

— режим «Редактировать и выполнить». В этом режиме проблемная программа может быть отредактирована и сразу же выполнена без промежуточной каталогизации в библиотеку абсолютных модулей;

— режим «Редактировать и каталогизировать». В этом режиме программа редактируется, затем каталогизируется в библиотеку абсолютных модулей, после чего может быть выполнена в любое время.

Стандартный режим работы системы поддерживается программой УПРАВЛЕНИЕ ЗАДАНИЯМИ. Установленный программистом режим действует в пределах одного задания. По окончании выполнения задания программа УПРАВЛЕНИЕ ЗАДАНИЯМИ восстанавливает стандартный режим.

Редактирование и запоминание информации о метках файлов. Метки файлов обрабатываются во время выполнения программы. Информация о них, необходимая для сопоставления с ней меток томов и файлов с целью их проверки или для формирования меток, поставляется программой УПРАВЛЕНИЕ ЗАДАНИЯМИ во время подготовки программы к выполнению. В свою очередь, программа УПРАВЛЕНИЕ ЗАДАНИЯМИ считывает эту информацию из управляющих операторов, вводимых программистом, редактирует ее, приводя к виду, близкому к формату меток, и помещает в резидентный пакет дисков для последующей обработки (для дальнейшего использования программами системы управления вводом-выводом).

В резиденции системы для хранения информации о метках файлов выделяется специальный участок, называемый цилиндром меток. Цилиндр меток разделяется на четыре области. Первые три области отведены для хранения информации о метках, поступающей из соответствующих трех разделов основной памяти. В четвертой области хранится информация о метках файлов, с которыми могут работать программы любого из разделов. Такие файлы называются *системными файлами*, а область резиденции системы, где они хранятся, — *областью меток системных файлов* или *областью меток системы*. Информация в эту область обычно записывается во время генерации ДОС и сохраняется постоянно во время работы, что избавляет программиста от необходимости готовить для своего задания управляющие операторы, описывающие метки системных файлов.

Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ обеспечивает запись информации о метках, вводимую с помощью управляющих операторов, всегда в одну из областей цилиндра меток: в область меток системных файлов или в область меток, соответствующую тому разделу основной памяти, в котором записано задание (номер области указывается программистом).

Информация о метках, вводимая программистом для его задания, может быть записана в область меток системных файлов, и тогда она может использоваться всеми программами, выполняющимися в любом из разделов, и сохраняется до тех пор, пока программист не потребует записи в эту область новой информации о метках.

Если информация о метках предназначена для использования в программах, выполняющихся в одном и том же разделе, она записывается в область меток этого раздела. В зависимости от того, на каких дорожках диска этой области будет записана информация о метках, она может храниться либо от задания к заданию, пока в очередном задании не поступит требование записать новую информацию для этого раздела, либо только на время выполнения одного задания.

Запись информации в область связи. Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ воспринимает из управляющих операторов и заносит в область связи основной памяти следующую информацию:

- имя задания;
- дату выполнения задания;
- программные переключатели.

Подготовка к запуску программы с контрольной точки. Эта подготовка сводится к тому, что программа УПРАВЛЕНИЕ ЗАДАНИЯМИ вызывает в основную память подпрограмму СУПЕРВИЗОРА, осуществляющую подвод магнитных лент, восстановление области связи и собственно запуск программы с контрольной точки. Повторный запуск должен производиться в том же разделе, в котором программа выполнялась в момент создания контрольной точки.

Программа ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ обеспечивает подготовку операционной системы к выполнению одиночной программы. Она выполняет функции, аналогичные функциям, которые выполняет программа УПРАВЛЕНИЕ ЗАДАНИЯМИ для пакетированных заданий.

Каждый раз, когда в разделе переднего плана основной памяти должна быть выполнена одиночная программа, оператор с пультовой пишущей машинки запрашивает выполнение ИНИЦИАТОРА ОДИНОЧНЫХ ПРОГРАММ. Запрос, осуществляемый с помощью директивы программы ВНИМАНИЕ, воспринимается СУПЕРВИЗОРОМ, который вызывает из библиотеки абсолютных модулей программу ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ и ей передает управление. После этого оператор должен вводить директивы, необходимые для запуска одиночной программы.

Функции ИНИЦИАТОРА ОДИНОЧНЫХ ПРОГРАММ, по сравнению с функциями программы УПРАВЛЕНИЕ ЗАДАНИЯМИ, ограничены. Основные ограничения и особенности заключаются в следующем:

- повторный запуск одиночной программы с контрольной точки исключается;
- уменьшено количество типов физических устройств, которые могут быть назначены системным логическим устройствам;
- после завершения одиночной программы раздел, в котором она выполнялась, освобождается. Выполнение новой программы в этом разделе может быть начато только по указанию оператора;

— программы, выполняющиеся в одиночном режиме, не могут использовать информацию, находящуюся в области связи основной памяти.

В качестве одиночных программ не могут выполняться системные обслуживающие программы, трансляторы, а также проблемные программы, требующие доступа к области связи.

14.4 УПРАВЛЯЮЩАЯ ПРОГРАММА ДОС ЕС. СИСТЕМА УПРАВЛЕНИЯ ВВОДОМ-ВЫВОДОМ

В ДОС ЕС система управления вводом-выводом выполняет функции управления данными, предоставляя программисту стандартные процедуры для управления данными и обеспечения доступа к ним.

Имеется два уровня управления данными: физический и логический. Каждому из этих уровней соответствует своя совокупность программ, называемых соответственно физической системой управления вводом-выводом (PIOCS) и логической системой управления вводом-выводом (LIOCS).

Физический уровень управления данными требует точного знания физических внешних устройств для организации работы с ними. При использовании логического уровня программисту эти знания не требуются, он заботится лишь о логической структуре своих данных.

Физическая система управления вводом-выводом. Система PIOCS, используемая всеми программами, выполняющимися в рамках ДОС, обеспечивает контроль реализации канальных программ. Она включает средства для планирования и установления очередности операций ввода-вывода, контроля выполнения операций, относящихся к устройствам ввода-вывода, обработки прерываний ввода-вывода. Все эти функции выполняются подпрограммами, являющимися составной частью СУПЕРВИЗОРА.

При необходимости выполнения канальной программы программист обращается к СУПЕРВИЗОРУ и сообщает ему информацию, представляющую собой *блок управления данными*. В этом блоке указывается, например, логическое устройство, для которого должна быть реализована канальная программа, адрес канальной программы, какая должна быть реакция СУПЕРВИЗОРА в случае возникновения ошибок.

Использование системы PIOCS дает возможность программисту применять в своей программе аппарат логических устройств.

Логическая система управления вводом-выводом. Эта система предназначена для обработки файлов данных. Обработка файлов включает такие операции: считывание записей файла, собственно их обработку и запоминание обработанной записи в ВЗУ. В отличие от физической системы управления вводом-выводом, для системы LIOCS существенным является логическое содержание данных, их формат и организация данных в файлы.

В состав файла могут входить записи с форматом фиксированной длины, переменной (но известной) длины и неопределенной (заранее не известной) длины. В перфокарточных устройствах допускаются записи только с форматом фиксированной длины. Файлы на магнитных лентах и дисках могут иметь метки, которые используются программами системы LIOCS для опознавания файлов, для организации работы с ними, а также для организации защиты файлов.

Методы доступа к данным. Различные методы доступа к данным отличаются используемыми способами организации данных и их обработки. В ДЭС применяются три метода доступа: методы последовательного, прямого и индексно-последовательного доступа.

Для *метода последовательного доступа* характерной является последовательная организация данных, при которой логические записи на внешнем накопителе располагаются в том порядке, в котором они затем будут обрабатываться. В этом случае для доступа к требуемой логической записи файла необходимо перебрать все предшествующие записи этого файла, поэтому такой метод применяется, если требуется обрабатывать все или большинство записей файла. Файлы с последовательной организацией записей могут храниться на перфокартах, магнитных лентах, дисках и т. д.

Метод прямого доступа позволяет сразу выбрать любую логическую запись файла по ее адресу. Этот метод применяется при организации работы с файлами, расположенными на магнитных дисках. При этом не накладывается никаких ограничений на структуру файла и его расположение на дисках, эти характеристики определяются самим программистом. При использовании метода прямого доступа программист указывает действительный адрес необходимой записи в файле, следовательно, он несет ответственность за правильность программирования, требуемого для определения местонахождения записей. В этом недостаток этого метода, обладающего, тем не менее, большой гибкостью.

Метод индексно-последовательного доступа, введенный также для организации работы с файлами на дисках, предполагает индексно-последовательную организацию файла. В отличие от обычного последовательного файла, в котором записи располагаются в порядке их последующей обработки, каждая запись индексно-последовательного файла снабжается ключом (индексом) и записи в файле располагаются в последовательности ключей. Индексы используются для быстрого доступа к отдельным записям, а также для последовательной обработки записей файла. При создании файла с индексно-последовательной организацией на диске выделяется область, в которую в случае надобности можно добавлять новые записи. Это позволяет избежать перезаписи всего файла, которая обычно неизбежна в подобном случае при использовании файла с последовательной организацией записей. Таким образом, метод индексно-последовательного доступа сочетает в себе преимуще-

щества первых двух методов и облегчает функции программиста по сравнению с методом прямого доступа.

Каждому из указанных методов доступа к данным соответствуют свои макрокоманды системы LIOCS. Программист, выбирая метод доступа для обработки своего файла, использует макрокоманды двух видов: декларативные и императивные. Первые служат для создания таблиц характеристик файлов (для всех подлежащих обработке файлов такие таблицы создаются обязательно, они используются программами системы LIOCS). Вторые предназначены для включения в проблемную программу необходимых подпрограмм системы LIOCS, а также для указания действий, которые должны выполняться в ходе обработки файла.

При трансляции или редактировании проблемной программы с помощью императивных макрокоманд в нее включаются все подпрограммы системы LIOCS, затребованные этими макрокомандами. Обычно эти подпрограммы транслируются отдельно от проблемных программ и помещаются в библиотеку объектных модулей. Включение их в проблемную программу по императивным макрокомандам осуществляется РЕДАКТОРОМ.

Совокупность подпрограмм системы LIOCS, затребованных программистом для обработки своего файла, образует *обобщенный модуль* этой системы. Для методов прямого и индексно-последовательного доступа, а также для каждого из типов внешних устройств, к которым может быть применен метод последовательного доступа, существуют отдельные обобщенные модули. Подпрограммы этих модулей учитывают особенности и возможности методов доступа и специфику используемых внешних устройств.

В ДОС имеется также обобщенный модуль, обеспечивающий для некоторых системных логических устройств обработку файлов с последовательной организацией записей, независимую от типов физических внешних устройств. Следовательно, не делая никаких изменений в проблемной программе, для файлов, связанных с этими системными логическими устройствами, можно использовать любое из физических устройств, которые могут быть назначены системным логическим устройством.

14.5 СИСТЕМНЫЕ ОБСЛУЖИВАЮЩИЕ ПРОГРАММЫ ДОС ЕС

РЕДАКТОР. Процесс подготовки программы к выполнению в ДОС ЕС делится на три основных этапа: программирование, трансляция и редактирование. Результатом выполнения первого этапа, который представляет собой запись алгоритма решаемой задачи на одном или нескольких исходных языках, является один или несколько *исходных модулей*. Во время второго этапа исходные модули обрабатываются соответствующими трансляторами. В результате трансляции получаются *объектные модули*. Наконец, во время третьего этапа объектные модули обрабатываются

программой РЕДАКТОР. В результате редактирования получается одна или несколько *программных фаз*.

Таким образом, основной функцией РЕДАКТОРА является формирование готовых к выполнению программ из объектных модулей, создаваемых трансляторами. Отредактированные программы помещаются в библиотеку абсолютных модулей.

Для отредактированной программы РЕДАКТОРОМ может быть затребована процедура каталогизации, и тогда эта программа помещается постоянно в библиотеку абсолютных модулей, благодаря чему ее можно затребовать оттуда в любое время. Программа, отредактированная, но не прошедшая процедуры каталогизации, хранится в библиотеке абсолютных модулей временно: либо до окончания выполнения текущего задания, либо до момента, когда в этом задании производится новое обращение к РЕДАКТОРУ.

На этапе редактирования все связи между различными автономными частями программы (поскольку на предыдущих этапах была необходимость составлять программу по частям), представленные ранее символически на исходных алгоритмических языках (они сохраняются в объектных модулях как внешние ссылки), получают свои конкретные значения.

В результате редактирования из объектных модулей формируются либо однофазные, либо многофазовые программы.

В простейшем случае отредактированная проблемная программа состоит из одной фазы, и при ее выполнении она вызывается в основную оперативную память целиком. Большие по объему программы имеют многофазовую структуру, причем могут быть две разновидности такой структуры: структура с перекрытием с корневой фазой и структура с перекрытием без корневой фазы.

Многофазовая структура программы с перекрытием с корневой фазой предполагает наличие в составе программы так называемой корневой фазы, которая во время выполнения программы постоянно находится в основной памяти. Корневая фаза вызывает в основную память другие фазы программы, последовательно сменяющие друг друга. Тем самым достигается существенная экономия емкости основной памяти, необходимой для выполнения программы. Корневая фаза обычно содержит общие для остальных фаз подпрограммы и константы.

Многофазовая структура программы с перекрытием без корневой фазы предполагает последовательную смену фаз в основной памяти, причем каждая предыдущая фаза вызывает последующую и перекрывается ею.

Выбор структуры программы осуществляется программистом и планируется еще на исходном языке. Главным критерием при выборе структуры является емкость основной памяти, необходимая для размещения программы. На этапе составления программы на исходном языке программист сам устанавливает связи между автономными ее частями (между исходными модулями) и способы

передачи информации между ними, используя для этого средства, предоставляемые исходным языком. На этапе редактирования программист с помощью управляющих операторов должен указать РЕДАКТОРУ информацию, необходимую для формирования программы выбранной структуры.

Каждый объектный модуль может состоять из нескольких более мелких частей — программных секций, обрабатываемых РЕДАКТОРОМ независимо друг от друга. Программные секции, принадлежащие одному и тому же объектному модулю, могут размещаться в различных программных фазах. Разделение объектного модуля на программные секции и выдача указаний РЕДАКТОРУ, в какую программную фазу должна быть включена та или иная программная секция, осуществляется программистом.

По требованию программиста в редактируемую программу могут быть включены объектные модули, находящиеся как в личной, так и в системной библиотеках объектных модулей. Такое включение может осуществляться автоматически, поскольку РЕДАКТОР располагает соответствующим аппаратом.

Конкретные действия, которые выполняет РЕДАКТОР во время процедуры редактирования, зависят от исходных данных и от выбранной структуры программы. Выделяются при этом два этапа: формирование текста фаз и формирование записей оглавления.

Текст фазы программы формируется из текста объектных модулей с помощью управляющих операторов РЕДАКТОРА. Основные действия РЕДАКТОРА на этом этапе следующие:

- компоновка текста фазы из отдельных программных секций объектного модуля;

- автоматическое включение (если оно затребовано) в фазу недостающих объектных модулей из библиотеки объектных модулей;

- замена некоторых частей текста объектных модулей новым текстом, если потребителем заданы специальные карты изменений;

- перемещение скомпонованного текста фазы по адресу загрузки, заданному потребителем;

- присвоение конкретных значений внешним связям между объектными модулями при объединении их в фазу;

- выбор точки входа в каждую фазу (точкой входа называется адрес той команды фазы, с которой должно начинаться выполнение фазы; вызванной в основную память);

- построение общих областей памяти для всей редактируемой программы;

- определение раздела основной памяти, для которого редактируется программа.

Одновременно с формированием текста фазы формируется запись об этой фазе для оглавления. Эта запись содержит данные, необходимые операционной системе при поиске фазы в библиотеке

и вызове ее в основную память для выполнения. Запись, в частности, содержит имя фазы, адрес ее загрузки в основную память, точку входа в фазу, координаты текста фазы в библиотеке на диске. Совокупность таких записей для всех фаз называется *оглавлением фаз программы*. Оглавление размещается в системной рабочей области, представляющей собой специальную часть резидентного дискового пакета.

Завершающие действия РЕДАКТОРА направлены на размещение отредактированной программы в библиотеке абсолютных модулей — временно или постоянно, если предполагается многократное использование программы. В последнем случае программа подвергается каталогизации. Процедура каталогизации заключается в перенесении оглавления фаз программы из системной рабочей области в оглавление библиотеки абсолютных модулей.

БИБЛИОТЕКАРЬ. БИБЛИОТЕКАРЕМ называют совокупность программ, предназначенных для создания, обслуживания и корректировки библиотек ДОС, отображения их состояния и копирования.

В операционной системе ДОС имеются три вида библиотек:

- библиотека абсолютных модулей;
- библиотека объектных модулей;
- библиотека исходных модулей.

Библиотека абсолютных модулей может быть только системной, остальные библиотеки могут быть как системными, так и личными. Сама система ДОС представляет собой совокупность этих трех библиотек, причем для ее функционирования обязательно должна присутствовать библиотека абсолютных модулей, две другие библиотеки могут и отсутствовать. Системные библиотеки входят в состав резидентного файла. Личные библиотеки в состав резидентного файла не входят, однако располагаются также на пакете дисков.

Каждая из названных библиотек имеет свое оглавление, в котором содержатся записи с информацией об элементах библиотек. Эти записи используются для определения расположения элементов в библиотеке на магнитных дисках.

В состав библиотеки абсолютных модулей включаются системные управляющие и обслуживающие программы, трансляторы, программы сортировки, а также готовые к выполнению программы потребителей. В эту библиотеку программы помещаются только РЕДАКТОРОМ, все они (кроме самоперемещающихся программ) имеют абсолютный формат, т. е. отредактированы для фиксированного места в основной памяти. Наименьшей единицей в библиотеке абсолютных модулей, с которой может оперировать БИБЛИОТЕКАРЬ, является фаза.

Элементом библиотеки объектных модулей является объектный модуль, представляющий собой результат одной трансляции любым из трансляторов ДОС. В библиотеке хранятся часто ис-

пользуемые модули. Помещение их в библиотеку осуществляется только БИБЛИОТЕКАРЕМ. В оглавлении библиотеки для каждого объектного модуля содержится запись со следующей информацией: имя модуля, адрес модуля на диске, количество физических блоков на диске, содержащих модуль, версия и модификация (служат для отражения состояния разработки компонентов ДОС). Запись оглавления о конкретном модуле необходима для определения его расположения в библиотеке.

Библиотека исходных модулей предназначена для накопления исходных модулей, написанных на языках программирования ДОС, и макроопределений, реализующих макрокоманды АССЕМБЛЕРА. Наименьшей единицей в библиотеке является исходный модуль, или книга, представляющая собой последовательность операторов на любом из исходных языков. Более крупная единица — подбиблиотека, в которую входит набор книг на исходном языке, предназначенных для использования определенным транслятором. Все книги библиотеки распределены по соответствующим подбиблиотекам. Такое распределение позволяет присваивать книгам, написанным на различных языках, совпадающие имена.

Для библиотеки исходных модулей имеется свое оглавление, составленное из записей для каждой книги. Запись о книге содержит следующие данные: признак (имя) подбиблиотеки, имя книги, адрес книги на диске, количество физических блоков, на которых размещается книга, версия и модификация. Запись оглавления о конкретной книге необходима для определения ее местоположения в библиотеке и выборки из библиотеки. Книги в библиотеку исходных модулей могут помещаться только БИБЛИОТЕКАРЕМ.

Личные библиотеки (объектных и исходных модулей) имеют структуру, аналогичную структуре соответствующих системных библиотек, но оформляются они как самостоятельные файлы. Потребитель может иметь любое необходимое количество личных библиотек, каждая из которых служит для конкретных целей. При формировании программных фаз РЕДАКТОР может использовать модули, расположенные как в личных, так и в системных библиотеках.

Функции, выполняемые программами БИБЛИОТЕКАРЯ, разделяются на три основных класса:

- функций корректировки;
- сервисные функции (печать и перфорация);
- функции создания и копирования библиотек.

Эти функции распространяются на все типы библиотек ДОС и реализуются соответствующими программами БИБЛИОТЕКАРЯ. Указание о том, какую именно функцию необходимо выполнить и применительно к какой библиотеке, дает программист с помощью управляющих операторов БИБЛИОТЕКАРЯ.

Функции корректировки включают такие процедуры: каталогизацию, удаление, переименование, обновление, сжатие, установление параметра автоматического сжатия, распределение.

Процедура каталогизации заключается в добавлении фазы в библиотеку абсолютных модулей (выполняется только непосредственно после редактирования, если для РЕДАКТОРА указан режим каталогизации), модуля в библиотеку объектных модулей и книги в библиотеку исходных модулей.

Процедура удаления преследует цель вычеркнуть из оглавления некоторой библиотеки запись, соответствующую удаляемому элементу (фазе, модулю, книге). После выполнения этой процедуры элемент для операционной системы перестает существовать, однако физически занимаемое им место в библиотеке не освобождается и не может быть использовано для других элементов. Это место освобождается и оказывается доступным для использования только после выполнения над соответствующей библиотекой процедуры сжатия, назначение которой и состоит в том, чтобы исключить логически удаленные элементы из библиотеки и вычеркнутые записи из оглавления.

Процедура переименования используется для присвоения новых имен элементам той или иной библиотеки.

Процедура обновления действительна только для библиотеки исходных модулей. Она позволяет изменять содержимое отдельных книг путем добавления, удаления или вставки некоторых операторов книг.

Процедура установления параметра автоматического сжатия дает возможность БИБЛИОТЕКАРЮ следить за резервом доступных для записи физических блоков в библиотеках и производить автоматическое сжатие в случае, когда этот резерв становится меньше указанного параметра. Установление параметра заключается в том, что он запоминается в оглавлении соответствующей библиотеки.

Процедура распределения предназначена для изменения размеров областей на магнитных дисках, отведенных под системные библиотеки и их оглавления. Используя эту процедуру, можно увеличить, уменьшить, исключить или добавить системные библиотеки и их оглавления резидентного файла. Размеры личных библиотек определяются во время их создания и не изменяются в течение всего времени работы с ними.

Сервисные функции делаются на три группы: печать, перфорация, печать и перфорация. С помощью этих функций можно:

— вывести на печать содержимое целых библиотек или отдельных их элементов;

— вывести на печать содержимое оглавлений отдельных или всех библиотек, а также таблицы состояний системных и личных библиотек (в этих таблицах содержатся сведения о системных и личных библиотеках: информация о распределении пакета дисков между библиотеками и их оглавлениями, о заполненности библиотек и оглавлений);

— отперфорировать содержимое целых библиотек или отдельных их элементов;

— одновременно вывести на печать и отперфорировать содержимое целых библиотек или отдельных их элементов.

Функции создания и копирования используются для получения копии (полной или частичной) резидентного файла, а также для создания или копирования личной библиотеки.

АВТОТЕСТ. Как уже отмечалось, АВТОТЕСТ предназначен для оказания помощи программисту при отладке программ. Отлаживаемая программа поступает в распоряжение АВТОТЕСТА в виде объектных модулей, создаваемых с языков АССЕМБЛЕР, ФОРТРАН, РПГ, ПЛ/1. АВТОТЕСТ, подобно РЕДАКТОРУ, обрабатывает эти объектные модули и формирует из них программные фазы, в которые включается информация, соответствующая отладочным запросам программиста. АВТОТЕСТ позволяет: без перетрансляции заменить, добавить или удалить некоторые команды отлаживаемой программы, в затребованных точках выполнения программы напечатать интересующую программиста информацию (например, содержимое указанных областей основной памяти, содержимое общих регистров), получить список фаз с указанием адресов их загрузки в основную память.

Запросы программиста по отладке своей программы производятся с помощью управляющих операторов АВТОТЕСТА. Программа АВТОТЕСТ всегда выполняется в фоновом разделе основной памяти.

УТИЛИТЫ. УТИЛИТАМИ называются программы, обеспечивающие выполнение часто повторяющихся вспомогательных операций. Эти программы обычно хранятся в библиотеке абсолютных модулей и могут выполняться в режиме пакетной обработки в любом из разделов основной памяти. Однако они могут быть представлены и в виде объектных модулей, и тогда их редактирование осуществляется вместе с проблемными программами.

Программы УТИЛИТЫ разбиваются на две группы: УТИЛИТЫ типа «Файл—файл» и специальные УТИЛИТЫ. Первые из них служат для перемещения файлов с одного носителя данных на другой: с дисков, лент, карт на диски, ленты, карты и печать в любом сочетании. Перезапись может осуществляться: без изменения формата записей и размера блока (копирование), с изменением только размера блока (перезапись), с изменением внутренней структуры каждой записи (перекомпоновка), с изменением того и другого (перезапись с перекомпоновкой). Специальные УТИЛИТЫ представляют собой набор самостоятельных программ, позволяющих выполнять такие функции, как очистка одной или нескольких областей диска, назначение запасных дорожек диска вместо дефектных, установление идентичности двух ленточных файлов путем их сравнения и др.

УТИЛИТЫ не могут выполняться как одиночные программы. В то же время такие операции, как перемещение файла с одного носителя информации на другой удобно выполнять в режиме оди-

ночных программ, рассматривая их как независимую самостоятельную работу. Поэтому в ДОС имеется набор макрокоманд, с помощью которых программист получает нужный ему УТИЛИТ, выполняемый как одиночная программа в любом разделе основной памяти.

Программы сортировки. Основное назначение программ сортировки состоит в сортировке файлов и их слиянии.

Процедура сортировки применяется к файлам, состоящим из неупорядоченных записей. При ее выполнении записи в файле располагаются в определенной последовательности (убывающей или возрастающей) по некоторому признаку.

Процедура слияния представляет собой процесс объединения нескольких файлов с упорядоченными записями в один файл упорядоченных записей, называемый выводным файлом. Выходная последовательность записей, полученная в выводном файле, должна быть такой же, как и входная последовательность для файла, подлежащего объединению с другими файлами (например, слияние файлов с возрастающей по некоторому признаку последовательностью записей).

В ходе выполнения заданных процедур программы сортировки создают контрольные точки, позволяющие прервать процесс сортировки или слияния и возобновить его с фиксированной промежуточной точки.

В ДОС имеются программы сортировки на магнитных лентах и программа сортировки на дисках и магнитных лентах. Каждая из этих программ может выполняться как самостоятельное задание в режиме пакетной обработки в любом разделе основной памяти, а программа сортировки на дисках и магнитных лентах может включаться в проблемные программы в качестве подпрограммы.

Программа сортировки на магнитных лентах, включающая несколько фаз и постоянно хранящаяся в библиотеке абсолютных модулей, производит упорядочение файлов, расположенных на магнитных лентах, и выводной файл помещает также на магнитные ленты.

Для программы сортировки на магнитных дисках и лентах исходные (иначе: вводные) файлы могут располагаться на дисках и лентах (для конкретного выполнения либо только на дисках, либо только на лентах). Выводной файл помещается на диски или ленты. Эта программа представляет собой набор объектных модулей, из которых путем редактирования могут быть получены программы сортировки с различными возможностями. Если полученный конкретный вариант программы сортировки каталогизирован в библиотеку абсолютных модулей, то в дальнейшем он используется без предварительного редактирования.

ОЛТЕП. Программа ОЛТЕП, предназначенная для контроля функционирования устройств ввода-вывода, может выполняться в мультипрограммном режиме и во время выполнения размещается

только в фоновом разделе основной памяти. Для различных типов устройств ввода-вывода имеются свои тестовые программы, которые настраиваются, запускаются и выполняются под управлением программы ОЛТЕП. Тестовая проверка правильности функционирования устройств ввода-вывода позволяет выявить ошибки в их работе и локализовать источники ошибок.

14.6. ТРАНСЛЯТОРЫ. ГЕНЕРАЦИЯ ДОС ЕС

В состав ДОС ЕС включены трансляторы с языков АССЕМБЛЕР, РПГ, ФОРТРАН, ПЛ/1.

Транслятор АССЕМБЛЕРА. Язык АССЕМБЛЕРА предназначен для автоматизации процесса программирования на уровне машинно-ориентированного языка. Он может применяться для решения задач любого типа, так как позволяет непосредственно использовать систему команд ЕС ЭВМ.

При использовании языка АССЕМБЛЕР потребитель получает:

- символическую адресацию элементов программы;
- мнемонические коды операций, символически определяющие функции команд;
- разнообразные способы представления данных;
- использование данных в операндах команд;
- неявное представление в командах длин операндов;
- возможность разбиения исходного модуля на программные секции;
- средства обращения к элементам отдельно транслируемых частей программы;
- возможность включать и транслировать ранее написанные части исходной программы, хранящиеся в библиотеке исходных модулей;
- возможность управлять форматом ввода и вывода исходной программы;
- макросредства, позволяющие изменять порядок трансляции и содержание исходных операторов.

Транслятор АССЕМБЛЕРА производит перевод исходной программы в объектный модуль и выполняется только в фоновом разделе. Рабочие файлы транслятора располагаются на магнитных дисках или лентах.

Транслятор РПГ. Язык РПГ является проблемно-ориентированным языком. Он предназначен для составления программ по задачам, результаты решения которых представляют собой разнообразные отчеты, напечатанные по требуемой форме.

Язык РПГ позволяет:

- описывать типы и структуру вводных и выводных записей, организацию и способ обработки файлов;
- выполнять арифметические и другие операции обработки над данными записей;

- редактировать и печатать каждую вводимую запись и результаты обработки;
- контролировать вводимые записи на их упорядоченность по возрастанию или убыванию заданных признаков;
- группировать записи разных файлов, совпадающие по определенным признакам, и выполнять их совместную обработку;
- производить поиск по таблицам;
- создавать и обрабатывать файлы с различной организацией (последовательной, прямой, индексно-последовательной);
- использовать готовые программы, хранящиеся в библиотеке объектных модулей.

Транслятор РПГ переводит исходную программу в объектный модуль. Он может выполняться только в фоновом разделе. Рабочие файлы транслятора РПГ размещаются на дисках и лентах.

ФОРТРАН. В состав ФОРТРАНА, предназначенного для автоматизации программирования научно-технических задач, входят проблемно-ориентированный язык Базисный ФОРТРАН IV, транслятор с этого языка и библиотека подпрограмм. Язык Базисный ФОРТРАН IV по сравнению с другими алгоритмическими языками отличается простотой, большими возможностями по использованию средств ввода-вывода, наличием средств для определения связей между автономно транслируемыми частями программы.

Транслятор Базисного ФОРТРАНА IV переводит исходную программу в объектный модуль. Он включает средства диагностики, которые могут быть использованы программистом для локализации ошибок в исходной программе.

Библиотека подпрограмм ФОРТРАНА является частью библиотеки объектных модулей. Она включает подпрограммы, предназначенные для вычисления различных математических функций и выполнения некоторых служебных действий. Объединение этих подпрограмм с программой, полученной в результате трансляции, производится РЕДАКТОРОМ.

Транслятор Базисного ФОРТРАНА IV выполняется только в фоновом разделе, рабочие файлы транслятора размещаются на магнитных дисках или лентах.

ПЛ/1. В состав ПЛ/1 входят язык ПЛ/1, транслятор, преобразующий исходный модуль в объектный модуль, и библиотека подпрограмм. Язык ПЛ/1 — один из наиболее универсальных и высоко развитых языков программирования. Он в одинаковой степени пригоден для программирования как научно-технических, так и экономических задач. Транслятор работает только в фоновом разделе, а его рабочие файлы размещаются на дисках или лентах. Библиотека подпрограмм является частью системной библиотеки объектных модулей и включает подпрограммы, предназначенные для выполнения различных служебных функций, обеспечения диагностики при отладке программы и реализации

встроенных функций языка. Включение подпрограмм в проблемную программу при ее обработке осуществляется РЕДАКТОРОМ.

Генерация системы. Потребитель получает машину вместе с ДОС, которая хранится на магнитных дисках или лентах и представляет собой совокупность трех библиотек: абсолютных, объектных и исходных модулей.

В составе библиотеки абсолютных модулей имеется СУПЕРВИЗОР, УПРАВЛЕНИЕ ЗАДАНИЯМИ, программа ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА, РЕДАКТОР, БИБЛИОТЕКАРЬ, АССЕМБЛЕР (язык и транслятор) и некоторые УТИЛИТЫ. Потребителю поставляется один из конкретных вариантов СУПЕРВИЗОРА, который хотя и может быть применен независимо от конфигурации конкретной машины ЕС ЭВМ, однако он не учитывает особенностей этой машины и в нем отсутствуют многие необходимые для потребителей свойства. Основное назначение поставляемого СУПЕРВИЗОРА состоит в том, чтобы обеспечить выполнение процедуры генерации системы. Другие программы библиотеки абсолютных модулей отредактированы для выполнения в фоновом разделе под управлением этого конкретного СУПЕРВИЗОРА.

В библиотеке объектных модулей содержатся объектные модули всех программных компонентов ДОС и среди них: программа УПРАВЛЕНИЕ ЗАДАНИЯМИ, РЕДАКТОР, БИБЛИОТЕКАРЬ, АВТОТЕСТ, УТИЛИТЫ, программы сортировки, трансляторы, конкретные модули логической системы управления вводом-выводом.

Библиотека исходных модулей содержит макроопределения для системных макрокоманд АССЕМБЛЕРА, а также некоторые проверочные задачи для различных обслуживающих программ и трансляторов.

Потребитель, используя полученную ДОС и средства генерации системы, может образовать конкретную ДОС, соответствующую составу оборудования его машины и области ее применения. СУПЕРВИЗОР образованной системы учитывает конфигурацию машины и затребованные потребителем свойства, а в системные библиотеки помещаются только нужные потребителю компоненты ДОС.

Процесс генерации сводится к следующему:

— описываются необходимые свойства СУПЕРВИЗОРА, для чего используются макрокоманды генерации СУПЕРВИЗОРА; эти макрокоманды транслируются АССЕМБЛЕРОМ, в результате получается затребованный вариант СУПЕРВИЗОРА;

— полученный СУПЕРВИЗОР редактируется и каталогизируется в библиотеку абсолютных модулей;

— формируются новые системные библиотеки, при этом из них удаляются модули тех программных компонентов системы, которые не предполагается использовать в машине.

14.7. ПОСТАНОВКА ЗАДАЧИ К РЕШЕНИЮ НА МАШИНЕ С ПОМОЩЬЮ ДОС ЕС

Процесс постановки задачи к решению на машине ЕС ЭВМ содержит следующие этапы (рис. 14.2):

- планирование структуры программы и ее кодирование;
- подготовка программы к выполнению (трансляция и редактирование);
- выполнение программы.

На первом этапе, когда планируется структура программы решаемой задачи, кодируется программа на одном из исходных языков программирования, готовятся исходные данные задачи, всю смысловую нагрузку несет программист. Для составления программы выбирается наиболее подходящий исходный язык программирования, соответствующий характеру задачи. Задача может быть разделена на части и для каждой части использован свой язык, так как система обеспечивает независимую трансляцию каждой автономно подготовленной части и объединение этих частей в одну программу затребованной структуры.

Программист может спланировать программу либо простой структуры, когда вся программа для его выполнения вызывается в основную память, либо с перекрытием, когда программа разбивается на сегменты, которые при выполнении вызываются в основную память поочередно.

После деления программы на сегменты (если в этом есть необходимость) производится планирование структуры каждого исходного модуля, заключающееся в разбиении модуля на подпрограммы, которые должны создаваться самим программистом, включении в состав модуля готовых подпрограмм из библиотек, установлении связей между подпрограммами модуля (такие связи называются прямыми).

Последующие этапы постановки задачи к решению выполняются на машине.

На этапе подготовки программы к выполнению производится трансляция и редактирование. В результате трансляции полу-

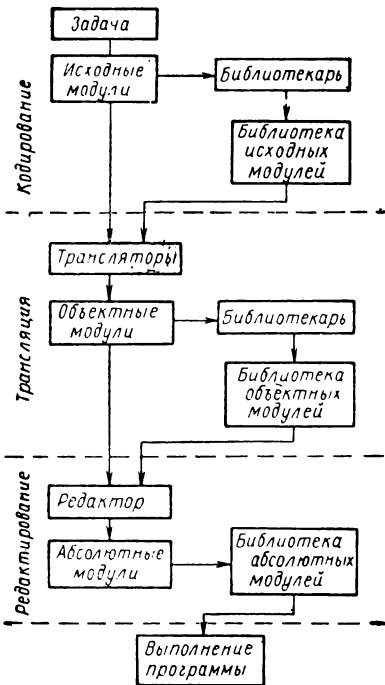


Рис. 14.2. Этапы постановки задачи к решению на ЭВМ с помощью ДОС ЕС

чаются объектные модули, которые затем подвергаются редактированию.

Если объектный модуль некоторой программы не является в какой-то степени универсальным, т. е. он не может быть использован при выполнении других программ или при выполнении той же программы, но в другое время, то этот модуль после его редактирования исчезает из системы. В противном случае объектный модуль отправляется в библиотеку объектных модулей. Во время редактирования программист с помощью управляющих операторов РЕДАКТОРА описывает структуру своей программы, указывая ее деление на фазы и состав каждой фазы.

После редактирования готовая программа либо выполняется немедленно и после выполнения исчезает из системы, либо помещается в библиотеку абсолютных модулей, если она потребуется в течение некоторого времени неоднократно.

Перед тем как отредактированная программа должна выполняться, необходимо подготовить задание на ее выполнение. Две полностью независимые друг от друга программы предпочтительнее оформлять как отдельные задания, а не как шаги одного и того же задания, так как в случае ненормального завершения одного шага задания снимается все задание. Трансляция и редактирование некоторой программы обычно оформляются как шаги одного задания.

При формировании заданий необходимо учитывать, что эффективность функционирования однопроцессорной вычислительной системы в мультипрограммном режиме часто оценивается пропускной способностью, определяемой количеством стандартных (по времени выполнения) заданий, которые выполняются системой в единицу времени. Основная цель организации мультипрограммного режима — увеличить пропускную способность системы за счет более эффективного использования ее ресурсов, в частности за счет совмещения во времени операций ввода-вывода и операций обработки.

Например, в то время, когда программы переднего плана выполняют операции ввода-вывода, фоновая программа может выполнять функции обработки. При работе в мультипрограммном режиме желательно, чтобы часто используемые программы были самоперемешивающимися, т. е. такими, которые могут выполняться в любом месте основной памяти (во время выполнения таких программ осуществляется модификация их адресных констант).

Пропускная способность системы повышается, если в обоих разделах переднего плана основной памяти допускается режим пакетной обработки, когда программа УПРАВЛЕНИЕ ЗАДАНИЯМИ обеспечивает непрерывное выполнение поступающих заданий, не требуя вмешательства оператора.

14.8. ОПЕРАЦИОННАЯ СИСТЕМА ЕС ЭВМ. СОСТАВ И ФУНКЦИИ ОС ЕС

Операционная система (ОС) ЕС ЭВМ предназначена для повышения производительности (пропускной способности) ЭВМ путем управления распределением всех имеющихся ресурсов, включая машинное время центрального процессора, емкость основной памяти и внешних запоминающих устройств, устройства ввода-вывода, любые программы, содержащиеся в различных библиотеках. Кроме того, ОС обеспечивает взаимодействие вычислительной системы и оператора (программиста) в удобных для него формах, а также гибкое и упорядоченное изменение состава самой системы при расширении сферы ее применений.

Операционная система может быть использована на всех совместимых моделях ЭВМ ЕС (ЕС-1020, ЕС-1030, ЕС-1040, ЕС-1050) и, кроме того, на всех машинах, программно совместимых с этими моделями. Для применения ОС в вычислительной системе необходимо иметь хотя бы одно ВЗУ на магнитных дисках и емкость основной оперативной памяти не менее 64К байт. Для эффективного применения ОС емкость основной памяти должна быть не менее 128К байт.

Состав и структура ОС определены с учетом удовлетворения двух главных требований: универсальности и возможности настройки на некоторый конкретный круг решаемых задач.

Универсальность ОС заключается в возможности ее использования во всех областях применений и на всех конфигурациях и моделях ЭВМ ЕС. Универсальность ОС обеспечивается: наличием различных средств автоматизации программирования; модульной структурой и применением единой системы внутренних связей, что позволяет комбинировать системные компоненты различным образом; возможностью включения в нее новых программ; независимостью программ и данных от конфигурации устройств ввода-вывода.

Возможность настройки ОС на конкретный круг задач (решаемых вычислительной системой) реализуется в процессе ее генерации, когда формируется ее рабочий вариант, ориентированный на конкретный класс решаемых задач и определенную конфигурацию технических средств.

Для ОС ЕС ЭВМ характерны четыре режима работы:

- однопрограммный режим;
- мультипрограммный режим с фиксированным числом задач;
- мультипрограммный режим с переменным числом задач;
- мультиобработка.

В однопрограммном режиме, который целесообразно применять при малой емкости основной памяти (не более 128К), в каждый момент времени выполняется только одно задание, использующее все имеющиеся в системе ресурсы.

В мультипрограммном режиме с фиксированным числом задач допускается одновременное выполнение до 15 заданий, содержащих не более 255 задач. Задания выполняются в фиксированных разделах основной памяти, максимальное количество которых определяется при генерации ОС. Этот режим целесообразно применять при емкости основной памяти свыше 128К.

В мультипрограммном режиме с переменным числом задач, который целесообразно применять при емкости основной памяти свыше 512К, распределение основной памяти между заданиями осуществляется динамически. В общем случае, чем больше задач выполняется одновременно в системе, тем больше требуется емкости внешней памяти и устройств вывода информации, в частности, накопителей на магнитных дисках и АЦПУ.

Режим мультиобработки предусматривается главным образом для многопроцессорных вычислительных систем. Например, на базе машин ЕС-1050 и ЕС-1030 можно организовать двухпроцессорную систему, содержащую два процессора, до восьми блоков основной памяти емкостью по 256К каждый, составляющих общую основную память, средства прямого управления, устройства ввода-вывода, соединенные с помощью двухканальных переключателей, пульт управления системы. В режиме мультиобработки операционная система осуществляет распределение работы между двумя или более связанными между собой процессорами. В двухпроцессорной системе в каждый момент времени в стадии выполнения находятся две задачи или одна задача и ее подзадача, выполняемые одновременно различными процессорами. Процессоры в системе рассматриваются как ресурсы, распределяемые на основе приоритетов. Любой из процессоров, а также другое системное оборудование (каналы, устройства ввода-вывода, ВЗУ, блоки основной памяти) могут быть переведены оператором в автономное состояние и таким образом исключены из числа распределяемых ресурсов. Необходимо отметить, что всякое задание, записанное для выполнения в двухпроцессорной системе, может выполняться однопроцессорной конфигурацией этой системы, когда один из процессоров работает в автономном режиме.

Система ОС обеспечивает работу ЕС ЭВМ как в режиме пакетной обработки, так и в режиме деления времени, когда осуществляется оперативное взаимодействие центрального процессора и основной памяти с удаленными и локальными абонентскими пунктами (терминалами) в реальном масштабе времени.

Операционная система ЕС ЭВМ состоит из управляющей программы и обрабатываемых программ (рис. 14.3). Основные функции ОС реализуются с помощью управляющей программы. Именно она определяет порядок выполнения обрабатываемых программ и проблемных программ потребителей в различных режимах работы и обеспечивает их необходимым сервисом.

В состав обрабатываемых программ входят трансляторы, сервисные программы, поставляемые разработчиком, а также обра-

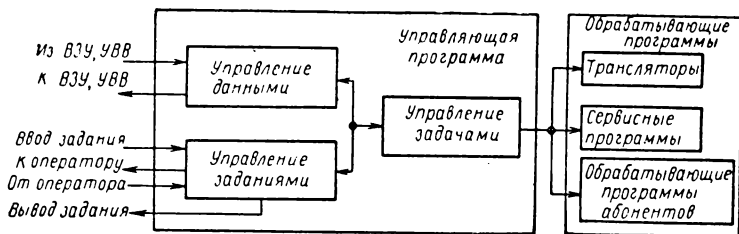


Рис. 14.3. Структура ОС ЕС

батывающие программы, написанные абонентами. Набор обрабатывающих программ может быть расширен усилиями как разработчика, так и абонентов.

Обрабатывающие программы предназначены для сокращения объема работы и времени, затрачиваемых на написание, подготовку и выполнение проблемных программ абонентов.

Трансляторы операционной системы предназначены для следующих языков программирования; ФОРТРАН IV, АЛГОЛ-60, КОБОЛ-65, ПЛ/1, АССЕМБЛЕР, РПГ (генератор программ отчетов) — язык для представления данных в табличной форме. Используя эти языки программирования, программист имеет доступ к средствам операционной системы. Результатом работы трансляторов являются объектные программы (модули) стандартного формата. Объединение объектных модулей в один модуль осуществляется редактором связей.

Сервисные программы содержат редактор связей, загрузчик, программу СОРТИРОВКА—ОБЪЕДИНЕНИЕ и программы-УТИЛИТЫ.

Редактор связей, как уже отмечалось, предназначен для объединения объектных модулей в одну программу, которую можно загружать в основную оперативную память. Он дает возможность вносить изменения в программу без повторной трансляции или без повторного ассемблирования готовой программы (повторно транслируются только те разделы программы, в которые внесены изменения), а также разделять программу на части (сегменты), если она слишком велика для имеющейся емкости основной памяти.

Загрузчик выполняет те же функции по редактированию программ, что и редактор связей, но, кроме того, непосредственно осуществляет загрузку отредактированной программы в основную память.

Программа СОРТИРОВКА—ОБЪЕДИНЕНИЕ, являющаяся по своему функциональному назначению многоплановой, используется для сортировки или объединения записей фиксированной или переменной длины в определенном (возрастающем или убывающем) порядке. Операции сортировки и объединения могут выполняться для ввода, вывода и промежуточного хранения ин-

формации на магнитных лентах или дисках. Программа СОРТИРОВКА—ОБЪЕДИНЕНИЕ используется независимо от других программ. Кроме того, она может непосредственно использоваться программой, составленной транслятором с языка КОБОЛ.

Программы-УТИЛИТЫ выполняют следующие функции:

— передачу информации из одного запоминающего устройства или устройства ввода-вывода в другое (передача производится с изменением формата и редактированием);

— редактирование, перераспределение и обновление программ и данных в библиотеке системы;

— изменение индексной структуры каталога системной библиотеки;

— печать сведений о наличии данных и программ, каталогизированных в библиотеке системы.

14.9. УПРАВЛЯЮЩАЯ ПРОГРАММА ОС ЕС

Функции управляющей программы ОС (см. рис. 14.3) можно разделить на три основные группы [33]:

- управление данными;
- управление заданиями;
- управление задачами.

Управление данными — это комплекс программ, предназначенных для управления процессами организации, идентификации, хранения, каталогизирования и выборки всех данных, обрабатываемых в системе. Они позволяют программисту хранить, модифицировать и обращаться к программам и данным, хранящимся в памяти прямого доступа (ВЗУ на магнитных дисках), предоставляют ему единый способ идентификации и размещения программ и данных и единые методы обращения к данным после их размещения.

Вся операционная система, проблемные программы и данные, обрабатываемые этими проблемными программами, хранятся в виде библиотеки в ВЗУ. Библиотека состоит из наборов данных. Набор данных представляет собой совокупность логически связанных данных, имеющую имя и описанную в системе. Все наборы данных каталогизируются, так что библиотека имеет встроенный каталог, который используется для идентификации и обнаружения любого типа данных.

Каталогизирование данных осуществляется одинаково независимо от их принадлежности к тому или иному типу. Отличие различных типов данных заключается в самом характере этих данных и способах их использования. Так, набор записей данных обрабатывается программой абонента, исходная программа — транслятором, объектная программа — редактором связей.

Каталог, представляющий собой совокупность иерархически упорядоченных индексов, располагается в запоминающем устройстве прямого доступа (на пакете дисков). В ОС предусмотрены

средства для конструирования библиотечного каталога, состоящего из нескольких уровней индексов. Имя набора данных, помещенное в каталог, может быть составным, состоящим из нескольких слов. Каждое слово такого имени соответствует определенному уровню индексов каталога.

В ОС имеется обширный набор программ-утилит, предназначенных для упорядочения (классификации) данных в библиотеке и управления использованием ВЗУ. В этом наборе имеются программы для перегруппировки библиотеки набора данных, для создания, переименования или вычеркивания индексов или наборов данных, для периодического обследования библиотеки с целью уточнения ее структуры и содержания.

Программист избавлен от необходимости написания программ доступа к своим наборам данных для каждой проблемной программы. Для этого в ОС имеется набор программ ввода-вывода (программ методов доступа). Программы методов доступа предназначены для обработки логических и физических записей, находящихся в ВЗУ и УВВ и размещенных в последовательном или в произвольном порядке, а также для обработки сообщений, полученных с удаленных устройств в произвольные моменты времени. Они, кроме того, автоматически выполняют функции по совмещению обработки с операциями ввода-вывода. Информацию, необходимую для отыскания требуемых наборов данных, программы ввода-вывода получают из описаний этих наборов, где, в частности, есть такие сведения, как организация данных, физическое размещение, размер наборов данных.

В операционной системе заложена возможность автоматического разграничения доступа к защищенным наборам данных, хранящимся в ВЗУ. Функции по организации разграничения доступа к данным выполняются программами управления данными и заключаются в следующем:

— в разрешении доступа каждому зарегистрированному абоненту (оператору, программисту) к санкционированной ему информации. Доступ разрешается по паролю, присвоенному абоненту на определенный период времени и однозначно устанавливающему (идентифицирующему) его личность;

— в блокировке доступа к защищенным наборам данных по запросу абонента, которому доступ к этим данным не санкционирован;

— в автоматической регистрации обращений к защищенным наборам данных. При каждом обращении фиксируется пароль абонента, обратившегося с запросом, имя набора данных, к которому производится обращение, характеристика содержания запроса, время обращения. Вся эта информация составляет специальный набор данных.

Операционная система построена таким образом, что программа обработки набора данных не нуждается в изменении в зависимости от того, какой тип ВЗУ или УВВ используется для

хранения этого набора или регистрации результатов обработки. Следовательно, программисту при составлении своей программы не нужно учитывать, какие именно типы устройств имеются в конкретной системе, его программа может быть выполнена системами с различными типами УВВ и ВЗУ.

Управление заданиями — это комплекс программ, предназначенных для управления процессом прохождения через вычислительную систему непрерывного потока заданий с минимальным вмешательством оператора или совсем без его вмешательства. Сведение до минимума степени участия оператора в управлении обработкой заданий не только способствует увеличению пропускной способности системы, но и гарантирует меньшую подверженность этих заданий ошибкам оператора.

В общем случае управление заданиями имеет следующие функции:

- анализ входного потока заданий, во время которого осуществляется просмотр введенных наборов данных, выделение, интерпретация и анализ управляющих операторов, подготовка необходимых управляющих таблиц, содержащих описание каждого задания;

- распределение УВВ и ВЗУ: выделение необходимых устройств ввода-вывода, выделение требуемого объема памяти на устройствах прямого доступа, выдача инструкций оператору для установки томов на НМЛ или устройствах прямого доступа;

- выбор заданий для выполнения последовательно или в соответствии с приоритетами;

- связь между оператором и системой.

Перед обработкой задания оно может быть объединено с другими заданиями для образования входного пакета заданий, который записывается на магнитной ленте или в памяти прямого доступа. Формирование входного пакета может производиться одновременно с обработкой заданий.

В процессе обработки заданий получаемая выходная информация может формироваться в выходной пакет, который запоминается на магнитной ленте или на дисках. Выходной пакет содержит, например, объектные программы, наборы исходных программ, наборы программных тестов, сообщения для программиста из операционной системы, выходы проблемных программ и т. д. После составления выходного пакета он обрабатывается в соответствии с требованиями программиста (например, производится печать наборов программ или перфорация колоды карт объектной программы). Обработка выходного пакета может производиться одновременно с обработкой заданий.

По желанию программиста входной набор данных, необходимый для выполнения его объектной программы, может быть помещен непосредственно в задание как часть общего входного пакета. Точно также выходные наборы данных, выдаваемые программой, могут составлять часть общего выходного пакета.

В соответствии с выполняемыми функциями программы управления заданиями разделяются на две части — планировщик заданий и главный планировщик.

Планировщик заданий предназначен для подготовки заданий к выполнению и планирования их выполнения либо последовательно (последовательный планировщик), либо параллельно с учетом приоритетов (приоритетный планировщик). Он выполняет три функции: чтение — интерпретацию, инициирование — завершение и обработку системного вывода.

Функции чтения — интерпретации выполняются программой интерпретатором планировщика заданий. Интерпретатор читает операторы управления заданиями, анализирует их содержание и строит таблицы, которые используются во время инициирования и выполнения пунктов (шагов) задания. В случае приоритетного планирования интерпретатор обеспечивает занесение управляющей информации в очередь входных работ по приоритетам, а также перепись данных из входного потока заданий на запоминающее устройство прямого доступа. Функции чтения — интерпретации могут выполняться либо одновременно, либо независимо от обработки заданий.

Функции инициирования — завершения выполняются программой инициатором планировщика заданий. Программа инициатор обеспечивает: выборку очередного пункта задания для выполнения из входного потока или из очереди входных работ, анализ требований на устройства ввода-вывода от пунктов задания и выделение им этих устройств, выдачу инструкций оператору об установке томов и проверку правильности установки томов на соответствующих устройствах. Для уменьшения времени простаивания системы оператор может заранее установить помеченные тома на любом свободном ВЗУ. Если необходимый том уже установлен, процедура выдачи сообщения о необходимости установки тома пропускается. Это особенно удобно в случае, когда планирование работы обычно выполняется по заранее установленному, повторяемому образцу. В такой ситуации оператору обычно известно не только то, какие тома должны использоваться, но и в какой последовательности они используются.

Функции обработки системного вывода реализуются программами системного вывода планировщика заданий. В зависимости от используемого в системе варианта планировщика заданий эти программы могут обеспечивать запись выходных наборов данных либо на одно устройство вывода, либо одновременно на несколько таких устройств. Запись, как и чтение, может выполняться либо одновременно, либо независимо от обработки заданий.

Приоритетный планировщик планирует порядок, в котором должны обрабатываться задания, в соответствии с назначенными для заданий приоритетами и с учетом имеющихся в наличии ресурсов вычислительной системы. При параллельном планировании обрабатывается несколько заданий или пунктов задания одно-

временно и осуществляется совместное использование ресурсов системы.

Главный планировщик предназначен для выполнения функций прямой и обратной связи между оператором и системой, т. е. он обрабатывает команды оператора и выдает ему сообщения.

Со стороны оператора допускаются следующие типы команд:

— команды, воздействующие на задание. Они вызывают изменение статуса задания, например, изменение приоритета задания, отмену или временную приостановку выполнения задания;

— команды, воздействующие на систему. Они вызывают изменение действий, выполняемых планировщиком заданий. Например, это команда, которая информирует систему о подключении нового устройства для входного потока заданий, или команда об устройстве, которое не может использоваться в течение определенного промежутка времени;

— команды, представляющие собой запросы на информацию о состоянии системы, о статусе некоторых заданий;

— команды, несущие ответы на системные или программные запросы или сообщаемые системе текущее время и дату.

Команды оператора обычно вводятся в систему с такого устройства, как пультовая печатающая машинка, однако они могут быть помещены в виде отдельных операторов в поток ввода.

Программист сообщает системе о своем задании или пунктах задания на языке управления заданиями. Для этого используются управляющие операторы управления заданиями. Например, связь между программистом и планировщиком заданий осуществляется с помощью следующих управляющих операторов: оператор задания, исполнительный оператор, оператор определения данных, командный оператор, разделительный оператор, пустой оператор. Параметры, закодированные в этих операторах, дают возможность планировщику заданий регулировать выполнение заданий и пунктов заданий, отыскивать и размещать данные, распределять ресурсы ввода-вывода, осуществлять связь с оператором.

Оператор задания обозначает начало задания, а во входном потоке, кроме того, — конец управляющих операторов предшествующего задания.

Исполнительный оператор используется для обозначения начала пункта задания и идентификации первого выполняемого модуля загрузки или каталогизированной процедуры (каталогизированная процедура — это набор операторов управления заданиями, которые помещаются в набор данных с библиотечной организацией, называемый библиотекой процедур).

Оператор определения данных описывает наборы данных, используемые в пункте задания, и запрашивает распределение ресурсов ввода-вывода. Параметры этого оператора идентифицируют набор данных, задают информацию о том и об устройстве, о размещении набора данных, описывают метки набора данных.

Командный оператор используется оператором системы для помещения команд во входной поток, которые могут запустить или остановить системные вводные или выводные устройства, запросить вывод на печать, выполнить ряд других функций оператора.

Разделительный оператор является маркером во входном потоке. Он используется для включения данных во входной поток, для разделения данных от следующих за ними управляющих операторов.

Пустой оператор, являющийся во входном потоке также маркером, используется для обозначения конца задания, в котором последний оператор определяет набор данных во входном потоке.

Кроме указанных, имеются и другие управляющие операторы управления заданиями, используемые для выполнения таких действий, как передача наборов от одного пункта задания к другому (при этом не затрачивается время на поиск требуемого набора), обеспечение одновременного использования одних и тех же наборов данных несколькими независимыми заданиями при работе в мультипрограммном режиме, сокращение записей операторов путем ссылок на уже существующие имена наборов данных и управляющие операторы и др.

Управление задачами — это комплекс программ, предназначенных для управления процессом выполнения задач при различных режимах работы системы, в том числе для управления процессом одновременного выполнения нескольких задач по обработке данных. Под задачей, как уже отмечалось, понимается работа, которая должна быть выполнена программой, указанной в пункте задания. Задача — это тот наименьший объем работы, который может претендовать на использование ресурсов системы как машинных (машинное время процессора, емкость оперативной памяти, устройства ввода-вывода), так и программных, таких как программы системных библиотек.

Функции управления задачами сводятся к следующему:

- распределение основной памяти;
- загрузка программ в основную память;
- планирование использования программ в основной памяти;
- управление выполнением задач в соответствии с определенным приоритетом;
- переключение управления от одной задачи к другой;
- защита задач от непредусмотренного взаимного влияния.

Последние три функции выполняются в системах, предоставляющих возможность одновременного решения нескольких задач по обработке данных. Пункты, входящие в состав одного и того же задания, выполняются последовательно. Совмещение выполнения возможно только для пунктов, принадлежащих разным заданиям. Если некоторая программа неоднократно и одновременно требуется для нескольких различных задач, то она загру-

жается в основную память не многократно, а один раз. При этом экономится память и сокращается время загрузки программ в нее.

Основной компонентой программ управления задачами является СУПЕРВИЗОР. Часто используемые программы СУПЕРВИЗОРА входят в состав ядра операционной системы, которое всегда находится в основной памяти.

После того как пункт задания в виде задачи передан СУПЕРВИЗОРУ, последний передает управление первому загрузочному модулю и наблюдает за его выполнением. В режиме выполнения одной задачи ресурсы для нее выделяются немедленно по запросу на эти ресурсы. При мультипрограммной работе СУПЕРВИЗОР распределяет ресурсы, основываясь на классе задач и их приоритете. После выполнения очередной задачи об этом уведомляется планировщик заданий и управление снова передается ему.

Эффективность использования возможностей управления заданиями зависит от решения таких вопросов, как назначение приоритета, порядок образования новых задач, синхронизация событий, распределение основной памяти, защита памяти, передача и совместное использование основной памяти, установка временных интервалов, окончание задачи, квантование времени. Остановимся кратко на этих вопросах.

В режиме одновременной работы с несколькими задачами порядок, в котором задания выбираются для выполнения, и удовлетворение запросов на машинные и программные ресурсы определяется классом задания и приоритетом задания внутри класса. Классификация заданий позволяет управлять типами заданий, одновременно выполняемых в системе. Первоначально класс задания и приоритет указываются в соответствующем управляющем операторе. Оператор системы может модифицировать приоритет вплоть до момента, когда задание будет выбрано для выполнения. В мультипрограммном режиме работы приоритеты могут изменяться управляющей программой в процессе выполнения задач. Когда очередной пункт задания устанавливается планировщиком заданий для выполнения, приоритет задания внутри класса используется для назначения диспетчерского приоритета и предельного приоритета. Первый из них используется СУПЕРВИЗОРОМ при рассмотрении запросов на ресурсы системы, а второй определяет ограничения на изменение диспетчерского приоритета и используется для контроля над динамическим присвоением приоритетов.

Образование новых задач осуществляется в процессе обработки заданий. Например, при работе системы в мультипрограммном режиме программист с помощью специальной макрокоманды может образовать новую задачу, которая является в сущности подзадачей порождающей ее задачи. Предельный и диспетчерский приоритеты новой задачи те же, что и для задачи, ее породившей. Новые задачи образуются также, когда оператор выполняет опе-

рацию начальной загрузки или использует команды, запускающие программные компоненты планировщика заданий.

Под синхронизацией событий понимается комплекс некоторых процедур, реализуемых в ходе выполнения задач. Синхронизация событий задерживает выполнение задачи до наступления некоторого специфического события, например, до тех пор, пока не будет введена требуемая запись. Программист с помощью специальной макрокоманды может задержать выполнение задачи до наступления некоторого числа событий.

Существует такая форма синхронизации, которая позволяет задачам совместно использовать последовательно подключаемые ресурсы. В качестве такого ресурса может быть, например, таблица, содержание которой обновляется при выполнении многих задач. Выполнение этих задач должно быть организовано таким образом, чтобы каждая задача заканчивала использование таблицы прежде, чем другая задача получит к ней доступ. При обращении к подобному ресурсу образуется очередь задач, требующих доступа к нему, и сам доступ организуется по последовательному принципу.

В системе под управлением СУПЕРВИЗОРА осуществляется динамическое распределение памяти. Запросы на выделение или освобождение основной памяти могут указываться в задании в явном виде. Возможен также неявный запрос, например, когда необходимо вызвать программу из библиотеки или когда заканчивается использование программы и происходит неявное освобождение основной памяти. Запросы на основную память могут быть условными или безусловными. В последнем случае задача не может выполняться без выделения ей указанного в запросе объема памяти.

При работе в мультипрограммном режиме обязательна организация защиты памяти. Функции защиты памяти реализуются системой защиты памяти, которая представляет собой комплекс программно-аппаратных средств, обеспечивающих предотвращение взаимного искажения массивов информации, одновременно находящихся в оперативной памяти. Основные функции системы защиты памяти заключаются в следующем: защита программных компонентов операционной системы, защита участков памяти, занимаемых проблемными программами абонентов, от воздействия выполняемых программ, защита отдельных участков массивов, блокировка обращения на считывание из общего массива данных при обновлении информации в нем, защита личных массивов абонентов от использования другими абонентами. Системы защиты памяти строятся по различным принципам, отвечающим возможностям по размещению программ, допустимым затратам аппаратуры и машинного времени, а также методам организации вычислительного процесса. В настоящее время наибольшее распространение получили системы защиты с граничными регистрами, системы защиты по маскам и особенно системы защиты по ключам.

При совместном использовании основной памяти несколькими задачами в задании предусматривается образование так называемых подпулов. Подпул — это блок (область) основной памяти определенного объема (например, 2К), выделяемая для конкретной задачи. В мультипрограммном режиме подпулы могут передаваться от одной задачи к другой или совместно использоваться несколькими задачами.

Установка временных интервалов осуществляется датчиком временных интервалов, или интервальным таймером. Таймер используется управляющими программами для измерения времени, затрачиваемого на выполнение пункта задания, и для проверки того, что время, отведенное некоторому пункту задания, не превышено.

Таймер используется для выполнения следующих функций:

- посылка запроса СУПЕРВИЗОРУ из любой точки программы на выдачу даты и текущего времени;

- посылка запроса СУПЕРВИЗОРУ после заданного интервала времени. По этому запросу СУПЕРВИЗОР удерживает задачу в состоянии ожидания на вполне определенное время (например, пока не истечет некоторый интервал реального времени) или обеспечивает продолжение выполнения задачи до истечения некоторого интервала времени, после чего управление передает заданной точке в выполняемом программном модуле;

- установление того, сколько истекло запрошенного времени, или отмена ранее заданного интервала.

При работе в мультипрограммном режиме таймер может использоваться для ограничения времени, затрачиваемого на выполнение каждого задания.

Система в любом случае получает информацию о прекращении выполнения задачи независимо от того, завершилась она нормально или аварийно. Аварийное прекращение выполнения задачи может вызываться любой программой, выполняемой в рамках этой задачи. Программист располагает средствами для указания того, какие типы программных прерываний должны обрабатываться его программой, а какие — СУПЕРВИЗОРом.

Квантование (разделение) времени является дополнительным средством для создания удобств абонентам. Оно используется при мультипрограммном режиме работы системы. Задачи, которые должны выполняться в режиме квантования времени, определяются заблаговременно. Заранее устанавливается также величина кванта времени и порядок использования задачами центрального процессора (в соответствии с приоритетами задач). Задача, не входящая в группу квантования (т. е. не входящая в группу задач, выполняющихся в режиме квантования), подготавливается и выбирается в соответствии с ее текущим приоритетом. Следовательно, такая задача выбирается только тогда, когда она является задачей более высокого приоритета по сравнению с задачами группы квантования и находится в состоянии готовности.

АППАРАТНЫЕ СРЕДСТВА ТЕЛЕОБРАБОТКИ ДАННЫХ

15.1. ОБЩИЕ СВЕДЕНИЯ ОБ ОРГАНИЗАЦИИ ТЕЛЕОБРАБОТКИ

В составе ЕС ЭВМ имеются аппаратные и программные средства, дающие возможность ЭВМ обмениваться информацией по линиям связи с ее удаленными источниками и потребителями, т. е. производить телеобработку данных. Это позволяет эффективно решать разнообразные задачи:

- сбора, хранения и поиска информации;
- учета и планирования;
- управления и обслуживания.

Кроме собственно ЭВМ в аппаратные средства телеобработки входят:

- устройства сопряжения ЭВМ с аппаратурой передачи данных — мультиплексоры передачи данных (МПД);
- аппаратура передачи данных (АПД);
- абонентские пункты (АП), содержащие устройства ввода-вывода и отображения информации, в общем называемые оконечным сбородованием данных (ООД).

При большом удалении абонентских пунктов от ЭВМ время передачи информации по линиям связи становится не только соизмеримым, но может превосходить время ее обработки в машине. Поэтому, в зависимости от требований к скорости передачи информации и быстрей действия процессорной аппаратуры, средства АПД выполняют различного быстрей действия. Например, модели ЕС ЭВМ соответственно их быстрей действию комплектуют средствами АПД, способными передавать цифровую информацию со скоростями от 50 до 48 000 бит/с.

Как и АПД, мультиплексоры передачи данных имеют различную пропускную способность и разные возможности по числу обслуживаемых ими абонентов. Абонентские пункты обеспечивают потребителям, удаленным от вычислительных центров, широкие возможности дистанционного доступа к ЭВМ. В ЕС ЭВМ предусмотрены абонентские пункты, различные по составу аппаратуры ввода-вывода и отображения информации.

В целом совокупность ЭВМ, абонентских пунктов и соединительной сети составляет систему телеобработки. Существуют

понятия: *аппаратурные звенья системы (пункты) и звенья передачи данных* (или соответственно *звенья и звенья данных*). Основным звеном каждой системы телеобработки является ЭВМ; абонентские пункты — массовые звенья. Звено данных образуют две ЭВМ, ЭВМ и один или несколько абонентских пунктов либо два абонентских пункта, непосредственно связанных между собой. Соответственно этому различают звенья данных двух типов: *двухпунктовые* («пункт—пункт») и *многопунктовые*.

Двухпунктовые звенья данных могут работать как в режиме *соперничества*, так и *подчинения*, многопунктовые — всегда только в режиме *подчинения*. В режиме соперничества оба пункта равноценны в выполнении функций управления передачей информации. Для исключения конфликтных ситуаций, которые могут возникать при одновременном запросе разрешения передачи, один из соперничающих пунктов назначается основным, а другой — второстепенным. В режиме подчинения один из пунктов всегда является управляющим, ответственным за работу всей системы телеобработки.

Процессы телепередачи информации в звеньях данных осуществляются с помощью специальных стандартных процедур управления обменом. Процедуры управления состояются из служебных символов и формируемых из них последовательностей. Символы (табл. 8 и 9 приложения) для работы, например, по телефонным каналам связи кодируются согласно таблице стандартных 7-элементных кодов обмена информацией КОИ-7 (табл. 2.3).

Процесс обмена информацией разбивается на этапы:

- установление соединения звена данных;
- передача информации;
- разъединение звена данных.

Управление потоками сообщений, а также коммутацией сообщений, сбором данных, обработкой данных, справок, запросов, заданий и т. п. осуществляется программами программного обеспечения.

Программное обеспечение системы телеобработки дает возможность использовать все средства операционных систем. Оно также осуществляет контроль и диагностику технических средств с целью обнаружения и исправления неисправностей.

Функции программного обеспечения делятся на две части:

- установление порядка управления каналами связи с учетом типов подключаемых АП, передачи информации и подключения АП к мультиплексорам передачи данных; управление движением сообщений в соответствии с пожеланиями пользователей;
- обработка входящих и исходящих из ЭВМ сообщений.

Соответственно этому первая часть обеспечивается программами управления передачей, а вторая — обработки сообщений. Разграничение программ и соответствующая организация их функционирования делают независимыми процессы ввода-вывода информации и ее обработки.

Программы управления передачей сообщений составляются с учетом конфигурации системы телеобработки, методов передачи сообщений и т. п. Программы обработки составляются пользователями с учетом специфики решаемых задач и способов обработки информации.

15.2. СТРУКТУРЫ ЗВЕНЬЕВ ПЕРЕДАЧИ ДАННЫХ

Обмен информацией между двумя ЭВМ или между ЭВМ и абонентскими пунктами может происходить по следующим каналам связи:

- телеграфным некомутируемым;
- телеграфным комутируемым;
- телефонным некомутируемым (выделенным);
- телефонным комутируемым;
- широкополосным;
- физическим — обычным проводным или кабельным линиям.

Последний способ связи возможен при небольшом удалении друг от друга источников и приемников информации. Широкополосные каналы (с полосой 60,6—107,7 кГц) применяются в высокоскоростных линиях связи между удаленными друг от друга ЭВМ. Такие линии позволяют, например, передавать информацию со скоростями до 48 000 бит/с. В наиболее распространенных случаях связи ЭВМ с абонентскими пунктами, при их удалении от сотен метров до сотен и тысяч километров и скоростях передачи от 50 до 300 бит/с, используются телеграфные (Тлг), а свыше 300 бит/с — телефонные (Тлф) линии связи.

Для передачи информации могут использоваться режимы:

- дуплексный;
- полудуплексный;
- симплексный.

В дуплексном режиме информация одновременно может передаваться как в прямом, так и в обратном направлениях. Реализация этого режима требует наибольших аппаратных затрат. Поэтому он применяется в основном в таких звеньях передачи данных, как ЭВМ—ЭВМ или ЭВМ — комплексы абонентских пунктов.

Достаточно удобным и более экономичным является полудуплексный режим. В этом режиме информация в прямом и обратном направлениях передается по одной электрической цепи в разные моменты времени. Такая передача приемлема для звеньев данного типа: ЭВМ — абонентский пункт, абонентский пункт — абонентский пункт.

Односторонняя связь, основанная на передаче информации только в одном направлении (симплексный режим), применяется в специальных системах сбора данных.

Варианты структуры звеньев данных, реализуемых на основе некоммутируемых телеграфных и телефонных каналов, изображены на рис. 15.1.

Мультиплексор передачи данных МПД, представляющий собой автоматический многоканальный коммутатор, предназначен для сопряжения ЭВМ с аппаратурой передачи данных. По командам, поступающим из машины, он направляет информацию в нужные линии связи. Принимая информацию с разных направлений, поступающую по каждой линии связи с относительной низкой скоростью по сравнению с возможностями электронных схем ЭВМ; мультиплексор осуществляет ее временное уплотнение в канале непосредственного ввода в машину. Абонентские пункты через линии связи обслуживаются МПД в режиме разделения времени.

Для повышения надежности связи каждый мультиплексор может подключаться к двум трактам (подканалам) ввода-вывода мультиплексного канала ЭВМ. По каждому тракту присоединение осуществляется через цепи стандартного сопряжения устройства ввода-вывода (через интерфейс ввода-вывода ИВВ).

С аппаратурой передачи данных мультиплексоры соединяются через унифицированные цепи сопряжения, называемые стыками. Существует три типа стыков:

C_1 — для сопряжения аппаратуры передачи данных с каналами связи;

C_2 и C_3 — для сопряжения аппаратуры передачи данных с аппаратурой обработки данных.

Проще всего осуществляется связь с телеграфными аппаратами через выделенные телеграфные линии связи (звено данных I на рис. 15.1). В этом случае МПД непосредственно сопрягается с линией связи через стык C_1 . Телеграфный аппарат может подключаться к линии по симметричной или несимметричной схеме. Соответственно этому будет использоваться четырехпроводная или двухпроводная схема включения. Обмен данными между МПД и телеграфным аппаратом производится в пятиэлементном между-

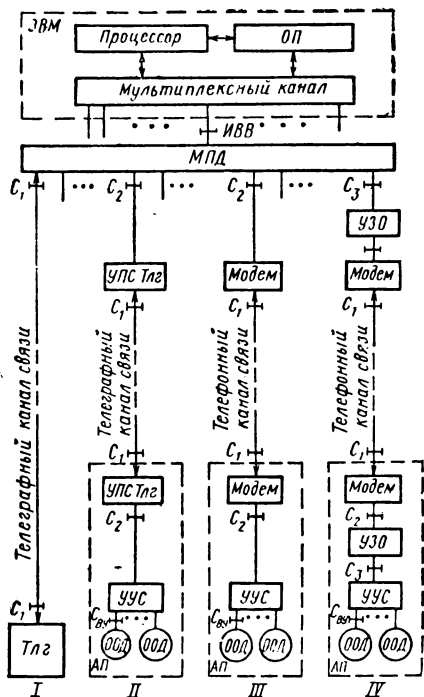


Рис. 15.1. Структуры звеньев данных на основе некоммутируемых телеграфных и телефонных каналов связи

народном телеграфном коде МТК-2 последовательностями токовых и бестоковых посылок.

При необходимости передачи данных на большие расстояния и в особенности по выделенным каналам телеграфной сети мультиплексор и устройство управления и сопряжения (УУС) абонентского пункта (АП) соединяются с телеграфным каналом через специальные устройства преобразования сигналов УПС Тлг (звено данных *II*). Устройства УПС Тлг к МПД и УУС подключаются через стыки типа C_2 . При автоматическом установлении соединения через стык такого типа данные передаются параллельно-последовательно потетрадно. В телеграфный канал УПС Тлг выдает данные в виде последовательностей токовых и бестоковых импульсных посылок.

Для передачи данных по телефонным линиям связи необходимо преобразование импульсов постоянного тока в частотно-модулированные сигналы и обратно. Модуляция передаваемых сигналов и демодуляция принимаемых также осуществляется с помощью устройств преобразования сигналов УПС. Если конструктивно модулятор и демодулятор объединяются в одном блоке, то вместо УПС применяется термин «модем». К телефонным каналам связи модемы подключаются через стыки C_1 .

По телефонным линиям связи передача данных осуществляется методом тонального телеграфирования (в полосе 300—3400 Гц). При этом комбинации двоичных символов передаются в виде последовательностей посылок частотно- или фазово-модулированных синусоидальных колебаний. Передаваемые сообщения кодируются — семиэлементным стандартным кодом КОИ-7. Возможен также так называемый «прозрачный режим» (для символьно-ориентированных звеньев данных), при котором данные могут передаваться непосредственно в восьмиэлементном машинном коде КОИ-8.

Скорости преобразования и передачи данных в модемах соответствуют пропускным способностям телефонных и широкополосных каналов. В системах телеобработки ЕС ЭВМ применяются модемы, рассчитанные на пропускные способности каналов: 200, 1200, 2400, 4800 и 48 000 бит/с.

С МПД и УУС модемы могут непосредственно соединяться через стыки C_3 (звено данных *III*). В этом случае они преобразуют параллельно-последовательные коды в последовательные и обратно, как и УПС Тлг.

Если к достоверности передачи данных предъявляются повышенные требования, то в структуры звеньев данных включают специальные устройства защиты от ошибок УЗО (звено данных *IV*). Они предназначены для обнаружения и исправления ошибок, возникающих в каналах связи и, частично, в самой аппаратуре передачи данных.

Устройства защиты от ошибок соединяются с модемами через стыки типа C_2 . Соединение же УЗО с МПД и УУС осуществляется через стыки C_3 . Эти стыки позволяют одновременно вести побайт-

ную передачу информации в обоих направлениях, а также передавать сигналы управления и индикации.

Для различных по пропускной способности каналов выпускаются и соответствующие УЗО. В номенклатуру средств телеобработки ЕС ЭВМ входят четыре типа устройств защиты: УЗО-200, УЗО-1200, УЗО-4800 и УЗО-48000. Помимо скоростных характеристик, они также различаются алгоритмами работы. Одинаковым в логике работы у всех них является возможность автоматического повторения блока или блоков данных по запросу приемного устройства.

Передаваемая информация делится в УЗО на блоки фиксированной длины, в каждый из которых, содержащий служебные информационные знаки, вводятся контрольные биты, образуемые, например, с помощью циклического кодирования. Длина блока зависит как от качества канала связи, так и от скорости передачи данных.

На абонентских пунктах УУС соединяются с оконечным оборудованием данных ООД (внешними устройствами ВУ) через стыки-интерфейсы $C_{ву}$. Количество и назначение соединительных линий в этих стыках зависят от конструкции конкретных типов ВУ.

Обмен данными между пунктами системы телеобработки может осуществляться также через абонентские системы телефонной и телеграфной связи. В этих случаях такие системы предоставляют услуги абонентам в организации линий связи. Абонентские системы строятся по радиально-узловому принципу с установкой в узлах автоматических или ручных коммутационных станций.

На рис. 15.2 показаны структуры связей с абонентскими пунктами через автоматическую телефонную станцию (АТС) и автоматическую станцию абонентского телеграфа (АТА).

При связи через АТС к соответствующему выходу МПД подключается автоматическое вызывное устройство (АВУ). Оно обеспечивает автоматическое соединение с абонентскими пунктами, посылая соответствующие запросы. Установление соединений со

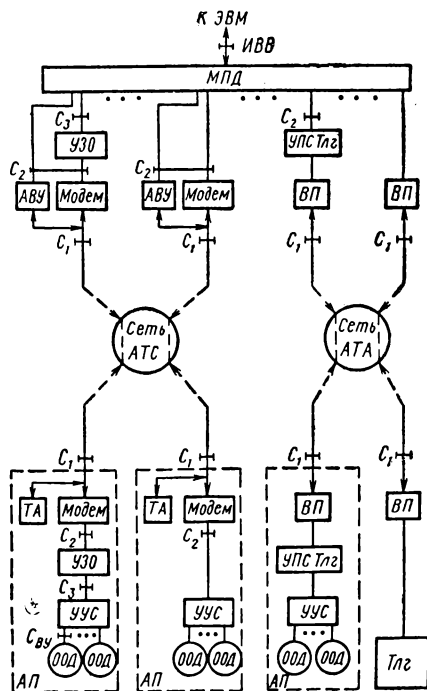


Рис. 15.2. Структуры связей с абонентскими пунктами через автоматическую телефонную станцию АТС и автоматическую станцию абонентского телеграфа АТА

стороны абонентских пунктов производится с помощью телефонных аппаратов (ТА). Машина вызывается набором номера. Получив сигнал ответа, оператор начинает «разговор» с машиной с помощью устройства, имеющего клавиатуру (печатающая машинка, телеграфный аппарат или специальное вводно-выводное устройство — «терминал» — быстродействующий буквопечатающий аппарат со специальной клавиатурой, имеющий больше знаков, чем обычный телеграфный аппарат), или включает какое-либо автоматическое быстродействующее устройство ввода-вывода.

Вызов абонента или машины через АТА напоминает вызов в телефонной сети. От вызывного прибора ВП на станцию АТА посылается сигнал вызова. Устанавливаясь в состояние готовности приема импульсного кода номера от вызывающего абонента, станция посылает ему разрешающий сигнал. После приема этого сигнала на станцию передается номер вызываемого абонента. Если соответствующая аппаратура абонентского пункта (ЭВМ) включена, то вызывающей станции посылается положительный ответ и производится либо передача запрашиваемого сообщения, либо подготовка к приему сообщения. При неготовности к работе аппаратуры АП запрашивающей станции может быть выдан отрицательный ответ. По окончании передачи сообщения разъединение звена данных производится сигналом отбоя.

Кроме аппаратуры автоматического вызова, структуры звеньев данных с коммутируемыми линиями связи аналогичны рассмотренным на рис. 15.1.

Связь с удаленными абонентскими пунктами может устанавливаться (коммутироваться) через пульт оператора с ручным управлением. С такого пульта может осуществляться автономное управление аппаратурой абонентских пунктов.

Если применяется ручной способ установления соединений, то в состав МПД включается пульт оператора с панелями коммутации и индикации, с телефонными, телеграфными и вызывными приборами.

Для того чтобы иметь возможность в любой момент принять сигналы запроса на установление соединения (управляющие символы), в аппаратуре абонентских пунктов и МПД предусматривается дежурное состояние. В этом состоянии станции осуществляют слежение за каналами связи.

15.3. АБОНЕНТСКИЕ ПУНКТЫ

В зависимости от назначения вычислительной системы она может комплектоваться дистанционной аппаратурой (оконечным оборудованием) ввода-вывода специализированного или универсального типа. Средства телеобработки на основе ЕС ЭВМ комплектуются универсальным оконечным оборудованием. В состав его могут входить:

— телетайпы;

Устройства ввода-вывода информации для абонентских пунктов ЕС ЭВМ

Устройство	Шифр	Краткая техническая характеристика
Устройство ввода с перфокарт	ЕС-6111	<p>Скорость считывания — до 50 колонок/с</p> <p>Носитель информации — 80-колонная перфокарта (ГОСТ 6198—64)</p> <p>Режим работы — стартстопный с остановом на каждой колонке</p> <p>Способ считывания — фотоэлектрический</p> <p>Направление движения перфокарты — прямое и обратное (управление от внешних сигналов или панели управления)</p> <p>Реверс возможен после считывания 80-й колонки</p>
Устройство ввода с перфокарт	ЕС-6012	<p>Скорость считывания — до 500 карт/мин</p> <p>Носитель информации — 80-колонная перфокарта (ГОСТ 6198—64)</p> <p>Режим работы — стартстопный, покартный</p> <p>Способ считывания — фотоэлектрический</p>
Устройство ввода с перфолент	ЕС-6121	<p>Скорость считывания — до 300 строк/с</p> <p>Носитель информации — бумажная перфолента с 5 и 8 дорожками</p> <p>Режим работы — стартстопный</p> <p>Способ считывания — фотоэлектрический</p>
Устройство ввода с перфолент и карт с краевой перфорацией	ЕС-6191	<p>Скорость считывания — до 40 зн/с в стартстопном режиме и до 100 зн/с в непрерывном режиме</p> <p>Носители информации — бумажные перфоленты с 5 и 8 дорожками и карты шириной 76,2 мм или 82,6 мм с краевой перфорацией</p> <p>Режим работы — стартстопный и непрерывный</p> <p>Направление движения носителя — прямое и обратное как для лент, так и для карт с краевой перфорацией</p>
Устройство вывода на перфокарты	ЕС-7111	<p>Скорость перфорации — до 50 колонок/с</p> <p>Носитель информации — 80-колонная перфокарта (ГОСТ 6198—64)</p> <p>Режим работы — стартстопный с остановом на каждой позиции</p>
Устройство вывода на перфокарты	ЕС-7012	<p>Скорость перфорации — до 250 карт/мин</p> <p>Носитель информации — 80-колонная перфокарта (перфорация по широкой стороне карты)</p> <p>Режим работы — стартстопный, покартный</p>
Устройство вывода на перфоленту	ЕС-7121	<p>Скорость перфорации — 150—200 строк/с</p> <p>Носитель информации — бумажная перфолента с 5 и 8 дорожками</p> <p>Режим работы — асинхронный</p> <p>Способ перфорации — электромеханический</p>

Устройство	Шифр	Краткая техническая характеристика
Устройство вывода на перфоленты и карты с краевой перфорацией	ЕС-7191	Скорость перфорации — 33—50 зн/с Носители информации — бумажные перфоленты с 5 и 8 дорожками и карты шириной 76,2 мм и 82,6 мм с краевой перфорацией Режим работы — стартстопный Реверс — имеется Способ перфорации — электромеханический
Алфавитно-цифровое печатающее устройство	ЕС-7030	Скорость печати — до 830 строк/мин Носитель информации — стандартная фальцованная бумага с краевой перфорацией и подсечками Количество разрядов в строке — 128 Количество знаков (символов) в разряде — 94
Пишущая машинка «Консул-260»	ЕС-7172	Скорость печати — до 15 зн/с Носитель информации — стандартная рулонная бумага Кодирование информации — по ГОСТ 13052—74 Количество печатаемых символов — 93 на двух регистрах Количество печатных знаков в строке — 100
Устройство отображения информации на электронно-лучевой трубке с групповым управлением	ЕС-7066	Скорость передачи данных — до 2400 бит/с Количество одновременно отображенных знаков — не менее 480
Устройство отображения информации на ЭЛТ с функциональной и алфавитно-цифровой клавиатурой (7061)	ЕС-7064	Скорость передачи данных — 1200—4800 бит/с Количество одновременно отображенных знаков — до 960 Количество различных символов, которые можно отображать — 96
Шаговый накопитель на магнитной ленте	ЕС-9001	Содержит: блок магнитной ленты, блок управления с буферной памятью и алфавитно-цифровую клавиатуру Скорость набора знаков на клавиатуре — до 20 знаков/с Количество различных вводимых символов — до 96 Плотность записи на магнитной ленте — 8 строк/мм Максимальная длина зоны — 180 знаков Длина магнитной ленты — 185 м

- пишущие машинки;
- быстродействующие печатающие устройства типа АЦПУ;
- устройства перфорации и ввода-вывода ленточного типа;
- устройства перфорации и ввода-вывода карточного типа;
- накопители на магнитных лентах;
- индикаторы на электронно-лучевых трубках (дисплеи) с клавиатурой.

Краткие технические характеристики этой аппаратуры приведены в табл. 15.1. Из этого оборудования совместно с устройствами управления и согласования УУС, устройствами защиты от ошибок УЗО и модемами или УПС Тлг формируются абонентские пункты АП. Не считая самостоятельных абонентских пунктов на основе телеграфных аппаратов, связываемых с ЭВМ телеграфными линиями, для совместной работы с машинами Единой системы предусмотрены абонентские пункты девяти типов: АП-1, АП-2, АП-3, АП-4, АП-5, АП-11, АП-61, АП-63 и АП-70 (рис. 15.3). Пункты различаются возможными скоростями обмена данными, что обусловлено применяемой аппаратурой ввода-вывода и типами используемых линий связи. В свою очередь, тип используемой аппаратуры определяет возможные способы и режимы дистанционного ввода-вывода информации и, следовательно, общения с машиной или другим АП.

Для установления соединений звеньев данных абонентские пункты снабжаются переговорно-вызывными устройствами. При связи по телефонным каналам это могут быть обычные телефонные аппараты с номеронабирателями и средствами для переключений. Абонентские пункты, включаемые через АТС и АТА, должны осуществлять процедуры идентификации (опознавания номера).

Абонентский пункт АП-1 (ЕС-8501) представляет собой гибкий комплекс аппаратуры ввода-вывода, позволяющей осуществлять подготовку и передачу данных в каналы связи, принимать информацию из каналов связи, производить ее учет и перезапись с одного носителя на другой. Пункт может обмениваться информацией как со всеми моделями ЭВМ Единой системы, так и с абонентскими пунктами, совместимыми с ним по алгоритму работы и скорости передачи данных.

С ЭВМ пункт может сопрягаться через все типы мультиплексов, входящих в состав Единой системы.

Абонентский пункт может комплектоваться различно: от пишущей машинки (ЕС-7172) и устройства управления и сопряжения УУС до полного комплекта, показанного на рис. 15.3. Широкая номенклатура устройств ввода-вывода позволяет на АП-1 принимать информацию от ЭВМ или других АП на печать, перфокарты, перфоленты и карты с краевой перфорацией, а также передавать в ЭВМ или другие АП данные с клавиатуры пишущей машинки, перфокарт, перфолент и карт с краевой перфорацией. Кроме аналогичных пунктов типа АП-1, двухпунктовые и многопункто-

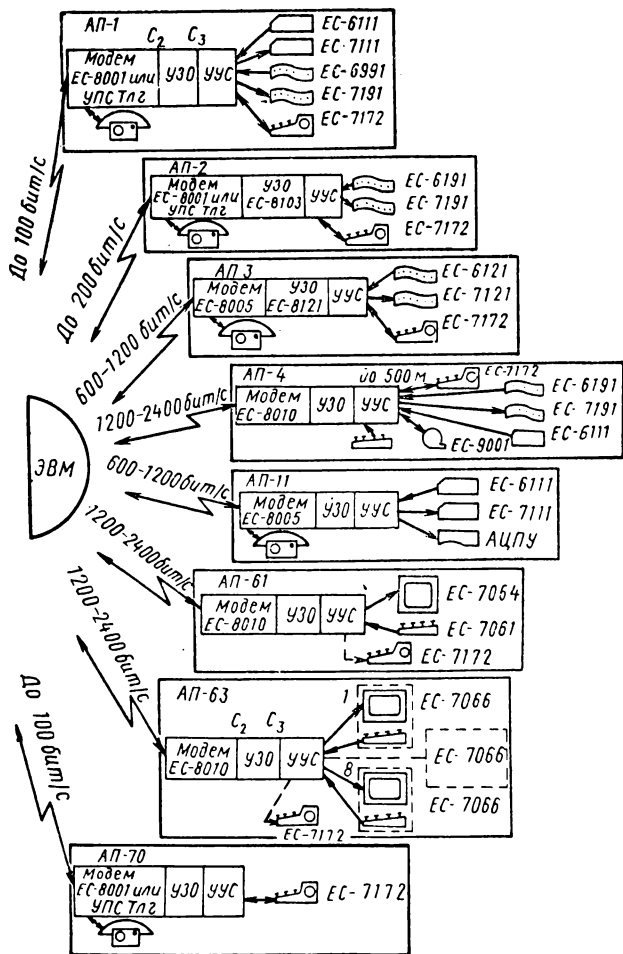


Рис. 15.3. Структура абонентских пунктов ЭС ЭВМ

вые звенья данных могут устанавливаться, например, с АП-70. С ЭВМ и другими АП в двухпунктовых и многопунктовых звеньях данных абонентский пункт АП-1 может работать как будучи подчиненным, так и равноправным (в режиме соперничества). В последнем случае, наряду с другими абонентами, он может управлять операциями передачи данных.

Абонентский пункт АП-1 имеет два независимых канала обработки информации: автономный и телеобработки. Они позволяют пункту одновременно речитать как в автономном режиме (например, подготовка и распечатка данных), так и в режиме телеобработки. По каналу телеобработки информация передается побайтно блоками переменной длины. Через модем EC-8001 (при работе

по телефонным каналам) либо через УПС Тлг (при работе по телеграфным каналам) данные передаются в коде КОИ-7 стартстопно со скоростью, не превышающей 100 бит/с.

В процессе обмена данными при телеобработке коррекция ошибок осуществляется устройствами ввода-вывода. При этом данные печатаются и перфорируются по принципу «грязной ленты», т. е. «вычеркивания» и исправления производится непосредственно на носителе информации. Это возможно благодаря стартстопному (с остановом на каждой позиции) режиму работы устройств ввода-вывода и наличию в них реверса носителей информации.

Работа аппаратуры АП-1 организуется таким образом, что при обнаружении ошибки в процессе ввода данных в ЭВМ производится повторная попытка передачи. Для этого перфолента или карта с краевой перфорацией возвращается к началу блока данных, переданного с ошибкой, и осуществляется повторное считывание. Если данные вводятся с перфокарт, то для повторного считывания перфокарта возвращается на первую колонку. Данные, переданные с ошибкой с клавиатуры пишущей машинки, исключаются путем отметки знаком подчеркивания «_» на конце блока (строки), после чего исправленная строка передается оператором повторно.

При обнаружении ошибки в процессе вывода делается повторный запрос информации. Если данные выводятся на перфоленту, то она возвращается к началу неверно принятого блока, который затем забивается. Перфокарта с информацией, содержащей ошибки, сбрасывается в специальный карман перфоратора, и повторная перфорация производится на следующей (чистой) карте. Для выделения неверно принятых данных, выводимых на печать, в конце каждого блока или строки, в которых обнаружены ошибки, делается отметка символом подчеркивания «_». Вслед за этим автоматически переводится строка и возвращается каретка.

Для повышения достоверности принимаемых и передаваемых данных в аппаратуре АП-1 используются групповые корректирующие коды с исправлением блоков (строк), содержащих ошибки, путем повторного запроса или передачи. При работе АП-1 с ЭВМ количество повторных попыток передачи определяется управляющей программой машины. В звеньях обмена данными непосредственно между абонентскими пунктами обычно при обнаружении ошибок производятся только две дополнительные попытки ввода-вывода. Если эти попытки оказываются неудачными, то вырабатывается звуковой или световой сигнал неисправности канала передачи данных и производится останов устройств ввода-вывода.

Абонентский пункт АП-2 (ЕС-8502) (рис. 15.3) предназначен для обмена данными с ЭВМ Единой системы или аналогичным АП-2 по коммутируемым и некоммутируемым телефонным каналам связи со скоростью до 200 бит/с. Данные могут передаваться синхронно в полудуплексном или дуплексном режимах.

Данный пункт может работать только в двухпунктовых звеньях данных (с ЭВМ или аналогичным АП-2), в которых абонентские пункты и ЭВМ равноправны в выполнении функций управления обменом информацией.

В автономном режиме (без выдачи в линию связи) аппаратура АП-2 позволяет одновременно с печатанием на бумаге наносить информацию на перфоленту или карты с краевой перфорацией, выводить на печать информацию с перфоленты или карт с краевой перфорацией, дублировать перфоленты и карты.

Абонентский пункт АП-2 может принимать и передавать данные автоматически без участия оператора. В этом случае управление осуществляется сигналами, поступающими по каналам связи от ЭВМ или другого абонентского пункта.

По окончании приема (передачи) данных абонентский пункт автоматически переключается в режим передачи (приема). Окончание передачи сопровождается выдачей символа конца передачи (КП). Конец приема распознается по полученному символу КП. Разъединение связи сопровождается выдачей пары символов: АР1 КП. Эти символы посылает в линию связи та сторона, по инициативе которой происходит разъединение.

Абонентский пункт АП-3 (ЕС-8503) (рис. 15.3) предназначен для обмена данными с ЭВМ Единой системы или другими АП-3 по коммутируемым и некоммутируемым телефонным каналам связи со скоростью 600—1200 бит/с. Он является среднескоростным пунктом, осуществляющим синхронную передачу или прием в полудуплексном режиме.

Как и в предыдущих пунктах, в АП-3 предусмотрен автономный режим работы. В этом режиме информация, вводимая через считывающее устройство, может быть автоматически отпечатана с помощью пишущей машинки. Можно также наносить информацию на перфоленты, либо посредством набора данных на клавиатуре пишущей машинки, либо путем дублирования с уже имеющихся перфолент.

Абонентский пункт АП-4 (ЕС-8504) предназначен для работы с ЭВМ в двухпунктовых звеньях данных при связи через некоммутируемые телефонные каналы. Скорость обмена данными может достигать 2400 бит/с.

Помимо устройств ввода-вывода перфоленточного типа, устройства ввода с перфокарт и печатающей машинки (рис. 15.3) АП-4 содержит шаговый накопитель на магнитной ленте. УУС пункта может управлять восемью различными устройствами ввода-вывода, удаленными от него на расстояние до 500 м.

Как и предыдущие пункты, АП-4 обеспечивает автоматические режимы обмена данными с ЭВМ, а также перенос информации с одного носителя на другой в автономном режиме.

Абонентский пункт АП-5 (ЕС-8505) предназначен для осуществления сбора данных в составе АСУП. Содержит специальные

счетные регистрирующие устройства, автоматически собирающие информацию о текущей производственной обстановке.

Абонентский пункт АП-11 (ЕС-8511) предназначен для обмена данными с ЭВМ ЕС или с аналогичным пунктом по коммутируемым и некоммутируемым телефонным каналам связи. Пункт рассчитан на работу в полудуплексном режиме. Возможные скорости обмена данными через указанные каналы могут составлять 600, 1200 и 2400 бит/с.

Состав АП-11 представлен на рис. 15.3. В зависимости от требований абонента состав пункта по количеству устройств ввода-вывода может быть различным. В комплект входит от одного до трех устройств в различных сочетаниях.

Совместно с ЭВМ пункты типа АП-11 могут работать в составе как двухпунктовых звеньев данных, так и в многопунктовых. Самостоятельно абонентские пункты АП-11 можно соединять только по двухпунктовой схеме. В последнем случае, работая в режиме соперничества, оба пункта равноправны в осуществлении управления передачей данных. Поэтому могут возникать конфликтные ситуации при одновременных запросах. Для исключения этого один из пунктов назначается главным, а второй — второстепенным. Главному пункту предоставляется возможность посылать запросы на разрешение вести передачу периодически с интервалами (таймаутами) в одну секунду. Второстепенным пунктам интервалы между возможными посылками запросов устанавливаются равными трем секундам. Запросы как те, так и другие осуществляются посылкой сигналов КТМ («Кто там?»).

Пункт АП-11 может входить в связь и вести прием или передачу автоматически, без участия оператора. Автоматическая работа осуществляется по сигналам (управляющим символам), поступающим из канала связи.

Абонентский пункт АП-61 (ЕС-8561) предназначается для работы с ЭВМ главным образом в режиме «диалога». Для этого они оборудуются специальными пультами с функциональными и алфавитно-цифровыми клавиатурами (рис. 15.3), связанными с устройствами отображения информации на экранах электронно-лучевых трубок, и пишущими машинками.

Вводимая с клавиатуры пюльта информация отображается на экране стандартной ЭЛТ размером 35 см по диагонали. После визуального контроля, исправления и редактирования она передается для обработки в ЭВМ.

Обработанная информация в зависимости от назначения выводится либо на экран ЭЛТ, либо печатается с помощью пишущей машинки.

Одновременно на экране ЭЛТ можно отображать до 960 знаков, размещаемых в 12 строках. Ввод в ЭВМ с экрана и вывод на экран осуществляются либо строками, в которых может находиться до 80 знаков, либо целыми сообщениями длиной до 960 знаков. Передаваемые данные кодируются стандартным 7-эле-

ментным кодом КОИ-7. Обмен данными АП-61 с ЭВМ осуществляется по телефонным некоммутируемым каналам со скоростью 1200—2400 бит/с.

Абонентские пункты АП-61 могут работать с ЭВМ в двухпунктовых и многопунктовых звеньях данных. В любом из них при установлении соединений абонентские пункты подчиняются ЭВМ.

Абонентский пункт АП-63 (ЕС-8563) аналогично пунктам АП-61 в основном предназначаются для обмена данными с ЭВМ в режиме «диалога». Эти пункты также комплектуются устройствами визуального отображения информации на экранах ЭЛТ, пультами с алфавитно-цифровой клавиатурой и пишущими машинками (рис. 15.3). Отличие состоит в том, что имеющееся в составе аппаратуры АП-63 устройство группового управления способно обеспечивать одновременную работу от 8 до 16 индикаторов на ЭЛТ. Индикаторы могут быть удалены от устройства группового управления на расстояние до 500 м. Пишущая машинка располагается в непосредственной близости от УУС.

Если индикаторов на ЭЛТ не более восьми, то одновременно на экране каждого из них может отображаться до 480 алфавитно-цифровых знаков, расположенных в 12 строках по 40 знаков. При количестве индикаторов, большем восьми, одновременно на каждом экране может быть не более 240 знаков, располагаемых в шести строках.

Обмен данными с ЭВМ осуществляется блоками либо длиной до 40 знаков (строками), либо сообщениями, вмещающими кадр в 480 или 240 знаков. Так же как и АП-61, абонентские пункты АП-63 могут работать с ЭВМ через любой из мультиплексоров, входящих в номенклатуру оборудования Единой системы. Для связи применяются некоммутируемые телефонные каналы, обеспечивающие обмен информацией со скоростями 1200—2400 бит/с.

Абонентский пункт АП-70 (ЕС-8570) предназначается для использования в двухпунктовых и многопунктовых звеньях данных типа АП-ЭВМ или АП-АП. Пункты АП-70 могут работать как в режиме подчинения, так и соперничества.

По составу оборудования (рис. 15.3) — это самый дешевый пункт. При отсутствии необходимости в передаче или приеме данных машинка типа «Консул 260» может использоваться автономно, как обычная пишущая машинка.

В режимах приема и передачи данные вводятся с клавиатуры, а выводятся — путем печати на рулонную бумагу. Обмен данными, кодируемыми стандартным 7-элементным кодом КОИ-7, осуществляется со скоростью, не превышающей 100 бит/с. В качестве линий связи могут использоваться некоммутируемые телеграфные, а также некоммутируемые и коммутируемые телефонные каналы. По скорости АП-70 совместим с пунктами типа АП-1. Данные передаются блоками переменной длины. Максимальная длина блока ограничивается длиной печатной строки.

Простейшие абонентские пункты могут строиться также на основе стандартных рулонных телеграфных аппаратов, связываемых с МПД ЭВМ телеграфными линиями связи. Использование телеграфных аппаратов аналогично пунктам АП-70.

По состоянию аппаратуры всех рассмотренных выше абонентских пунктов различают следующие режимы работы:

- «телефон»;
- передача данных;
- автономный.

В режиме «телефон» аппаратура передачи данных абонентского пункта обеспечивает операции набора номера вызываемого абонента, установления соединения, приема сигналов «Занято», «Отбой» при связи по коммутируемым каналам, а также вызова оператора в случае использования некоммутируемых каналов связи.

Установление соединений со стороны абонентского пункта производится с помощью стандартных вызывных устройств. Ими могут быть переговорно-вызывные устройства (ПВУ), обычно применяемые при некоммутируемых каналах связи. Такое устройство, например, входит в состав аппаратуры АП-3. На коммутируемых телефонных каналах связи для установления соединений и ведения служебных переговоров используются стандартные телефонные аппараты. При работе по коммутируемым телефонным каналам связи аппаратура АП обеспечивает процедуру идентификации (опознавания номера при входящем соединении).

В режиме передачи данных осуществляется выдача информации в каналы связи или ее прием и вывод на перфоленты, перфокарты, магнитные ленты или на печать. В этом режиме абонентские пункты могут работать как с оператором, так и без него.

В автономном режиме аппаратура абонентских пунктов используется для подготовки перфолент и перфокарт, для распечатки данных, принятых на различные перфорационные носители, для дублирования перфолент и перфокарт и т. п.

Абонентские пункты рассчитаны на круглосуточную работу.

15.4. МУЛЬТИПЛЕКСОРЫ ПЕРЕДАЧИ ДАННЫХ

Потребность в одновременном подключении к современным ЭВМ многих устройств ввода-вывода, а также и процессоров других машин сделала необходимым применение стандартной системы сопряжения (стандартного интерфейса) периферийных устройств с центральным оборудованием через каналы.

Обслуживая внешние устройства, каналы осуществляют их оперативное подключение к магистралям ввода или вывода информации из ОП машины. Одновременно решаются такие задачи, как управление внешними устройствами, контроль процессов обмена информацией, размещение ее в нужные области памяти или выборка из определенных областей и т. п. Разумеется, что

при таком управлении происходит временное уплотнение информации во внутренних кодовых магистралях машины.

Телефонные и телеграфные каналы являются более медленно действующими цепями обмена информацией по сравнению с линиями связи устройств ввода-вывода, непосредственно подключаемыми к машине. По Тлф и Тлг линиям связи данные передаются последовательно, причем информационные посылки перемежаются со служебными (адресными, управляющими и т. п.). Поэтому, помимо временного уплотнения информации, принимаемой по Тлф и Тлг линиям связи, возникает задача формирования (сборки) из последовательностей посылок переменного или постоянного тока кодовых комбинаций, например, в виде 5-разрядных, 7-разрядных и т. п. Разумеется, что при передачах в линии связи должна решаться обратная задача.

Для решения таких задач применяются специальные многоканальные коммутаторы, получившие название «мультиплексоры передачи данных» (МПД). Обычно они выполняются в виде отдельных, конструктивно обособленных устройств.

По отношению к ЭВМ мультиплексоры передачи данных фактически представляют собой устройства управления удаленными на большое расстояние абонентскими пунктами. Через МПД также осуществляется связь с удаленными процессорами.

С ЭВМ мультиплексоры передачи данных, как и другие устройства управления внешними устройствами, соединяются через мультиплексные каналы, к которым они подключаются посредством стандартного интерфейса ввода-вывода (см. рис. 15.1). В зависимости от количества и типа обслуживаемых линий связи могут применяться различные МПД. В составе ЕС ЭВМ предусмотрены три типа мультиплексоров: МПД-1А, МПД-2 и МПД-3, технические параметры и возможности применения которых указаны в табл. 15.2.

Мультиплексор передачи данных МПД-1А (ЕС-8400) предназначается для работы с абонентскими пунктами АП-1, АП-70, АР-61, АП-63 и телеграфными аппаратами. Он обеспечивает обмен данными в полудуплексном режиме по Тлг и Тлф линиям связи в соответствии с алгоритмами функционирования перечисленных абонентских пунктов. Имея 15 выходов, мультиплексор может одновременно обслуживать 15 абонентов.

Мультиплексор обеспечивает выполнение следующих основных операций:

- установление соединений с абонентами;
- прием (передача) данных по линиям связи;
- адаптивное управление информацией, получаемой из линии связи или из мультиплексного канала;
- ввод (вывод) данных в мультиплексный канал.

Структурная схема МПД-1А показана на рис. 15.4. В состав мультиплексора входят:

- устройство сопряжения с мультиплексным каналом (УСК);

Технические параметры и возможности применения МПД ЕС ЭВМ

Тип МПД	Диапазон скоростей передачи данных, бит/с	Количество обслуживаемых каналов связи в режимах передачи		Используемые линии связи	Обслуживаемые абонентские пункты	МПД, с которыми возможен обмен данными	Тип звеньев данных, включающих абонентские пункты	АПД, возможная для использования в линиях связи	Модели ЭВМ, совместно с которыми преимущественно используются МПД
		полудуплексных	дуплексных						
МПД-1А ЕС-8400	50—2400	15	—	Тлг коммутируемые, Тлг некоммутируемые, Тлф некоммутируемые	АП-1, АП-61, АП-63, АП-70, Тлг А	—	Двухпунктовые, многопунктовые	Модем-200, Модем-2400	ЕС-1020, ЕС-1030
МПД-2 ЕС-8402	50—4800	8—176	4—88	Тлг некоммутируемые, Тлф коммутируемые, Тлф некоммутируемые	АП-1, АП-2, АП-3, АП-4, АП-5, АП-11, АП-61, АП-63, АП-70, Тлг А	МПД-2, МПД-3	То же	Модем-200, Модем-1200, Модем-2400, Модем-4800, АБУ Тлф, УЗО-200, УЗО-1200	ЕС-1050, ЕС-1060
МПД-3 ЕС-8403	50—48000	1—4	1—2	Тлг некоммутируемые, Тлф коммутируемые, Тлф некоммутируемые, широкополосные	АП-1, АП-3, АП-11, АП-70, Тлг А	МПД-2, МПД-3	То же	Модем-200, Модем-1200, Модем-2400, Модем-4800, Модем-48000, АБУ Тлф, УЗО-1200	ЕС-1030, ЕС-1040, (ЕС-1050, ЕС-1060)

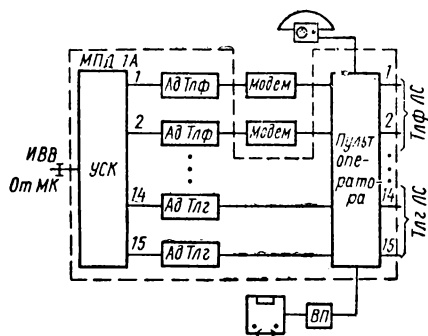


Рис. 15.4. Структурная схема МПД-1А

— 15 адаптеров, обеспечивающих работу по телеграфным и телефонным линиям связи;
— пульт оператора.

Конструктивно МПД выполняются в виде стойки, где размещается перечисленное оборудование, включая пульт оператора, с помощью которого устанавливаются соединения по выделенным телефонным каналам и коммутируемым телеграфным линиям связи.

Устройство сопряжения с каналом — это своего рода коммутатор, обслуживающий все подключаемые к МПД линии связи. Адаптеры (Ад) управляют обменом информацией с абонентскими пунктами. С их помощью осуществляется сопряжение либо непосредственно с линиями связи (телеграфными), либо с аппаратурой передачи данных (с УЗО или модемами при связи по телефонным каналам). На них также возлагается задача повышения достоверности передаваемых и принимаемых данных.

Устройство адаптера зависит от того, для какой связи он предназначен. В МПД-1А применяются адаптеры трех типов: Ад-1, Ад-2 и Ад-3.

Адаптеры Ад-1 предназначены для управления обменом данными с телеайпами по коммутируемым телеграфным линиям связи, входящими в состав сети абонентского телеграфа. При установлении связи по этим линиям набор номера абонента или выдача ответа осуществляется вручную с пульта оператора.

Управление обменом данными через некоммутируемые телеграфные каналы с АП-1 и АП-70 производится адаптерами типа Ад-2.

Телефонные адаптеры Ад-3 используются при работе с абонентскими пунктами АП-61 и АП-63. Через МПД-1А связь с этими АП возможна по некоммутируемому телефонным линиям.

Для обмена данными через УСК подключаются к интерфейсу мультимплексного канала. УСК передает от канала к адаптеру команды, определяемые программой машины, а в обратном направлении — байты состояния и уточненного состояния (Ад и АПД). Само устройство сопряжения с каналом не хранит ни команды, ни байты состояния. Выбранному адаптеру УСК выдает сигнал принять команду, следующую по шинам канала ШИН-К. Аналогично по подключенным шинам абонента ШИН-А через УСК передается информация о состоянии адаптера и аппаратуры передачи данных. Это осуществляется по сигналу — запросу ВЫДАТЬ СОСТОЯНИЕ, посылаемому адаптеру по шине УПР-А.

Основные команды ввода-вывода мультимплексора

Название команды	Код команды на ШИН-К	Действия, выполняемые по команде
	К 0123 4567	
ЧТЕНИЕ	0 0000 0010	Происходит прием данных из линии связи, формирование информационных байт и передача их в мультимплексный канал по ШИН-А
ЗАПИСЬ	0 0000 0001	Происходит передача данных, поступающих из мультимплексного канала по ШИН-К, в линию связи
ПРОВЕРИТЬ ВВОД-ВЫВОД	1 0000 0000	Байт состояния, хранящийся в регистре состояния адаптера, передается в мультимплексный канал по ШИН-А. Байт уточненного состояния из регистра уточненного состояния адаптера передается в мультимплексный канал по ШИН-А
УТОЧНИТЬ СОСТОЯНИЕ	0 0000 0100	Эта команда выдается каналом для получения более подробной информации о состоянии адаптера и аппаратуры передачи данных

В аппаратуре МПД-1А предусмотрена возможность выполнения 11 команд, четыре из которых приведены в табл. 15.3. Остальные команды: ПОДГОТОВКА, НАБОР НОМЕРА, ДИАГНОСТИЧЕСКАЯ ЗАПИСЬ, ДИАГНОСТИЧЕСКОЕ ЧТЕНИЕ, ХОЛОСТОЙ ХОД, ОСВОБОЖДЕНИЕ и РЕЗЕРВИРОВАНИЕ — в данной книге не рассматриваются.

Содержание байт состояния и уточненного состояния дано в табл. 15.4 и 15.5.

Т а б л и ц а 15.4

Байт состояния

Номер бита байта	Значение единичного состояния бита	Примечание
0	Внимание	Устанавливается, когда от канала поступила новая команда, а Ад еще выполняет предыдущую
1	Модификатор состояния	
2	Конец работы УВУ	
3	Занято	
4	Конец работы канала	Устанавливается при возникновении необычной ситуации, сопровождающейся указателем в байте уточненного состояния
5	Конец работы устройства	
6	Сбой в устройстве	
7	Особый случай	Устанавливается, когда из линии связи принят символ «Конец передачи» или отрицательный ответ на опрашивание

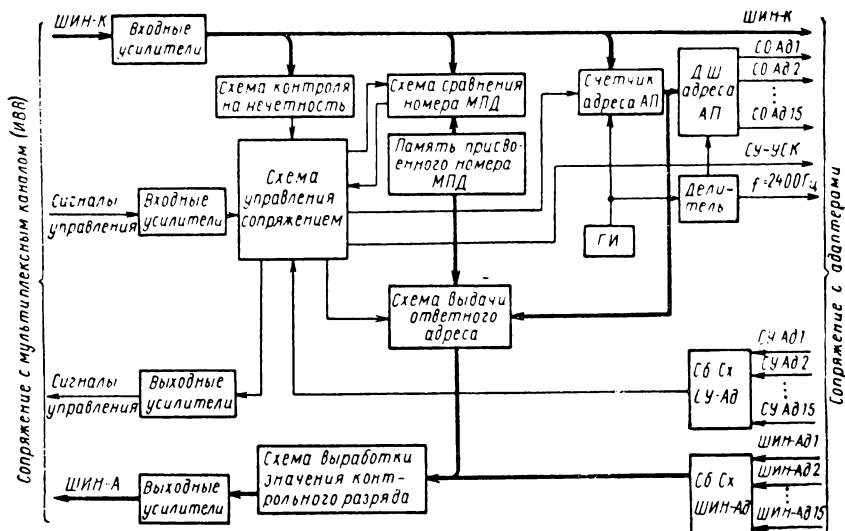


Рис. 15.5. Структурная схема устройства сопряжения с каналом

Таблица 15.5

Байт уточненного состояния

Номер бита байта	Значение единичного состояния бита	Примечание
0	Команда отвергнута	Устанавливается, когда код команды либо неприемлем для данного Ад, либо — четный
1	Требуется вмешательство	1) АПД либо выключена, либо не готова к работе 2) Аппаратурой АП обнаружена ошибка в блоке данных, переданных в МПД
2	Ошибка на ШИН-К	В байте данных или в команде обнаружена ошибка при контроле на нечетность
3	Сбой в оборудовании	Аппаратурой АП обнаружена ошибка в блоке данных, переданных в МПД
4	Ошибка в данных	Устанавливается при обнаружении ошибки в данных, принятых из линии связи
5	Переполнение	Буферный регистр Ад не освободился к моменту окончания формирования следующего байта данных, поступившего из линии связи
6	Потеря данных	Во время выполнения команды ЧТЕНИЕ от канала получен сигнал останова
7	Перерыв	В течение 3 с не установилась синхронизация при выполнении команды ЧТЕНИЕ

Устройство сопряжения с каналом (УСК), структурная схема которого представлена на рис. 15.5, содержит схемы входных и выходных усилителей сигналов, принимаемых из канала и выдаваемых в канал, схемы контроля информации, анализа и формирования адресов, последовательного опроса адаптеров и управления. Информация, поступающая побайтно из канала по линиям ШИН-К, в УСК проверяется на нечетность.

Байтам информации, передаваемым в канал, присваивается контрольный разряд (9-й), приводящий количество единиц в 9 разрядах к нечетному числу.

Информационные коды и сигналы управления поступают от адаптеров в УСК через собирательные схемы соответственно Сб. Сх. ШИН-Ад и Сб.Сх.СУ-Ад. Через схему выработки контрольного разряда и выходные усилители сигналы с выхода Сб.Сх.ШИН-Ад поступают в линии ШИН-А стандартного сопряжения с мультиплексным каналом*.

Рассмотрим работу аппаратуры УСК в режимах:

- начальной выборки;
- обмена данными;
- выдачи состояния.

Во время начальной выборки команда из канала передается адресуемому адаптеру. Перед командой по ШИН-К следует адрес адаптера, сопровождаемый идентифицирующим сигналом-признаком АДР-К по шине идентификации интерфейса. Получив этот сигнал, схема управления сопряжением с помощью схемы сравнения номера сравнивает часть адреса АП, содержащую номер мультиплексора, с номером, присвоенным данному МПД и закомутированным в нем в схеме памяти.

Если номера совпадают, то поступающий следом по шине управления интерфейса сигнал ВБР-К воспринимается схемой управления сопряжением. В ответ она выдает каналу управляющий сигнал РАБ-А, свидетельствующий о том, что МПД подключился к каналу. Если же номера не совпадают, то сигнал ВБР-К пропускается к следующему МПД, подключенному к мультиплексному каналу.

При совпадении номера по сигналу АДР-К адрес абонента заносится в счетчик адреса УСК. Расшифровав адрес, дешифратор адреса АП выдает сигнал опроса СО соответствующему адаптеру. В результате выбранный адаптер подключается к мультиплексному каналу.

В ответ на сигнал РАБ-А канал снимает сигнал АДР-К. Вслед за этим по сигналу управления СУ из адаптера на ШИН-А схемой выдачи ответного адреса выдается байт адреса, содержащего номер МПД и адрес АП. Байт адреса сопровождается сигналом—признаком АДР-А.

* Расположение информационных, управляющих и идентифицирующих сигналов относительно друг друга то же, что и в интерфейсе ввода-вывода (см. гл. 10).

Получив ответ и удостоверившись в правильности адреса, канал по ШИН-К посылает код команды и признак УПР-К. По этому признаку схема управления сопряжением формирует управляющий сигнал стробирования команды, поступающий по цепи СУ УСК в выбранный адаптер. В адаптере команда дешифрируется и запоминается в виде единичного состояния одного из триггеров регистра команд.

После окончания действия сигнала УПР-К устройство сопряжения сигналом УПР-А запрашивает байт состояния адаптера и выдает его по ШИН-А каналу. Эта выдача сопровождается также сигналом УПР-А.

Приняв и проанализировав байт состояния, в случае возможности дальнейшей работы сигналом ИНФ-К сбрасывается сигнал РАБ-А и адаптер начинает выполнение команды.

При обмене данными в процессе выполнения команды ЧТЕНИЕ или ЗАПИСЬ, ожидая сигнал опроса СО от УСК, адаптер выдает потенциальный сигнал запроса на обслуживание—ТРЕБОВАНИЕ ДАННЫХ (ТРБ-Д). Этот запрос вырабатывается, когда адаптер оказывается в состоянии передать каналу или получить от него очередной байт данных.

Поочередный опрос адаптеров осуществляется счетчиком адреса АП. На вход счетчика с генератора поступают импульсы, следующие с частотой 153,6 кГц. Как только счетчик досчитывает до числа, соответствующего адресу адаптера, выдающего запрос на обслуживание, на выходе дешифратора адреса АП формируется сигнал опроса СО Ад этого адаптера. Таким образом, адаптер подключается к УСК, которое теперь оказывается в состоянии воспринять от него сигнал ТРБ-Д. По этому сигналу устройство сопряжения выдает в мультиплексный канал сигнал ТРЕБОВАНИЕ АБОНЕНТА (ТРБ-А). С приходом в ответ сигнала ВБР-К УСК подсоединяется к каналу, посылая ему сигналы РАБ-А и АДР-А.

Сигнал—признак АДР-А разрешает и сопровождает выдачу каналу со счетчика адреса АП адреса подсоединенного адаптера. Приняв и опознав адрес, канал сигналом УПР-К дает разрешение продолжить операцию обмена данными с подключенным в данный момент абонентом. В ответ на это УСК посылает каналу идентификатор ИНФ-А. При этом сбрасываются сигналы АДР-А и УПР-К.

По идентифицирующему сигналу ИНФ-А адаптер либо передает каналу по ШИН-А очередной байт данных (при выполнении команды ЧТЕНИЕ), либо ожидает и принимает по ШИН-К байт от канала (при команде ЗАПИСЬ). В том и другом случаях, приняв или выдав байт данных, канал формирует на выходе сигнал ИНФ-К, по которому сбрасываются сигналы ИНФ-А, РАБ-А и адаптер отключается от мультиплексного канала. Адрес в счетчике АП по импульсу ГИ изменяется и осуществляется переход к опросу следующего адаптера.

По окончании операции ввода-вывода через какой-либо адаптер, как и в ответ на полученный код команды в процессе начальной выборки, происходит автоматическая выд а ч а к а н а л у б а й т а с о с т о я н и я. Если при начальной выборке УСК передает байт состояния, сопровождая его сигналом УПР-А, независимо от сигналов запроса на обслуживание от адаптера, то в конце операции обмена данными такой сигнал формируется адаптером.

Вслед за последним байтом передаваемых данных в байте состояния, находящемся в адаптере, заносятся единицы в разряды, соответствующие признакам: КОНЕЦ РАБОТЫ КАНАЛА и КОНЕЦ РАБОТЫ УСТРОЙСТВА. Одновременно выдается сигнал ТРЕБОВАНИЕ ВЫДАЧИ СОСТОЯНИЯ (ТРБ-С).

Устройство сопряжения с каналом реагирует на это требование, подключает адаптер к каналу и выдает состояния в основном так же, как очередной байт данных при команде ЧТЕНИЕ. Только вместо признака ИНФ-А в ответ на сигнал УПР-К от канала устройство сопряжения УСК выдает запрос УПР-А. Этот сигнал-запрос вызывает байт состояния и по ШИН-А передает его в канал.

Канал может воспринять или не воспринять байт состояния. Байт состояния, содержащий нули во всех разрядах, всегда воспринимается каналом. При наличии в некоторых разрядах единиц канал может не воспринять байт состояния. В этих случаях он должен продолжать храниться в регистре состояния адаптера.

Если канал воспринимает байт состояния, то в ответ выдается сигнал ИНФ-К, если нет — то снова УПР-К. В обоих случаях сбрасывается сигнал УПР-А и опрашиваемый адаптер отключается от УСК. Однако, если в первом случае адаптер, закончив операцию ввода-вывода, переходит в режим ожидания следующей команды от канала, то во втором — адаптер продолжает хранить байт состояния и выдавать сигнал ТРБ-С, периодически повторяя попытки передать информацию о состоянии в канал. Эти попытки повторяются до тех пор, пока канал не примет байт состояния и не разрешит отключение адаптера.

Канал может временно запрещать повторные попытки передачи состояния, выдавая по шине БЛК-К сигнал БЛОКИРОВКА. При снятии блокировки попытки передачи возобновляются.

Условия окончания операции могут быть определены каналом до останова адаптера. В этом случае отключение последнего происходит по инициативе канала путем послылки сигнала УПР-К, воспринимаемого как ОСТАНОВ. В любом случае окончания работы и отключения адаптера его регистр состояния сбрасывается и снимается сигнал РАБ-А.

Байт состояния, как и байт уточненного состояния, может выдаваться по специальным командам, одной из которых является команда ПРОВЕРИТЬ ВВОД-ВЫВОД (табл. 15.3). При исполнении таких команд процедуры начальной выборки адаптера и обмена данными выполняются аналогично рассмотренному выше.

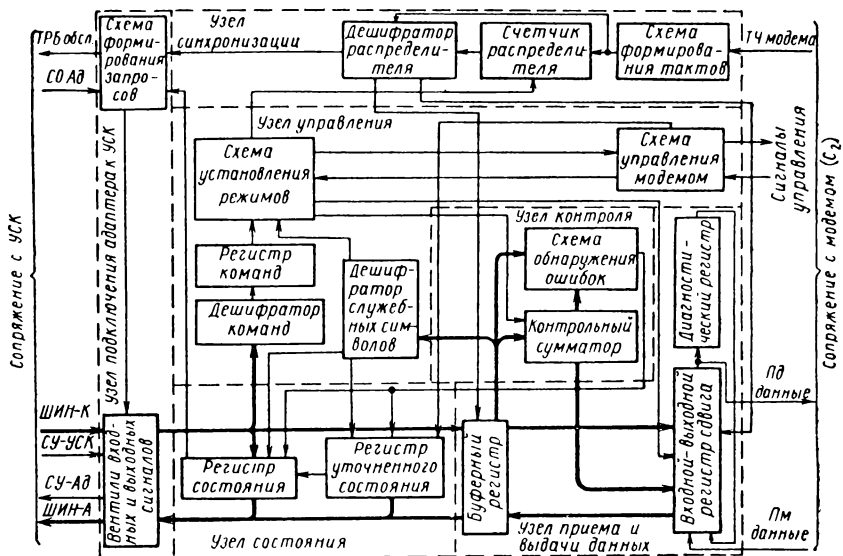


Рис. 15.6. Структурная схема телефонного адаптера

Адаптеры. В качестве примера рассмотрим телефонный адаптер Ад-3, структурная схема которого представлена на рис. 15.6. Адаптеры этого типа позволяют передавать данные по выделенным телефонным каналам со скоростями 600—2400 бит/с. Одним из основных назначений Ад-3 является управление обменом данными между ЭВМ и устройствами отображения информации на экранах ЭЛТ. Обмен осуществляется в режиме подчинения абонентских пунктов мультиплексу.

Структурная схема на рис. 15.6 содержит узлы:

- подключения адаптера к УСК;
- приема и выдачи данных;
- состояния;
- контроля;
- управления;
- синхронизации.

В узле подключения находятся триггеры запроса обслуживания, выдающие сигналы ТРБ-Д или ТРБ-С, назначение которых было рассмотрено выше. Сигнал требования данных ТРБ-Д формируется, когда в режиме передачи данных в ЛС буферный регистр узла приема и выдачи освобождается, а в режиме приема из ЛС—заполняется. По сигналу опроса адаптера СО Ад, поступающему из УСК, открываются вентили входных или выходных сигналов, тем самым разрешая прохождение информации или от УСК, или обратно.

Во входном-выходном регистре сдвига узла приема и выдачи данных информация, идущая к модему, преобразуется из парал-

лельного кода в последовательный, а в обратном направлении — из последовательного в параллельный. Буферный регистр обеспечивает такой темп передачи байт данных в сдвигающий регистр, чтобы поток информации в ЛС в режиме записи был непрерывным и согласованным с тактовой частотой синхронизации, задаваемой модемом.

Адаптер Ад-3 соединяется с модемом через стык, содержащий цепи (табл. 15.6): данных, управления, синхронизации и заземления.

Т а б л и ц а 15.6

Цепи стыка с модемом

Наименование цепи	Номер цепи	Направление	
		от модема	к модему
Защитное заземление	101	+	+
Сигнальное заземление	102	+	+
Передаваемые данные	103	—	+
Принимаемые данные	104	+	—
Запрос передачи	105	—	+
Готов к передаче	106	+	—
АДП готова	107	+	—
Присоединить к линии	108	—	+
Детектор сигнала, принимаемого из линии	109	+	—
Выбор сигнальной скорости	111	—	+
Тактовая частота (ТЧ) передачи	114	+	—
Тактовая частота приема	115	+	—
Данные, передаваемые в обратный канал	118	—	+
Данные, принимаемые из обратного канала	119	+	—
Детектор сигнала, принимаемого из обратного канала	122	+	—
Дистанционное включение	—	+	—
Дистанционное включение	—	—	+

Несмотря на то, что связь через данный стык дуплексная, адаптер производит обмен данными в полудуплексном режиме, т. е. одновременно передача осуществляется только в одном направлении.

В узле состояния формируется информация о состоянии адаптера, модема, АП. Эта информация (табл. 15.4 и 15.5) фиксируется в триггерах регистров состояния и уточненного состояния. Согласно процедурам обмена, рассмотренным, например, при описании УСК, или по специальным командам она выдается мультиплексному каналу.

Контроль принимаемой из ЛС информации, а также формирование контрольных байт передаваемых в ЛС блоков данных осуществляется в узле контроля. Основу его составляют восемь

параллельно работающих сумматоров накапливающего типа и схема обнаружения ошибок, содержащая схему анализа сумматора на нуль и схему контроля на четность байт, поступающих из ЛС в буферный регистр.

Количество единиц в каждом принимаемом последовательно байте должно быть четным. Кроме того, для реализации матричного метода контроля производится подсчет единиц в одноименных разрядах всех байт каждого проходящего через адаптер блока данных. Счет и накопление осуществляются в сумматорах.

При передаче в ЛС полученный в результате указанного подсчета контрольный байт сопровождает блок данных (передается последовательно за символом КТ — «Конец текста»). Когда же идет прием из ЛС, к полученному в результате подсчета в сумматорах контрольному байту поразрядно по модулю два подсуммируется следующий за блоком данных контрольный байт. При правильном приеме во всех разрядах должны получиться нули, что проверяется схемой анализа сумматоров на нуль. Если при проверке на четность байт или при контроле правильности приема блоков данных обнаруживаются ошибки, то в узле контроля вырабатываются соответствующие сигналы, фиксируемые в узле состояния.

Работой узла контроля, как и других узлов адаптера, управляет узел управления. Установление режимов работы и управление происходит по командам, принимаемым через УСК из мультиплексного канала, а также по управляющим символам, сопровождающим передаваемую информацию от канала или из ЛС. Управляющие символы предназначаются для управления связью. Они кодируются согласно таблице стандартного семиэлементного кода КОИ-7.

По стробирующему сигналу ПРИНЯТЬ КОМАНДУ, сопровождающему команду из УСК, она дешифрируется дешифратором команд. Сигналом с соответствующего выхода дешифратора возбуждается один из триггеров режима регистра команд. Если, например, дешифрируются команды ЧТЕНИЕ или ЗАПИСЬ, то соответственно устанавливаются в единичное состояние триггеры, выдающие сигналы: ПРИЕМ (Пм) или ПЕРЕДАЧА (Пд).

Переход адаптера, как и всего звена данных, из состояния управления, при котором возможно выполнение процедуры установления соединения, в режим передачи данных сопровождается установкой в узле управления в единичное состояние триггера, выдающего сигнал ТЕКСТ. Это происходит при дешифрировании управляющего символа «Начало текста». Триггер сбрасывается с появлением на входах дешифратора кодовых комбинаций, соответствующих управляющим символам: «Конец передачи» или «Начало заголовка».

На узел управления также возлагаются функции установления символической синхронизации (синхронизации приема-передачи байт). При каждом изменении направления передачи ввод

в синхронизм индицируется установкой в единичное состояние триггера, вырабатывающего сигнал **ФАЗИРОВАНИЕ**.

Работа всех схем адаптера синхронизируется опорными импульсами, вырабатываемыми узлом синхронизации. Задающая тактовая частота (ТЧ) поступает из модема. По ней формируются четыре серии тактовых импульсов, сдвинутых относительно друг друга. В распределителе, состоящем из счетчика и дешифратора, вырабатываются циклические серии по восемь импульсов. Эти импульсы, поступая, например, в цепь управления регистром сдвига, синхронизируют прием или передачу каждого бита информации.

В целом адаптер представляет собой совокупность аппаратных средств, обеспечивающих реализацию алгоритма обмена информацией, определяемого программой ЭВМ. Главным звеном при обмене является мультиплексор, поочередно опрашивающий подчиненные абонентские пункты. Опрос осуществляется посылкой команды и адреса АП. Для выяснения готовности пункта к работе по команде ему вместе с адресом выдается сигнал приглашения к записи (ПРЗ) или к чтению (ПРЧ). Посылая ответный подтверждающий сигнал (ДА), абонентский пункт переходит в режим подчинения. Если по каким-либо причинам переход в этот режим невозможен (пункт не имеет данных для передачи или не готов к приему), то посылается отрицательный ответ (НЕТ). В этом случае УСК осуществляет переход к опросу следующего абонентского пункта.

Таким образом, для адаптера характерными являются два режима:

- прием команды;
- выполнение команды.

В процессе начальной выборки, когда звено данных находится в состоянии управления, УСК передает адаптеру команду согласно рассмотренной выше процедуре. Если команда приемлема, то в результате дешифрирования устанавливается в единичное состояние один из триггеров регистра команд. В случаях, когда команда оказывается неприемлемой, в регистр уточненного состояния заносится признак **КОМАНДА ОТВЕРГНУТА**, а в регистр состояния — **СБОЙ В УСТРОЙСТВЕ**.

Содержимое регистра состояния в конце процедуры начальной выборки по сигналу УПР-А посылается в мультиплексный канал. «Удостоверившись» в возможности дальнейшей работы, канал посылает адаптеру сигнал УПР-К, сбрасывающий в нулевое состояние регистр состояния и разрешающий выполнение команды.

В процессе обмена данными по командам **ЗАПИСЬ** или **ЧТЕНИЕ** сообщения, передаваемые по телефонным линиям связи, сопровождаются управляющими символами синхронизации СИН, начала текста НТ, конца текста КТ и замыкаются контрольной последовательностью текста КПТ. В результате сообщение, обра-

зующее блок данных, в общем случае имеет следующий формат СИН СИН НТ < текст > КТ КПТ. В рассматриваемом адаптере контрольная последовательность КПТ — контрольный байт, формируется по матричному способу контроля.

Выполнение команды ЧТЕНИЕ начинается с поиска в информационных сигналах, поступающих из линии связи, кодовой комбинации символа СИН. Для этого после дешифрирования команды в узле управления устанавливаются в единичное состояние триггеры, выдающие сигналы ПРИЕМ (Пм) и ФАЗИРОВАНИЕ, запускается специальный счетчик времени вхождения в синхронизм и блокируется работа распределителя импульсов узла синхронизации.

Как только специальный дешифратор обнаруживает во входном сдвигателе кодовую комбинацию СИН, счетчик распределителя импульсов устанавливается в исходное состояние и распределитель начинает работу. Прием во входной сдвигатель второго символа СИН должен закончиться в момент выработки дешифратором распределителя восьмого импульса после начала работы счетчика. Выполнение этого условия означает правильность установления символьной синхронизации и вызывает окончание режима фазирования (сбрасывается в нулевое состояние триггер, выдающий сигнал ФАЗИРОВАНИЕ).

Если в течение 3 с не происходит установления символьной синхронизации, то сигнал переполнения со счетчика времени прерывает выполнение команды ЧТЕНИЕ (сбрасывается в нулевое состояние триггер, выдающий сигнал ПРИЕМ) и в регистр уточненного состояния заносится признак ПЕРЕРЫВ (табл. 15.5).

При правильном установлении символьной синхронизации следующий за вторым символом СИН байт кодовой комбинации переписывается с входного сдвигателя в буферный регистр и через УСК мультиплексному каналу выдается запрос на передачу данных ТРБ-Д. Одновременно эта кодовая комбинация дешифрируется дешифратором служебных символов. При обнаружении символа начала текста НТ устанавливается режим передачи данных («Текст»). Все последующие байты при прохождении через буферный регистр поступают также в сумматор узла контроля.

После обнаружения символа конца текста КТ в адаптере начинается завершение выполнения операции чтения. Следующий за символом КТ контрольный байт КПТ суммируется по модулю два с содержимым сумматора. Если полученная сумма равна нулю, то операция заканчивается занесением только в регистр состояния признаков: КОНЕЦ РАБОТЫ КАНАЛА и КОНЕЦ РАБОТЫ УСТРОЙСТВА. В случае ненулевой суммы в регистр состояния заносится еще признак СБОЙ В УСТРОЙСТВЕ, а в регистр уточненного состояния — ОШИБКА В ДАННЫХ.

В процессе чтения постоянно контролируется состояние и работа аппаратуры передачи данных по цепям 106, 107 и 109 стыка с модемом. Если в любой из этих цепей пропадает сигнал,

то выполнение команды немедленно прекращается и в регистр состояния заносится признак СБОЙ В УСТРОЙСТВЕ, а в регистр уточненного состояния — ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО.

Выполнение команды чтения прерывается, если по линии связи приходит символ КОНЕЦ ПЕРЕДАЧИ. В этом случае адаптер возвращается в режим управления и в регистр состояния заносится признак ОСОБЫЙ СЛУЧАЙ. Аналогичные действия осуществляются в адаптере, когда при установлении соединения в ответ на запрос приходит отрицательный ответ НЕТ. Если отрицательный ответ поступает в адаптер в режиме ТЕКСТ, то выполнение операции прекращается как и в случае обнаружения ошибки в принятых данных. Отрицательный ответ во время передачи данных может быть послан абонентским пунктом в случае обнаружения им ошибки в принятом ранее тексте.

При выполнении команды ЗАПИСЬ режим передачи в ЛС также начинается с процесса фазирования. В результате дешифрирования команды в узле управления устанавливаются в единичное состояние триггеры, выдающие сигналы ПЕРЕДАЧА и ФАЗИРОВАНИЕ, модему посылаются дважды кодовая комбинация СИН, а в УСК — запрос данных ТРБ-Д.

После этого, в следующем цикле обслуживания данного канала, когда вновь опрашивается адаптер, сигналом ИНФ-К в буферный регистр заносится очередной байт данных. Код, содержащийся в байте, анализируется дешифратором служебных символов. При обнаружении символа НАЧАЛО ТЕКСТА (НТ) устанавливается в единичное состояние триггер, выдающий сигнал ТЕКСТ. Затем этот символ переписывается во входной—выходной регистр сдвига, а каналу дается запрос ТРБ-Д на следующий байт данных. Передача байтов в линию связи будет происходить до тех пор, пока от канала не поступит либо символ КОНЕЦ ТЕКСТА (КТ), либо сигнал останова.

В первом случае в узле управления формируется сигнал КОНЕЦ ТЕКСТА, и вслед за символом КТ в линию связи с сумматора узла контроля посылается сформированный в процессе передачи контрольный байт. После этого в узле управления вырабатывается сигнал конца операции. Во втором случае сигнал конца операции вырабатывается сразу после завершения передачи в модем содержимого сдвигающего регистра.

Как и при чтении, в процессе записи постоянно контролируется аппаратура АПД с помощью сигналов, поступающих по управляющим цепям от модема.

Если в режиме передачи происходит задержка выдачи каналом очередных байтов данных, то для сохранения непрерывности следования информации в линии связи адаптер заполняет возникающие интервалы символами СИН.

Процедуры выполнения диагностических команд записи и чтения отличаются от рассмотренных выше лишь тем, что обмен

информацией осуществляется не с линией связи, а со специальным диагностическим регистром, имеющимся в узле приема и выдачи данных.

Как уже указывалось выше, содержимое регистра состояния автоматически выдается мультимплексному каналу после выработки признаков КОНЕЦ РАБОТЫ КАНАЛА и КОНЕЦ РАБОТЫ УСТРОЙСТВА, вырабатывая запрос на передачу состояния ТРБ-С. Содержимое регистра уточненного состояния выдается каналу по специальному запросу от него.

Пульт оператора предназначается:

— для подключения и сопряжения модемов с выделенными телефонными линиями связи, а также телеграфных адаптеров с коммутируемыми телеграфными ЛС;

— для коммутации различных режимов и состояний МПД;

— для вызова абонентов по телефону и установления предварительных соединений вручную;

— для включения и выключения питания МПД;

— для индикации различных состояний.

Мультимплексор передачи данных МПД-2 (ЕС-8402) обеспечивает возможность работы со всеми типами перечисленных в предыдущем параграфе абонентских пунктов, а также с МПД-2 и МПД-3 при обмене информацией между машинами. Кроме устройства сопряжения с каналом и адаптеров, МПД-2 содержит:

— оперативное запоминающее устройство на ферритовых сердечниках, содержащее зону для буферного запоминания до четырех байт принимаемых и передаваемых данных и зоны для записи и хранения управляющей информации всех обслуживаемых направлений;

— устройство управления запоминающим устройством, обеспечивающее функции управления, общие для всех типов АП и МПД;

— специальные устройства управления, осуществляющие реализацию специальных процедур управления, характерных для определенных типов АП;

— устройства предварительного уплотнения для групп направлений, в каждой из которых обмен данными с АП идет с одинаковой скоростью и однотипной символьной синхронизацией.

Для работы по коммутируемым телефонным каналам МПД-2 может сопрягаться с автоматическим вызывным устройством. В МПД-2 обеспечивается возможность сопряжения с двумя мультимплексными каналами с автоматическим поочередным подключением к каждому из них.

В зависимости от количества и типов обслуживаемых пунктов МПД-2 может различно комплектоваться адаптерами и устройствами предварительного уплотнения в пределах максимальных возможностей, указанных в табл. 15.2.

Мультимплексор передачи данных МПД-3 (ЕС-8403) предназначен для обеспечения связи машин с абонентскими пунктами,

содержащими устройства ввода-вывода перфорационного типа и средства печати, такие как печатающие машинки и АЦПУ, а также — с МПД-2 и МПД-3 при организации обмена данными между машинами. Они могут подключаться как к мультиплексным, так и к селекторным каналам ЭВМ.

15.5. АППАРАТУРА ПЕРЕДАЧИ ДАННЫХ

Аппаратура передачи данных предназначается для установления соединений по некоммутируемым и коммутируемым линиям связи, передачи (приема) по ним дискретной информации и при необходимости — повышения ее достоверности.

В зависимости от используемых линий связи и требований к достоверности передачи данных АПД может иметь в своем составе (рис. 15.2):

- устройства преобразования сигналов телеграфного типа (УПС Тлг);
- устройства преобразования сигналов для телефонных каналов (модемы);
- вызывные устройства;
- устройства защиты от ошибок.

Основные типы и краткие технические характеристики ряда устройств, применяемых в АПД системы телеобработки ЕС ЭВМ, приведены в табл. 15.7. Входящие в состав Единой системы модемы делятся на низкоскоростные, среднескоростные и высокоскоростные. К низкоскоростным относится модем 200, предназначенный для связи с абонентскими пунктами АП-1, АП-2 и АП-70 в основном по телефонным коммутируемым каналам общего пользования.

Т а б л и ц а 15.7

Аппаратура передачи данных

Устройство	Шифр	Краткая техническая характеристика
Устройство преобразования сигналов телеграфного типа (УПС Тлг)	ЕС-8040	Максимальная скорость передачи данных — 200 бод Обеспечивает передачу (прием) информации по 2-проводным симметричным и несимметричным цепям
Модем-200	ЕС-8001	Максимальная скорость передачи (приема) данных — 200 бит/с Тлф ЛС — коммутируемые, 2-проводные и некоммутируемые, 2- и 4-проводные Модуляция — частотная По двухпроводным ЛС возможны полудуплексный и дуплексный режимы передачи Используется в ЛС с АП-1, АП-2, АП-70

Устройство	Шифр	Краткая техническая характеристика
Модем-1200	ЕС-8005	Скорость передачи данных — 600, 1200 бит/с Тлф ЛС — коммутируемые, 2-проводные и некоммутируемые, 2- и 4-проводные Модуляция — частотная Используется в ЛС с АП-3
Модем-2400	ЕС-8010	Скорость передачи данных — 2400 бит/с Тлф ЛС — некоммутируемые, 4-проводные Модуляция — фазовая Используется в ЛС с АП-11, АП-61, АП-63 и между машинами
Модем-4800	ЕС-8015	Возможные скорости передачи данных — 2400, 3600 и 4800 бит/с Тлф ЛС — некоммутируемые, 4-проводные Модуляция — фазовая Используется в ЛС между машинами
Модем-48000	ЕС-8019	Режим обмена — дуплексный Скорость передачи данных — 48 000 бит/с Каналы связи — широкополосные, первичной группы 60—108 кГц Модуляция — амплитудная, с одной боковой полосой частот Используется для построения дуплексных каналов высокоскоростного обмена данными между удаленными друг от друга ЭВМ
Устройство защиты от ошибок с автозапросом (УЗО-1200)	ЕС-8121	Скорость передачи данных — 600, 1200 бит/с
Устройство защиты от ошибок с автозапросом (УЗО-4800)	ЕС-8135	Скорость передачи данных — 2400, 4800 бит/с
Устройство защиты от ошибок с автозапросом (УЗО-48000)	ЕС-8140	Скорость передачи данных — 48 000 бит/с
Телефонное вызывное устройство	ЕС-8061	Предназначено для телефонных коммутируемых каналов общего пользования Обеспечивает автоматический набор номера, а также ответ и переключения: «Телефон—данные» и «Данные—телефон»

Среднескоростными являются модемы 1200 и 2400. Они предназначаются для организации связи с абонентскими пунктами, содержащими более быстродействующую аппаратуру ввода-вывода (рис. 15.3). К среднескоростным также относится модем 4800, который, как и высокоскоростной модем 48000, предназначается для организации межмашинных связей. Если первые три типа модемов в основном используются в звеньях данных с полу-

дуплексным режимом обмена информацией, то последние два — в звеньях с дуплексными каналами.

Модем любого типа состоит из передающей и приемной частей. В общем случае передающая часть содержит:

- передатчик прямого канала;
- приемник обратного канала;
- детектор качества сигнала обратного канала;
- блок фильтров, разделяющих прямой и обратный каналы;
- линейный блок, осуществляющий переключение «Телефон—данные»;
- блок управления.

В состав приемной части входят:

- приемник прямого канала;
- детектор качества сигнала прямого канала;
- передатчик обратного канала;
- блок фильтров;
- переключающий линейный блок;
- блок управления.

Модемы, как и УПС Тлг, обеспечивают непосредственную выдачу в линии связи и прием служебной и смысловой информации. Закодированные дискретные данные по линиям связи передаются последовательно. Для удобства соединения с коммутируемыми и некоммутируемыми каналами входные и выходные цепи выполняются в соответствии с требованиями стандарта на стык C_1 (рис. 15.1). Этот стандартом определяются входные и выходные параметры сопряжения с 2- и 4-проводными линиями.

С МПД и УУС абонентских пунктов модемы и УПС Тлг соединяются через многопроводные стандартные стыки типа C_2 . Пример такого стыка представлен в табл. 15.6. В случаях применения устройств защиты от ошибок УЗО, соединения с ними выполняются через 34-проводные стандартные стыки C_3 .

Стык C_3 содержит девять цепей параллельной передачи к УЗО и девять цепей параллельного приема от него байтов данных с контрольными (девятью) битами. Остальные из 34-х цепей используются для управляющих и идентифицирующих сигналов, таких, как УПРАВЛЕНИЕ ПЕРЕДАТЧИКА АПД, УПРАВЛЕНИЕ ИСТОЧНИКА ООД, УПРАВЛЕНИЕ ПОЛУЧАТЕЛЯ ООД, УПРАВЛЕНИЕ ПРИЕМНИКА АПД, ООД ГОТОВО, АПД ГОТОВА и т. п.

Устройства защиты от ошибок (табл. 15.7) применяются для обнаружения и устранения ошибок, возникающих в каналах связи и в аппаратуре передачи данных. Для этого УЗО снабжаются запоминающими устройствами, позволяющими накапливать блоки данных определенной длины. Размеры блоков зависят от качества каналов связи и от скорости обмена данными. При использовании УЗО для помехозащитного кодирования применяются циклические коды, образующие полиномы которых выбираются с учетом параметров распределения ошибок в каналах

связи. Ошибки исправляются путем автоматических повторений передач блоков данных по запросам от принимающих устройств.

Независимо от того, используются ли в звене данных УЗО или нет, для защиты каналов связи от помех принимают самостоятельные меры. К ним относятся: средства контроля качества принимаемых сигналов, включаемые в состав аппаратуры модемов, проверка в МПД и в УУС АП форматов принимаемых сообщений, а также временное ограничение, накладываемое на выполнение отдельных операций. Примером последнего может служить упомянутое в предыдущем параграфе применение счетчика времени для контроля и поддержания в заданных пределах допустимого времени установления символьной синхронизации.

15.6. ФОРМАТЫ СООБЩЕНИЙ ДЛЯ ПОЛУДУПЛЕКСНОГО РЕЖИМА

Аппаратура управления абонентских пунктов и МПД может находиться в состояниях:

- дежурном;
- управления;
- передачи данных.

В дежурном состоянии она следит за каналом связи с целью обнаружения управляющих служебных символов (см. приложение, табл. 8). Если звенья данных функционируют по принципу соперничества, т. е. станции (АП, ЭВМ) равноправны в выполнении функций передачи информации, то одна из них, желающая начать передачу, посылает управляющую последовательность запроса с символом КТМ. Этим действием другая станция выводится из дежурного состояния. При обнаружении символа КТМ она переходит в состояние управления. В режиме подчинения управляющая станция сообщает подчиненным об освобождении (конце передачи) символом КП. В данном случае этот символ служит сигналом для перевода всех станций, находившихся в дежурном состоянии, в состояние управления.

Находясь в состоянии управления, станция следит за каналом связи с целью обнаружения относящихся к ней управляющих последовательностей. Если в режиме соперничества оба звена данных всегда соединены и для разрешения вести передачу достаточно послать лишь символ КТМ, то в многопунктовых звеньях, работающих в режиме подчинения, перед символом КТМ посылается определенная управляющая последовательность, называемая префиксом. Префикс содержит адрес вызываемого абонента, а также дополнительную служебную информацию, например, указывающую, зачем производится вызов. Обращение к станции может происходить либо с целью передачи ей информации (выборка для записи), либо с целью выяснения наличия данных у абонента (опрос для чтения) или просто для запроса номера и информации о состоянии аппаратуры АП (идентификация).

Обнаружив предназначенную ей последовательность, станция посылает положительный или отрицательный ответ (символы ДА или НЕТ). Перед управляющим символом ответа следует префикс, содержащий адрес вызываемой станции. Послав ответ, станция переходит к передаче или готовится к приему сообщения. Если же на вход поступает управляющая информация, относящаяся к другому пункту, то управляющая аппаратура АП возвращается в дежурное состояние.

В полудуплексном режиме при посылках запросов и ответов происходят необходимые переключения в канале связи, вызываемые изменением направления передачи. В синхронных системах связи это сопровождается каждый раз процедурами фазирования с целью установления посимвольной синхронизации. Поэтому например, в режиме подчинения выбор станции для передачи ей информации будет осуществляться посылкой управляющей последовательности

СИН СИН < префикс > КТМ

Если станция работоспособна и готова к приему, то последует положительный ответ

СИН СИН < префикс > ДА

Обмен данными по каналу связи в системе телеобработки начинается после перехода аппаратуры пунктов в состояние передачи. Сообщения могут передаваться либо полными текстами, либо разделенными на блоки определенной или произвольной длины. Начала и концы целых текстов или составляющих их блоков при поблочной передаче указываются управляющими символами.

Пример формата сообщения и работа аппаратуры МПД при его передаче по синхронному каналу были рассмотрены выше, в п. 15.4. Здесь следует лишь добавить, что в передаваемом тексте может выделяться заголовок либо как отдельный блок

СИН СИН НЗ < заголовок > КБ КПБ,

после которого происходит изменение направления передачи для получения подтверждающего (СИН СИН < префикс > ДА) или отрицательного (СИН СИН < префикс > НЕТ) ответов, либо в виде части или промежуточного блока, не требующих ответа, т. е.

СИН СИН НЗ < заголовок > НТ < текст > КТ КПТ

или

СИН СИН НЗ < заголовок > РИ1 КПБ | СИН СИН НТ <
< текст > КТ КПТ.

В последнем случае применение разделителя РИ1 информации позволяет передать контрольную последовательность блока заголовка КПБ.

Заголовки и сами тексты сообщений могут разбиваться на блоки, например

СИН СИН НТ < текст > КБ КПБ ... (ответ) ... | СИН СИН НТ
< текст > КТ КПБ ... (ответ).

Для передачи текста в «прозрачном режиме» сообщение формируется в виде:

СИН СИН АР1 НТ < «прозрачный текст» > АР1 КТ КПТ

Если требуется специальный запрос у приемной станции текущего ответа, то ей посылается символ КТМ. Вслед за ним, как и за символами КТ, КБ, ДА, НЕТ, в полудуплексном режиме следует автоматическое изменение направления передачи, в синхронных каналах сопровождающееся каждый раз фазированием.

В отличие от синхронных каналов, для которых были рассмотрены все приведенные выше примеры форматов сообщений, в стартстопных линиях связи границы каждого передаваемого символа обозначаются специальными стартовыми и стоповыми сигналами (посылками).

По окончании передачи разъединение ЭВМ с АП производится только в режиме подчинения. В коммутируемой сети это производится посылкой специальной управляющей последовательности АР1 КП (разрыв). В некоммутируемой линии связи об окончании передачи свидетельствует посылка машиной пункту прямого управляющего сообщения: <префикс > КП (возврат в состояние управления). Оканчивающий передачу символ КП может посылаться в качестве отрицательного ответа на опрос пункта с целью выяснить готовность его аппаратуры к передаче информации.

СРЕДСТВА АВТОМАТИЧЕСКОГО КОНТРОЛЯ РАБОТЫ И ДИАГНОСТИКИ НЕИСПРАВНОСТЕЙ ЭВМ

16.1. ОБЩИЕ СВЕДЕНИЯ О СИСТЕМЕ ОБСЛУЖИВАНИЯ

Мультипрограммные режимы работы требуют организации оперативного автоматического контроля работоспособности ЭВМ непосредственно в процессе вычислений. Независимо от вида рабочего режима должно осуществляться постоянное слежение за техническим состоянием машины и правильностью передачи и преобразования в ней информации.

В случае обнаружения ошибок они должны автоматически анализироваться. Если причиной их являются сбои, обусловленные кратковременным нарушением нормальной работы отдельных схем, устройств или машины в целом, то средства контроля должны обеспечивать автоматическое устранение последствий таких ошибок. Случайные сбои относят к самоустраняющимся нарушениям нормальной работы. Однако в машинах могут возникать и неисправности, характеризуемые как отказы, т. е. события, заключающиеся в полной или частичной утрате работоспособности машины. Это случается при выходе из строя одного или нескольких элементов, нарушении каких-либо контактов, соединений и т. п. В общем случае выход из строя может представлять собой устойчивое отклонение по тем или иным причинам одного из параметров машины от допустимых значений. Возникновение отказов также должно своевременно обнаруживаться.

В сложных многопрограммных вычислительных машинах, помимо контроля, необходимо также наличие специальных средств автоматического поиска (диагностики) места неисправности и указания его оператору. Это очень важно для сокращения времени их восстановления.

В ЭВМ Единой системы все перечисленные выше действия предусмотрены. Они осуществляются так называемыми системами контроля и диагностики, представляющими собой совокупности взаимосвязанных аппаратных и программных средств.

Аппаратные средства — это специально вводимые в устройства контролирующие схемы. Они позволяют контролировать как отдельные простейшие операции (передача и прием информации, дешифрирование, сдвиг, сложение и т. п.), так и их опре-

деленные последовательности, из которых складывается выполнение целых команд. Ошибки в выполнении операций обнаруживаются в процессе работы устройств в темпе следования синхронизирующих сигналов. В процессорах скорость реакции на сигнал ошибки такова, что время распространения последней ограничивается одним-двумя тактами синхронизации.

Средства аппаратного контроля указывают район неисправности, определяемый местоположением контролирующей схемы, выявившей ошибку. Кроме того, они дают возможность обнаруживать ошибки, появляющиеся вследствие сбоев. Благодаря наличию специальных средств, обеспечивающих программное повторение как отдельных команд, так и сегментов программ, последствия сбоев могут автоматически устраняться.

Модели ЭВМ Единой системы снабжены средствами аппаратного контроля, позволяющими непосредственно и косвенно контролировать почти полностью (на 90—95%) оборудование процессоров. Однако из-за чрезмерного усложнения и удорожания устройств полностью охватить все схемы машины, включающие некоторые цепи управления и синхронизации, приема и выдачи информации и сигнализации, практически невозможно. Поэтому совместно с аппаратными применяют программные средства контроля.

Программный контроль осуществляется с помощью специальных программ-тестов, имеющих в комплексе программ технического обслуживания, входящем в состав математического обеспечения машин. Тесты предназначаются для проверки правильности функционирования как отдельных устройств, так и машин в целом при их наладке и контроле в процессе профилактики, а также при поиске причин возникновения ошибок в ходе выполнения программ решаемых задач. В связи с этим различают тесты наладочные, проверочные и диагностические. В моделях ЕС-1030 и ЕС-1050 широко применяются диагностические тесты, которые позволяют осуществлять диагностику как в процессе профилактики, так и оперативную в рабочих режимах по сигналам системы контроля. В последнем случае диагностические тесты, объединяемые в некоторую иерархическую систему, позволяют локализовать неисправности с определенной разрешающей способностью, например, с точностью до типового элемента замены.

Программы комплекса технического обслуживания дают возможность машинам гибко реагировать на все обнаруживаемые ошибки и сигналы прерываний от системы аппаратного контроля, управлять проведением автоматической диагностики с помощью соответствующих тестов, накапливать и анализировать информацию о состоянии машины и особенностях функционирования устройств в различные периоды и в различных условиях эксплуатации, оперативно выдавать необходимую информацию о состоянии и работоспособности машины. Программы этого комплекса используются также при профилактике и ремонте.

Взаимодействие аппаратных и программных средств контроля и диагностики осуществляется следующим образом. Аппаратные средства обнаруживают ошибки в работе устройств непосредственно при возникновении сбоев или отказов. При этом выдается соответствующий сигнал в систему прерывания. Если прерывание по сигналу системы контроля не запрещено, то после окончания синхронизации фиксируется состояние процессора путем записи содержимого основных его регистров и триггеров в установленную область оперативной памяти. Запоминание внутреннего состояния процессора дает возможность в дальнейшем автоматически продолжить решение прерванной задачи, если причиной ошибки был сбой в работе технических средств.

Система прерывания осуществляет соответствующее обращение к программе анализа ошибки. Если устанавливается, что обнаруженная ошибка должна быть устранена, то осуществляется переход к процедурам локализации места неисправности. При этом используются как программные, так и аппаратные средства.

Взаимодействие аппаратного и программного контроля сокращает непроизводительные потери времени на выяснение причин возникновения ошибок, позволяя применять многоуровневую организацию диагностики с последовательным повышением ее разрешающей способности в процессе поиска.

Кроме рассмотренного, для обеспечения оптимальных условий эксплуатации, а также испытаний и ремонта аппаратуры, в комплекты машин включают дополнительные аппаратные средства, которые могут использоваться как в непосредственной связи с машиной, так и отдельно. В совокупности с системой оперативного контроля и диагностики они образуют так называемую *систему обслуживания машины*.

В целом система обслуживания ЭВМ Единой системы в своем составе имеет:

А. Аппаратные средства:

- пульт управления и сигнализации;
- аппаратура для автономной проверки устройств машины;
- система автоматического контроля;
- система автоматической диагностики;
- аппаратура для проведения профилактических и граничных испытаний устройств машины;
- контрольно-испытательная аппаратура.

Б. Программные средства:

- процедуры ввода начальной программы;
- монитор контроля и диагностики;
- комплект тест-программ, независящих от диспетчера операционной системы;
- комплект проверочных тестов и других обслуживающих программ, управляемых операционной системой.

Рассмотрим кратко основные принципы реализации средств оперативного контроля и диагностики.

16.2. АППАРАТНЫЕ СРЕДСТВА КОНТРОЛЯ

Аппаратный контроль основан на включении в устройства машин дополнительной аппаратуры, позволяющей тем или иным способом следить за правильностью функционирования контролируемых схем и каналов передачи информации. Дополнительная аппаратура нужна для представления избыточной информации, без которой невозможен контроль. В устройствах машин Единой системы используются различные способы получения избыточной информации. Она получается либо повторным запросом или преобразованием, либо дублированием аппаратуры, либо дополнением основной преобразуемой информации вспомогательной, представленной в виде специальных (контрольных) кодов. Во всех моделях ЭВМ Единой системы в качестве основного принят контроль по mod 2. Он используется в процессорах, каналах, внешних запоминающих устройствах и устройствах ввода-вывода. Другие способы контроля, основанные на использовании дублирующей информации, других специальных кодов и т. п., находят применение в качестве дополнительных средств контроля во внешних устройствах, в устройствах ввода-вывода и в аппаратуре передачи данных.

Для сравнительного анализа избыточной информации также нужна специальная дополнительная аппаратура. Основу такой аппаратуры обычно составляют схемы сравнения. Кроме того, необходима аппаратура для формирования и преобразования контрольных кодов. Схемы сравнения информации, формирования и преобразования контрольных кодов составляют основу аппаратуры контроля. Во всех моделях они используются в больших количествах. Их включают в различные точки структур устройств таким образом, чтобы обеспечивался непрерывный сквозной контроль всех передач и преобразования информации. Например, в процессоре ЕС-2050 непрерывный аппаратный контроль охватывает:

- связи с основной и регистровой оперативными памятьями;
- цепи передачи информации между блоками и между регистрами в блоках;
- сумматор блока АС;
- сумматор блока АЦ;
- сумматор модификации адреса;
- счетчики блоков центрального управления и АЛУ;
- дешифраторы в схемах управления.

Это позволяет оперативно контролировать выполнение элементарных операций:

- передачи кодов;
- сдвига;
- счета;
- суммирования;
- дешифрирования;
- выработки управляющих сигналов.

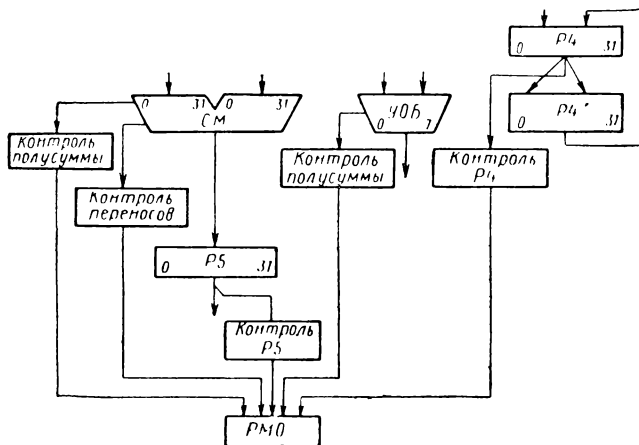


Рис. 16.1. Распределение точек контроля преобразования информации в АЛУ ЕС-2030

Всего в процессоре и основной памяти имеется 66 точек контроля. Для контроля по mod 2 в ЕС ЭВМ каждый байт информации снабжается контрольным разрядом. Так как процессор ЕС-2050 рассчитан на оперирование двойными словами, то схемы контроля строятся таким образом, что обеспечивают указание номера байта в двойном слове, в котором обнаруживается ошибка. Для регистрации сигналов ошибок в данном процессоре имеется 74-разрядный регистр ошибок РОШ. В нем 66 разрядов служат для фиксации сигнала от той или иной схемы контроля, а 8 разрядов — для номера искаженного байта. Для однобайтных схем контроля код ошибки определяется непосредственно по триггеру, в котором зафиксирован сигнал ошибки. Для многобайтных схем контроля код ошибки определяется по триггеру, зафиксированному виду ошибки, и по триггеру, соответствующему номеру байта, в котором она обнаружена.

В процессоре ЕС-2030 имеется 10 точек контроля. В цепях преобразования информации в АЛУ они распределяются, например, как показано на рис. 16.1. В АЛУ также контролируется работа всех счетчиков. Схема контроля счетчиков на рис. 16.1 не показана. В целом в модели ЕС-1030 аппаратным контролем охвачены:

- все передачи информации между блоками в процессоре;
- схемы арифметических и логических операций;
- блоки управления процессора и каналов;
- оперативная и постоянная память.

Сигналы с выходов схем контроля подаются на соответствующие триггеры 32-разрядного регистра ошибок, в данной модели называемого регистром машинных отказов (РМО).

Одним из видов аппаратного контроля является простое определение совпадения двух сравниваемых кодов.

Контроль проверкой на полное совпадение кодов позволяет обнаруживать ошибки любой кратности.

Контроль непосредственным сравнением информационных разрядов находит применение при считывании информации с магнитных лент, на которых либо для большей надежности дублируется запись, либо производится контрольное считывание при записи; он также может использоваться при проверках правильности ввода с перфокарт, перфолент и т. п. При воспроизведении магнитной или перфорационной записи дублирования информация в виде электрических сигналов параллельно с основной существует малое время, определяемое периодом следования считываемых кодов.

Контроль путем дублирования информации на всех этапах ее передачи и преобразования требует больших аппаратных затрат. Поэтому в современных ЭВМ широко применяют специальные коды, имеющие значительно меньшее количество разрядов, по сравнению с информационными n -разрядными словами.

Разрядность контрольных кодов в первую очередь зависит от требований, предъявляемых к системе контроля: должна ли она только обнаруживать некоторое количество ошибок одиночных, двойных, тройных и т. д. или еще и исправлять некоторые из них.

Одним из простейших и экономичных способов контроля с целью только обнаружения ошибок определенной кратности является контроль по модулю. Суть этого метода заключается в определении и анализе по определенным правилам контрольных кодов, представляющих собой наименьшие остатки r_x от деления на некоторый модуль самих чисел $X = x_0x_1 \dots x_{n-1}$ или сумм их цифр $\sum_{i=0}^{n-1} x_i$. В первом случае имеет место так называемый *числовой контроль* по модулю, во втором — *цифровой*.

Обозначая модуль при числовом контроле через q , а при цифровом — через d , эквивалентность величины числа его остатку можно записать как

$$X \equiv r_x \pmod{q},$$

а суммы цифр

$$\sum_{i=0}^{n-1} x_i \equiv r_x \pmod{d}.$$

Контроль по модулю можно применять не только для проверки правильности пересылок информационных слов, но и для обнаружения ошибок при выполнении арифметических и логических операций. Числовой контроль по модулю при выполнении ариф-

метических операций основан на известных из теории чисел тождествах:

для суммы — $X_1 + X_2 \equiv (r_{X_1} + r_{X_2}) \pmod{q}$;

для произведения — $X_1 X_2 \equiv (r_{X_1} r_{X_2}) \pmod{q}$;

для деления — $r_{X_2} r_{X_3} \equiv (r_{X_1} - r_{X_4}) \pmod{q}$,

где в последнем выражении r_{X_1} , r_{X_2} , r_{X_3} и r_{X_4} соответственно остатки от деления делимого X_1 , делителя X_2 , частного X_3 и остатка $X_4 = X_1 - X_2 X_3$ на модуль q .

Для двоичных кодов простейшим является контроль по $\text{mod } 2$, т. е. при $q = 2$. В данном случае при основании системы числения $N = 2$ и $d = q = 2$. Следовательно,

$$r_x \pmod{2} = r_x \pmod{2} = \left(\sum_{i=0}^{n-1} x_i \right) \pmod{2},$$

т. е. одноразрядный контрольный код определяется просто как сумма по $\text{mod } 2$ цифр числа X . Такой контроль в основном и применяется в ЭВМ Единой системы. Каждый байт информации в них автоматически сопровождается девятым, контрольным, разрядом.

Контроль по $\text{mod } 2$ может осуществляться либо проверкой четности, либо — нечетности. В первом случае значение контрольного разряда вырабатывается таким образом, чтобы сумма единиц в информационном слове, включая контрольный разряд, была всегда четной, а во втором — нечетной. При использовании проверки нечетности в разрядах слова вместе с контрольным разрядом в любом случае будет хотя бы одна единица. Появление во всех разрядах, в том числе и контрольном, нулей будет свидетельствовать либо об ошибке, либо о полном пропадании информации. В процессорах, оперативной памяти, каналах, устройствах ввода-вывода и аппаратуре передачи данных ЕС ЭВМ применяется проверка на нечетность преобразуемой и управляющей информации. По затратам дополнительного оборудования контроль по $\text{mod } 2$ является наиболее экономичным.

На рис. 16.2 показана структурная схема контроля двоичных кодов X , поступающих в 9-разрядный регистр P . Контрольный разряд, формируемый по условию нечетности и передаваемый в составе кода X , запоминается в 8-м разряде регистра. Для проверки правильности принятого кода схемой свертки — схемой суммирования по $\text{mod } 2$ ($M2$) — определяется сумма по $\text{mod } 2$ содержимого 0—7 разрядов регистра, которая сравнивается схемой сравнения CP с содержимым 8-го разряда. При их совпадении схема сравнения вырабатывает сигнал ошибки.

Пример построения схемы суммирования по $\text{mod } 2$ 8-разрядных кодов с включенной на выходе схемой сравнения дан на рис. 16.3. Схема суммирования выполнена трехкаскадной. Состав-

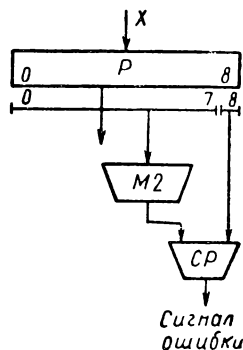


Рис. 16.2. Структура схемы контроля содержимого регистра

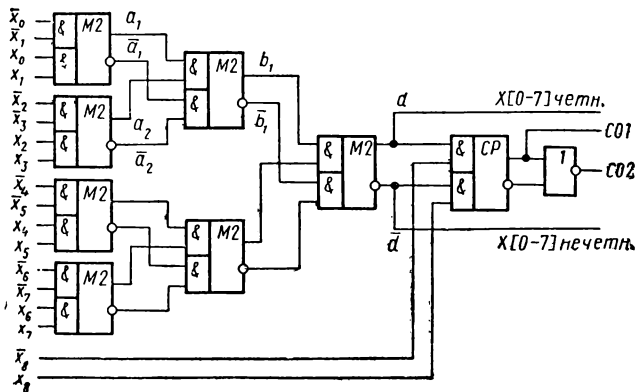


Рис. 16.3. Схема суммирования по mod2 8-разрядных кодов со схемой сравнения на выходе

ляющие ее двухразрядные схемы M2 имеют прямые и инверсные выходы. На прямом выходе сигнал высокого уровня формируется тогда, когда сумма по mod 2 двух входных переменных — четная. Если схема M2 строится на элементах И—ИЛИ—НЕ, то выходные переменные a_1 и \bar{a}_1 вырабатываются, например, согласно формулам

$$a_1 = x_0 \bar{x}_1 \vee \bar{x}_0 x_1;$$

$$\bar{a}_1 = \overline{x_0 \bar{x}_1 \vee \bar{x}_0 x_1}.$$

Во второй и третьей ступенях сигналы b_i и \bar{b}_i , d и \bar{d} получаются аналогичным образом. Поэтому сигнал d высокого уровня указывает на то, что 8-разрядный код X [0—7] — четный. Результат свертки сравнивается схемой сравнения со значением контрольного разряда x_8 , сформированным по условию нечетности. В случае совпадения их значений на прямом выходе схемы сравнения устанавливается потенциал высокого уровня, соответствующий сигналу ошибки CO1. Для выявления отказов в основном и контрольном оборудовании, в результате которых на обоих выходах схемы CP могут появиться низкие уровни потенциала, имеется на выходе дополнительная схема ИЛИ—НЕ. Она вырабатывает сигнал ошибки CO2, когда на обоих выходах схемы сравнения устанавливаются сигналы низкого уровня. Возникновение сигналов высокого уровня на обоих выходах обнаруживается сигналом CO1.

Арифметические и логические операции, выполняемые цифровыми машинами в процессе преобразования информации, состоят из последовательностей элементарных действия—микроразрядных операций — таких, как передача кода, его поразрядное инвертирование, сдвиг, сложение, преобразование в управляющий сигнал на определенной шине (дешифрирование) и т. д. Как и для передач информации, контроль по модулю приме-

ним и для проверки правильности выполнения таких действий. Правда, способы построения схем контроля в каждом конкретном случае имеют свои специфические особенности. Это обуславливается различием в принципах построения и работы контролируемых микрооперационных устройств (типовых узлов).

Из всего многообразия схем контроля перечисленных микроопераций рассмотрим принцип организации контроля по модулю процессов сложения.

В общем случае при модулях $q > 2$ для контроля сложения требуются дополнительные схемы, показанные на рис. 16.4. Одновременно со сложением чисел X и Y в специальном сумматоре (СМКК) по $\text{mod } q$ складываются их контрольные коды r_x и r_y . Сумма $(r_x + r_y) \pmod q$ в схеме сравнения СР сравнивается с контрольным кодом результата сложения в сумматоре числа СМЧ. Этот контрольный код формируется схемой свертки СВ по $\text{mod } q$. При несовпадении суммы контрольных кодов с контрольным кодом результата сложения выдается сигнал ошибки.

Как было указано выше, в ЭВМ Единой системы применяется контроль по $\text{mod } 2$. Особенность проверки процесса суммирования при таком модуле состоит в том, что для полноты контроля процессов, происходящих в сумматоре СМЧ, недостаточно только сравнение значения контрольного кода суммы r_s с суммой $(r_x + r_y) \pmod 2$. Необходимо еще учитывать и анализировать процессы, происходящие в цепях переносов.

Действительно, значение контрольного разряда r_s суммы двух байт определяется следующим соотношением:

$$\begin{aligned} r_s \pmod 2 &= s_0 \oplus s_1 \oplus \dots \oplus s_7 = (x_0 \oplus y_0 \oplus p_1) \oplus \\ &\oplus (x_1 \oplus y_1 \oplus p_2) \oplus \dots \oplus (x_7 \oplus y_7 \oplus p_{вх}) = (x_0 \oplus x_1 \oplus \dots \oplus x_7) \oplus \\ &\oplus (y_0 \oplus y_1 \oplus \dots \oplus y_7) \oplus (p_1 \oplus p_2 \oplus \dots \oplus p_{вх}) = \\ &= r_x \pmod 2 \oplus r_y \pmod 2 \oplus r_p \pmod 2 \end{aligned}$$

или просто

$$r_s = r_x \oplus r_y \oplus r_p, \quad (16.1)$$

где $x_0, \dots,$

x_7, y_0, \dots, y_7 — значения разрядов слагаемых;

s_0, \dots, s_7 — значения разрядов суммы;

$p_1, \dots, p_{вх}$ — переносы между разрядами и входной перенос в младший (7-й) разряд;

r_x, r_y и r_s — контрольные разряды слагаемых X, Y и суммы S ;

r_p — сумма по $\text{mod } 2$ возникающих при сложении переносов.

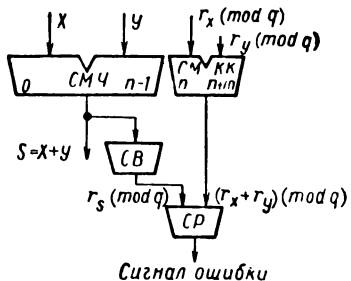


Рис. 16.4. Схема организации контроля по $\text{mod } q$ процесса сложения

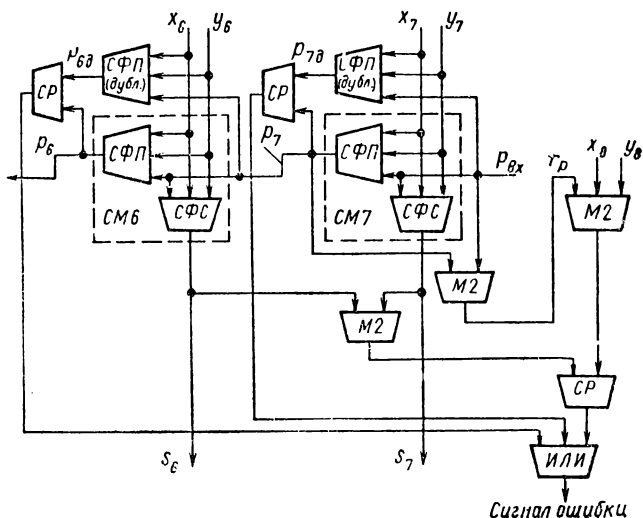


Рис. 16.6. Структурная схема двух разрядов сумматора с цепями дублирования переносов и контролем по mod 2

считать независимыми. Они обнаруживаются схемой сравнения сигналов p_6 и $p_{6д}$ ошибки же, которые могут возникнуть под влиянием ошибочного значения p_7 обнаруживаются схемой сравнения сигналов p_7 и $p_{7д}$ и т. д. Таким образом, в данной схеме обнаруживаются одиночные ошибки, которые могут возникать в цепях формирования поразрядных сумм s_i и в цепях формирования и распространения переноса. Контроль дублированием цепей переносов применяется также и в сумматорах, имеющих цепи параллельных переносов. Это имеет место, например, в сумматоре АЛУ процессора ЕС-2030.

Аппаратный контроль работы схем, в которых непосредственно выполняются микрооперации, и схем, которые вырабатывают управляющие сигналы, дает возможность машинам через системы прерывания быстро реагировать на возникающие сигналы ошибок.

16.3. ПРОГРАММНЫЕ СРЕДСТВА ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

В составе математического обеспечения ЕС ЭВМ предусмотрены комплексы программ технического обслуживания, как непосредственно связанные с операционной системой и работающие под ее управлением, так и независимые от нее. *Самостоятельными* являются программы, служащие для наладки машин при их изготовлении и вводе в эксплуатацию (наладочные тест-программы), для проверки устройств во время профилактики и ремонта, а также при обнаружении неисправностей в процессе эксплуатации (проверочные и диагностические тест-программы).

К этой группе программных средств также относятся: управляющая программа тест-монитор, предназначенная для самостоятельного управления выполнением проверочных и диагностических тест-программ, и вспомогательные служебные программы, необходимые для рациональной организации обслуживания тест-программ, расположенных на магнитных носителях. С помощью служебных программ обеспечиваются: генерирование диагностических магнитных лент, размещение тест-программ на магнитных лентах и дисках; копирование, компоновка, исключение, исправление, печатание тестов и т. п. *Управляемыми* операционной системой являются программы, обеспечивающие анализ сигналов прерываний от схем контроля, автоматические сбор и редактирование информации о сбоях процессоров и каналов, восстановление системы после сбоя, оперативную периодическую проверку внешних устройств как занятых пользователем, так и свободных.

Наладочные тесты предназначены для проверки устройств (от процессора до печатающей машинки) и обнаружения в них неисправностей при наладке машин. Испытания этими тестами проводятся в определенной последовательности. Каждый тест выполняется автономно. Для проверки совместной работы устройств и машины в целом некоторые из этих тестов объединяются во взаимосвязанный комплекс. Для каждой модели имеется свой комплект наладочных тест-программ и к ним управляющая программа, обеспечивающая их загрузку и автоматический автономный или комплексный прогон. В моделях ЕС-1030 и ЕС-1050 применяются наладочные тесты двух уровней. В начале вручную проверяется минимум аппаратуры, позволяющей ввести управляющую программу, с помощью которой затем загружаются и выполняются тесты второго уровня.

В модели ЕС-1050 начальными являются базовые наладочные тесты процессора, оперативной памяти, каналов и устройства ввода с перфокарт. После наладки этими тестами, дальнейшая наладка модели осуществляется с помощью проверочных программ — тест-секций, работающих под управлением тест-монитора. В составе проверочных программ предусмотрены тест-секции детальной проверки всех режимов работы процессора (всех операций, схем прерывания, защиты памяти, таймера, средств прямого управления), оперативной памяти и памяти ключей защиты ОП, селекторного канала, мультиплексного канала, накопителей и их устройств управления, всевозможных устройств ввода-вывода.

Тест—монитор управляет выполнением проверочных и диагностических тест—секций. Без участия оператора он может производить стандартную обработку прерываний, загрузку тест-секций в оперативную память, прогон как отдельных тестов, так и тестов всей модели, закливание отдельных тестов или примеров. При этом обеспечивается возможность гибкой оперативной связи с оператором.

Различают следующие фазы работы системы обслуживания:

- рабочая;
- профилактическая;
- ремонтная.

В рабочей фазе под управлением операционной системы осуществляется автоматическое накопление статистики по сбоям, возникающим при выполнении рабочих программ, и ее анализ, а также периодическая тестовая проверка процессора, каналов и внешних устройств параллельно с решением задач.

При обнаружении схемами контроля ошибки выполнение рабочей программы немедленно прекращается (рис. 16.7). Для ограничения распространения ошибок прерывание сопровождается блокировкой синхронизации в пределах одного-двух тактов после установки в единичное состояние триггера регистра ошибок РОШ. В модели ЕС-1050 сначала через полтакта в пределах рамы стойки блокируется та синхросерия, по которой сигнал ошибки зафиксировался в РОШ. Затем через такт после установки в единичное состояние триггера РОШ в пределах той же рамы блокируется другая серия синхронизации. Серии синхронизации других рам, в которых смонтировано оборудование процессора, блокируются через полтора-два такта после фиксации сигнала ошибки.

С остановом синхронизации в процессоре начинается выполнение режима «Учет». Он предназначается для запоминания состояния основных регистров и управляющих триггеров процессора в отведенную для этого область оперативной памяти. В модели ЕС-1050 в ОП, начиная со 128-й ячейки, записывается информация объемом в 39 двойных слов, из которых 34 содержат информацию основных регистров и управляющих триггеров, а 5 — значения соответствующих ей контрольных разрядов.

Запись состояния процессора в ОП осуществляется с помощью блока контроля и диагностики, работа которого синхронизируется синхроимпульсами с основного генератора. Как и для процессора,

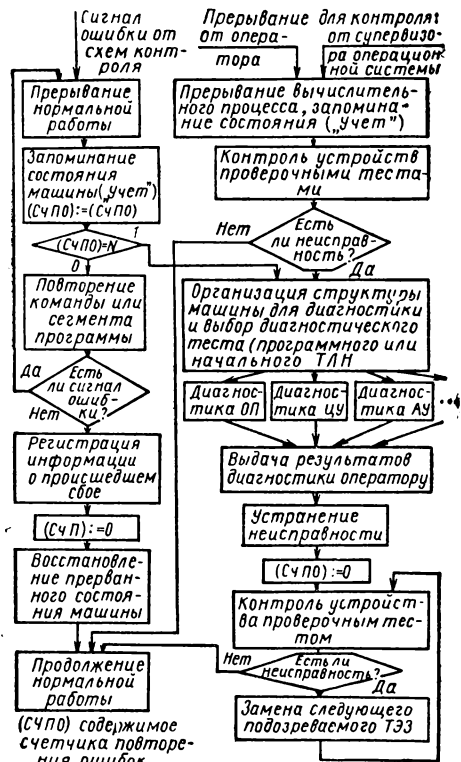


Рис. 16.7. Процесс автоматической проверки и диагностики машины

может регистрироваться состояние любого блока оперативной памяти или канала, где появляется ошибка. Это дает возможность в последующем анализировать накапливаемую таким образом информацию с целью выявления мест неустойчивых отказов.

После запоминания состояния процессора операционной системой в работу включается программа анализа ошибки. Для этого ССП исполнявшейся программы запоминается в ячейках 48—55 ОП, а из ячеек 112—120 ОП по прерыванию от схем контроля выбирается новое ССП. Для определения степени устойчивости ошибки задается определенное число N повторения команды или сегмента программы (рис. 16.7). Возможный уровень повторения устанавливается по порогу повторения для данной команды. Порог повторения считается не пройденным, пока операнды не искажены. Если к моменту возникновения ошибки порог повторения команды оказался пройденным, например, началась запись результата на место первого операнда, то повторение возможно только на уровне сегмента программы. Для того чтобы можно было повторять сегменты, в программах предусматриваются контрольные точки. В этих точках специальными командами осуществляется запоминание промежуточных результатов, позволяющих с данных мест повторять участки программ.

Если при повторении команды или сегмента программы ошибка не обнаруживается, то это означает, что был случайный сбой. В этом случае информация о состоянии машины, а также дополнительно регистрируемые специальной программой данные об устройствах ввода-вывода и внешних запоминающих устройствах, работавших в момент возникновения ошибки, астрономическое время, сведения об исполнявшейся программе и т. п. переписываются из фиксированной области оперативной памяти в долговременную память. Затем состояние машины восстанавливается и происходит переход к продолжению исполнения прерванной программы. Он организуется таким образом, чтобы ошибка, вызванная сбоем, была автоматически устранена.

Учет всей информации о состоянии машины и условиях, в которых происходили прерывания по сигналам ошибок, позволяют накапливать данные о всех нарушениях нормальной работы машины в процессе эксплуатации. Специальными программами, работающими под управлением операционной системы, производится систематизация, редактирование и анализ этих данных. По запросам обслуживающего персонала они в установленной форме могут быть выданы на печать.

Если в результате повторения N раз команды или сегмента программы программа анализа обнаруживает систематическую ошибку, то система обслуживания переключается в фазу ремонта. При этом управление передается тест-монитору. Он организует диагностические исследования устройства, где появилась неисправность, индцированная сигналом ошибки, и управляет ими.

Фаза ремонта разбивается на три этапа. К первому этапу относится диагностика отказавшего устройства и обнаружение места неисправности, ко второму — собственно ремонт, а на третьем, завершающем, этапе производится послеремонтная проверка устройства проверочным тестом (рис. 16.7).

Место неисправности с определенной точностью можно установить с помощью диагностических тестов. Однако если точность указания места неисправности недостаточна для восстановления машины за заданное время, то осуществляется переход к аппаратной диагностике. Накопление информации о работоспособности устройств машины и средств аппаратного контроля в период рабочей фазы можно осуществлять также с помощью специальной команды ДИАГНОСТИРОВАТЬ. Она не имеет мнемонического обозначения и поэтому не применяется в рабочих программах. Эта команда используется только в тестах процессора и каналов. Она поэтому доступна только супервизору операционной системы.

Команда ДИАГНОСТИРОВАТЬ имеет формат SI. В процессе, ее выполнения из основной оперативной памяти по адресу $(B_1) + D_1$ считывается двойное слово, которое поступает в блок контроля и диагностики. Оно является управляющим словом команды ДИАГНОСТИРОВАТЬ (УСКД), задающим принудительный режим проверки процессора или каналов. При этом вначале в процессоре или каналах принудительно выполняются действия режима «Учет». Наличие в блоке контроля и диагностики специальных счетчиков, исходная установка которых производится информацией из УСКД, позволяет задавать интервал времени, в течение которого в каждом периоде синхронизации (в модели ЕС-1050 через каждые 160 нс) можно повторять режим «Учет». Благодаря этому представляется возможным осуществлять «покадровую съемку» как отдельных команд, так и целых тестов. В модели ЕС-1050 максимальный интервал времени, задаваемый одной командой диагностики, в течение которого может идти «покадровая съемка», равен 10,48 мс. В каждом «кадре» в режиме «Учет» в ОП фиксируется 350 байт. Исполнение проверочных тестов в таком режиме позволяет использовать их как диагностические. При этом проверяться и диагностироваться могут и схемы контроля, для чего с помощью УСКД имеется возможность принудительно нарушать условие четности кодов в различных схемах, где имеется аппаратный контроль. Кроме сигналов ошибок и команды диагностики, режим «Учет» может выполняться также по сигналам с пульта управления, если желательно для последующего анализа запомнить состояние процессора в момент его останова.

Информация о сбоях во внешних устройствах получается с помощью канальной команды УТОЧНИТЬ СОСТОЯНИЕ.

Профилактическая фаза заключается в плавном периодическом выполнении проверочных тестов как отдельных устройств, так и системы в целом. Такой контроль может сопровождаться

автоматическим изменением напряжений источников питания и частоты синхронизации. При обнаружении неисправностей осуществляется переход в фазу ремонта.

16.4. ПРОГРАММНО-АППАРАТНАЯ ДИАГНОСТИКА

Диагностика устройств ЭВМ Единой системы осуществляется с помощью диагностических программ и специальных диагностических процедур, реализуемых аппаратурой встроенного в процессор блока контроля и диагностики (БКД). Диагностические программы, так же как и проверочные, выполняются под управлением тест-монитора. По результатам, получаемым после прогона тестов, автоматически заполняется диагностическая таблица. Эта таблица затем может анализироваться либо вручную, либо самой машиной. Для обеспечения возможности анализа составляются специальные каталоги, в которых устанавливается соответствие между видами заполнения таблиц, и конкретными неисправностями. При автоматическом анализе эти каталоги хранятся в памяти машины.

В модели ЕС-1050 диагностические тесты дают возможность локализовать неисправность во всех устройствах с точностью до 2—3 ТЭЗ. Для того чтобы можно было воспользоваться диагностическими тестами, необходима исправность так называемого «ядра» процессора — блоков управления памятью и центрального управления (УП и ЦУ). Блок УП, имеющий несложную структуру, проверяется вручную. Диагностика блока ЦУ осуществляется с помощью специальных тестов локализации неисправностей (ТЛН). Если и диагностические исследования начинаются с «ядра», то процесс диагностики всей ЭВМ обычно строится по методу последовательного расширения работоспособной области процессора и других устройств. Такой метод часто называют методом раскрутки.

Для диагностики устройств с большей разрешающей способностью, по сравнению с диагностическими тестами, в ЭВМ Единой системы находит широкое применение испытание ТЛН с помощью блока КД. Применение этих непрограммных тестов напоминает стендовую проверку приборов дискретного действия. Для различных блоков центрального управления и арифметикологических устройств имеются комплексы взаимосвязанных ТЛН. Граф взаимосвязи имеет вид дерева. Одна из ветвей дерева соответствует процессу исполнения некоторой последовательности ТЛН, при котором устанавливается отсутствие неисправности в устройстве. Другие ветви приводят к локализации одной или группы неисправностей. Точность локализации зависит от особенностей проверяемых цепей и от разрешающей способности тестов последовательно связанной группы (ветви дерева).

Испытание проверяемых блоков процессора тестами локализации неисправностей осуществляется блоком контроля и диагно-

стики. Оно представляет собой диагностический эксперимент, заключающийся в подаче на входы проверяемых узлов и блоков определенных воздействий, называемых стимулами, преобразовании их по заданной последовательности сигналов и сравнении результата с предварительно вычисленным (эталонным) значением. Вся информация: коды-стимулы, количество тактов, которое необходимо выполнить для требуемого преобразования стимулов, набор управляющих сигналов, эталонный результат, координаты контролируемого элемента и адреса следующих возможных для исполнения ТЛН, указывается в многобайтном коде каждого теста. В модели ЕС-1050 все ТЛН имеют фиксированный формат, содержащий 14 двойных слов (112 байтов).

ТЛН постоянно хранятся на магнитной ленте. В модели ЕС-1050 режим ТЛН задается с пульта управления. Первый ТЛН определенного тестового «дерева» программой начальной загрузки через селекторный канал записывается в фиксированную область оперативной памяти. Блок контроля выбирает из соответствующего поля ТЛН коды-стимулы и последовательно заполняет ими определенные регистры и триггеры проверяемого устройства. После этого выбирается код, определяющий количество тактов синхронизации, которое необходимо выполнить, и заносит его в специальный счетчик. Блок диагностики обеспечивает выполнение заданного количества тактов. Результат выполнения теста сравнивается с эталоном.

По результату сравнения устанавливается, какой следующий из двух возможных тестов должен быть вызван для исполнения в оперативную память. Последний ТЛН в той или иной ветви диагностического дерева указывает место неисправности. Информация об этом выдается оператору.

Если последний тест не обнаружил ошибку то осуществляется переход к проверке других узлов. Если требуется определение места неисправности с точностью до одного или нескольких модулей интегральных схем, то это можно сделать с помощью автоматически составляемых и выводимых на перфокарты диагностических таблиц.

После устранения неисправности соответствующее устройство проверяется контрольным тестом. Если ошибка не обнаруживается, то машина переводится в режим нормальной работы.

ОРГАНИЗАЦИЯ ФУНКЦИОНИРОВАНИЯ ЕС ЭВМ В РАЗЛИЧНЫХ РЕЖИМАХ

17.1. РЕЖИМЫ РАБОТЫ ЭВМ ЕДИНОЙ СИСТЕМЫ

Все модели ЕС ЭВМ относятся к категории мультипрограммных машин, в которых используется сочетание программных и аппаратных средств управления, образующих некоторую иерархическую структуру. На верхнем уровне этой структуры располагаются программные средства управления, а на нижнем — аппаратные. Программные средства составляют операционную систему, устанавливающую порядок функционирования всех устройств машины при различных режимах ее работы. Тем самым программисты избавлены от необходимости каждый раз вносить этот порядок в свои программы.

Программно-аппаратные средства управления являются более гибкими по сравнению с чисто аппаратными, характерными для машин первого и даже второго поколений. Они выполняют более разнообразные функции и содержат ряд специфических систем, таких как система прерываний и приоритетов, система динамического распределения памяти, система защиты памяти. Наличие развитой программно-аппаратной системы управления позволяет организовать несколько различных режимов работы машины, основными из которых являются следующие: однопрограммный режим, режим пакетной обработки, режим разделения времени, режим запрос-ответ, комбинированные режимы. Все они, кроме первого, относятся к мультипрограммным (многопрограммным) режимам работы.

Однопрограммный режим. Это такой способ организации решения задач, при котором все устройства машины заняты выполнением только одной программы.

Вообще возможны две разновидности однопрограммного режима.

1. Однопрограммный режим с непосредственным доступом абонента (пользователя, программиста) к машине. В этом режиме абонент сам работает за пультом ЭВМ: вводит свою программу, запускает ее, наблюдает за выводом результатов. Он может высвечивать на пульте содержимое регистров процессора или ячеек основной памяти, модифицировать инструкции, значения некоторых параметров. Находясь за пультом машины, абонент имеет

ее в своем полном распоряжении, он ведет с ней диалог. Преимущества такого режима могут использоваться только на уровне машинного языка.

Применение однопрограммного режима с непосредственным доступом чрезвычайно невыгодно с точки зрения использования производительности ЭВМ. Действительно, время реакции абонента (работа с пультом, время обдумывания дальнейшего плана действия и т. д.) очень велико по сравнению с временем реакции машины, поэтому машинное время используется нерационально, машина «простаивает». Такой режим работы характерен для начальной стадии применения ЭВМ. Применение его недопустимо при решении задач на машинах Единой системы.

2. Однопрограммный режим с последовательным выполнением программ без участия абонентов (режим последовательной обработки). В этом режиме в памяти ЭВМ имеется несколько программ абонентов, введенных заранее и выполняемых последовательно под управлением операционной системы. Автоматический переход к другой программе осуществляется только по окончании текущей программы или при обнаружении ошибки в программе во время ее загрузки или выполнения. При таком режиме работы возможности параллельной работы основных устройств машины используются не полностью.

Режим пакетной обработки. Это такой способ организации одновременного решения нескольких задач в ЭВМ, при котором вся необходимая информация (программы, исходные данные) вводится в память машины заранее, до начала решения задач, а в ходе их решения вмешательство потребителей с пульта не допускается, они не имеют непосредственного доступа к машине. Заранее подготовленные ими задачи (программы) вводятся в память машины, с помощью операционной системы из них формируются пакеты задач (заданий, программ), которые затем реализуются. Спустя определенное время потребители получают результаты решения своих задач.

Основная цель организации режима пакетной обработки — минимизация времени на решение заданного набора задач (программ), за счет обеспечения параллельной работы основных устройств машины, т. е. за счет повышения коэффициента использования каждого устройства ЭВМ. Выполнение программ, составляющих пакет, осуществляется под управлением операционной системы таким образом, что при этом обеспечивается наиболее широкое использование совмещений в работе отдельных устройств ЭВМ, а значит и повышение коэффициента их загрузки. Естественным показателем качества функционирования ЭВМ в режиме пакетной обработки является величина $T_{\text{знз}}$ — время на реализацию заданного набора задач.

Программы, собираемые в пакет, могут иметь различные приоритеты. В режиме пакетной обработки, как и в других режимах работы ЭВМ, может использоваться как система стати-

ческих приоритетов, так и система динамических приоритетов. В первом случае приоритет программы является абсолютным, не связанным с динамикой процесса реализации пакета. Во втором случае присвоение программе того или иного приоритета осуществляется в ходе выполнения программ пакета. Присвоение динамических приоритетов связывается, например, со временем ожидания программ в процессе реализации пакета.

Работа ЭВМ в режиме пакетной обработки связана с реализацией трех процессов: процесса назначения приоритетов для выполняемых программ, процесса прерывания выполнения программы по той или иной причине (необходимо вывести результаты решения, необходимо повторить счет из-за возникновения ошибки и т. д.), процесса защиты данных, относящихся к прерванным программам, от воздействия выполняемых в данный момент программ.

Организация работы ЕС ЭВМ в режиме пакетной обработки рассматривается в п. 17.2.

Режим разделения времени. Это такой способ организации решения задач, при котором некоторое число независимых абонентов (потребителей) с помощью периферийных устройств ввода-вывода имеет в процессе решения своих задач непосредственный, постоянный и одновременный доступ к ЭВМ. Оперативное взаимодействие удаленных абонентов с ЭВМ обеспечивается через специальные или общегосударственные линии связи. Удовлетворение требований (запросов) на решение задач (на выполнение своих программ) одновременно со стороны нескольких абонентов осуществляется в приемлемое для них время и состоит в реализации их программ в определенной последовательности под управлением операционной системы. При одновременном обращении нескольких абонентов машина реагирует на их запросы с задержкой времени, которая, с точки зрения абонента, представляется такой же, как и при индивидуальном пользовании, что следует из факта несоответствия скоростей реакции человека и современных ЭВМ. Режим разделения времени сочетает эффективное использование возможностей ЭВМ с предоставлением абоненту удобств индивидуального пользования.

Электронная вычислительная машина вместе со всем периферийным оборудованием и соответствующим математическим обеспечением, работающим в режиме разделения времени, может быть названа системой с разделением времени (СРВ).

Для режима работы с разделением времени на первый план выдвигаются вопросы, связанные с обеспечением удобств абонентов в процессе их взаимодействия с ЭВМ. Эти удобства определяются наличием общедоступных и простых в использовании языков программирования, качеством операционной системы, наличием достаточно простого командного языка для связи абонентов с машиной посредством операционной системы, организацией длительного хранения информационных массивов (в част-

ности, программ) абонентов, возможностью совместного использования одновременно работающими абонентами библиотеки стандартных программ, хранящихся в ЗУ машины, возможностью получения результатов решения задач в приемлемые сроки. Количественно оценить показатели, определяющие степень удобства тех или иных систем программирования, операционных систем, систем хранения информационных массивов, либо довольно трудно, либо вообще невозможно. Поэтому качество функционирования ЭВМ в режиме разделения времени чаще оценивается с помощью временных показателей.

Наиболее распространенными являются следующие показатели качества функционирования систем с разделением времени.

1. Полное время обслуживания абонента T_{oa} — это интервал времени от момента подачи запроса на обслуживание до момента получения результатов решения задачи абонента или ответа на запрос. Иногда это время называется также временем реакции системы на запрос.

2. Максимальное число абонентов, которые одновременно могут обслуживаться системой.

3. Коэффициент замедления — отношение времени T_{oa} , полученного при максимальном числе одновременно работающих абонентов, ко времени T_{oa} , определенному для случая, когда запрос на обслуживание поступил от одного абонента.

4. Коэффициент эффективности программ операционной системы η , с помощью которого оцениваются затраты времени на работу самой операционной системы; он представляет собой отношение $T_{oc}/T_{ц}$, где $T_{ц}$ — время одного цикла обслуживания; это время, в течение которого каждой из обслуживаемых рабочих программ абонентов предоставляется один квант времени; T_{oc} — время, затрачиваемое на работу программ операционной системы в течение $T_{ц}$.

Определение конкретных значений указанных показателей с достаточной точностью связано с необходимостью моделирования процесса функционирования системы с разделением времени с учетом специфических для такой системы факторов.

Существенной особенностью системы с разделением времени является то, что абоненты составляют неотъемлемую часть системы, поэтому точная оценка качества ее функционирования должна производиться с учетом их поведения. В СРВ абонент может рассматриваться как адаптивная и изменяющаяся система. Понятие взаимодействия абонента с центральным процессором СРВ включает в себя вопросы связи, обработки и обмена информацией. В процессе такой активной двусторонней связи происходит принятие решений (обычно на базе предварительно проведенных вычислений) и производятся некоторые направленные управляющие воздействия.

Можно выделить два аспекта в поведении абонента:

- абонент наперед программирует свои действия, т. е. он заранее решает, что необходимо сделать в той или иной ситуации;
- абонент производит действия после обдумывания, некоторой оценки, в результате которой выявляется наиболее приемлемый путь к достижению поставленной цели.

Оба эти аспекта могут иметь место в различном соотношении не только при решении различных задач в СРВ, но и при решении одной достаточно сложной задачи. Эффект взаимодействия человека с машиной в значительной степени определяется квалификацией абонентов, уровнем их подготовки, поэтому при полной и глубокой оценке качества функционирования СРВ каждый абонент должен представляться своей математической моделью. Тогда объединенная модель абонентов представляет собой сложную модель, состоящую из отдельных моделей абонентов, работающих с набором различных задач.

Отметим особенности и преимущества СРВ по сравнению с ЭВМ, работающими в режиме пакетной обработки.

1. Возможность коллективного использования центрального процессора. Многие абоненты имеют возможность независимо друг от друга непосредственно и оперативно контактировать с процессором. Наличие достаточно простой и стандартной периферийной аппаратуры, применение в СРВ довольно простых алгоритмических языков существенно облегчают общение с процессором и способствуют расширению круга абонентов. Вопрос обеспечения абонента средствами, открывающими ему доступ к машине, в СРВ решается наиболее удачно. В отличие от этого в ЭВМ с пакетной обработкой абоненты полностью отстранены от машины, в результате чего усложняется процесс обмена информацией между ними и машиной.

2. Наличие большой централизованной памяти программ и данных. Необходимость в такой памяти объясняется тем, что непосредственно на абонентском пункте запомнить весь требуемый абоненту массив информации не представляется возможным. В централизованной памяти хранятся личные архивы абонентов. Предусматривается возможность обмена информацией между архивами, что создает дополнительные преимущества абонентов.

3. Вычислительные мощности обходятся дешевле и используются более эффективно по мере роста скорости обработки информации, объема централизованной памяти и расширения набора абонентских пунктов.

Режим запрос-ответ. Это такой способ организации решения задач, при котором их программы постоянно хранятся в ЗУ машины, а запросы на их выполнение и необходимые исходные данные поступают извне от абонентов, имеющих прямой доступ к машине. Каждому запросу соответствует своя программа, которая вырабатывает соответствующий ответ. Число таких отвечающих программ ограничено емкостью ЗУ. Следовательно, ограничен и

набор допустимых запросов, поэтому режим работы запрос-ответ применяется в основном для решения таких задач, как резервирование мест на транспорте, выдача различных справок и т. д.

Поток запросов может быть детерминированным или вероятностным, причем в последнем случае возможен как стационарный, так и нестационарный поток. Запросы могут иметь различные приоритеты и, кроме того, временные ограничения на срок их обслуживания. В связи с этим возникает задача диспетчирования, определения порядка обслуживания запросов. Диспетчирование определяет некоторую стратегию обслуживания, с помощью которой из множества поступивших запросов выбирается совокупность запросов для обслуживания в первую очередь. Правила выбора запросов определяют дисциплину обслуживания.

Наиболее часто применяются следующие дисциплины обслуживания запросов:

- в порядке поступления запросов, т. е. по принципу «первый пришел — первый обслужен»;

- в порядке абсолютных приоритетов, когда все запросы разбиты на группы и каждой группе присвоен определенный приоритет. Обслуживание запросов из группы с некоторым приоритетом не начинается до тех пор, пока не будут обслужены все запросы из групп с более высоким приоритетом;

- в порядке относительных приоритетов, назначаемых в зависимости от принадлежности запроса к той или иной приоритетной группе и от момента его поступления;

- в порядке динамических приоритетов, назначаемых в зависимости от принадлежности запроса к той или иной приоритетной группе и от допустимого времени ожидания обслуживания.

Выбор дисциплины обслуживания определяется критерием, по которому оценивается качество функционирования ЭВМ, характером потоков запросов и производительностью машины.

Машина, работающая в режиме запрос-ответ, может быть загружена не полностью. В таком случае в промежутках между обслуживанием запросов машина может решать другие задачи, например, в режиме пакетной обработки. Другой пример организации комбинированного режима работы ЭВМ — это ее работа в режиме разделения времени, а в промежутках, когда она свободна от обслуживания запросов активных абонентов, — работа в однопрограммном режиме или в режиме пакетной обработки.

17.2. ОРГАНИЗАЦИЯ ФУНКЦИОНИРОВАНИЯ ЭВМ ЕДИНОЙ СИСТЕМЫ В РЕЖИМЕ ПАКЕТНОЙ ОБРАБОТКИ

Организация работы ЭВМ в режиме пакетной обработки связана с решением таких вопросов, как выбор дисциплины (порядка) реализации рабочих программ пакета, выбор вида пакетной обра-

ботки, подготовка рабочих программ к пакетной реализации, планирование загрузки устройств машины.

Выбор той или иной дисциплины реализации рабочих программ определяется назначением ЭВМ и характером решаемых задач. Если задачи дифференцируются по своей срочности и важности, для них устанавливаются соответствующие уровни приоритетов с таким расчетом, чтобы в первую очередь и в приемлемые сроки были решены наиболее важные задачи. Приоритетность задач внутри данного пакета может устанавливаться по степени их важности и по информационной связности: если для некоторой задачи в качестве исходных данных используются результаты (промежуточные или окончательные) решения другой задачи, то эта другая задача, естественно, должна решаться в первую очередь. Как уже отмечалось, в режиме пакетной обработки может использоваться как система статических приоритетов, так и система динамических приоритетов. Наилучшие результаты с точки зрения обеспечения минимальных затрат машинного времени на реализацию пакета задач получаются в том случае, если задачи независимы (информационно не связаны) и равноценны по своей важности и срочности.

Виды пакетной обработки. Можно выделить следующие виды пакетной обработки в мультипрограммной ЭВМ:

- реализация задач пакета без сегментации (распараллеливания) рабочих программ, принадлежащих этому пакету;
- реализация задач пакета с сегментацией рабочих программ.

В первом случае рабочие программы, входящие в реализуемый пакет, выполняются в такой последовательности, чтобы, с одной стороны, обеспечить в первую очередь выполнение программ с наиболее высоким уровнем приоритета и, с другой стороны, затратить минимум суммарного машинного времени на выполнение всех программ пакета. При этом рабочие программы не сегментируются, что, естественно, облегчает решение задачи определения порядка их выполнения (задачи планирования загрузки ЭВМ).

Второй вид пакетной обработки отличается от первого тем, что рабочие программы, входящие в пакет, программистами предварительно разбиваются на куски, сегменты, которые могут выполняться независимо и одновременно. Поэтому при том же количестве рабочих программ в пакете задача планирования загрузки ЭВМ усложняется, зато с лучшим результатом решается задача по обеспечению минимальных затрат машинного времени на реализацию пакета. Распараллеливание рабочих программ пакета особенно эффективно в том случае, когда число программ сравнительно невелико и они резко отличаются по своим размерам.

Планирование мультипрограммной работы. После установления связности рабочих программ и их распараллеливания решается задача планирования загрузки основных устройств ЭВМ: процессора, запоминающих устройств, устройств ввода-вывода. Эта

задача называется также задачей планирования мультипрограммной работы ЭВМ.

Планирование мультипрограммной работы ЭВМ является одной из важных и самостоятельных задач организации мультипрограммной работы машины. Основная цель такого планирования состоит в том, чтобы путем лучшей организации работы машины (а не за счет использования ЭВМ более высокой производительности) повысить эффективность ее функционирования. Это повышение достигается путем создания оптимального, субоптимального или приближенного плана выполнения рабочих программ пакета, в котором определяется последовательность выполнения программ или их сегментов с учетом приоритетности и связности. В качестве основного показателя эффективности планирования мультипрограммной работы ЭВМ, как правило, принимается время $T_{\text{зпп}}$, затрачиваемое на выполнение заданного набора рабочих программ. Это время зависит не только от качества плана мультипрограммной работы машины, но и от того, насколько качественно выполнен предшествующий этап по распараллеливанию рабочих программ.

Различают неоперативное планирование мультипрограммной работы ЭВМ и оперативное. Неоперативное планирование осуществляется для выработки плана выполнения программ пакета применительно к некоторым типовым ситуациям функционирования ЭВМ. Следовательно, оно производится заранее, не в динамике работы машины. Для каждой типовой ситуации составляется свой план выполнения программ пакета, а реализация планов осуществляется в процессе функционирования ЭВМ по мере возникновения соответствующих ситуаций. Очевидно, что неоперативное планирование целесообразно проводить для ЭВМ с ограниченным перечнем решаемых задач, для которых составлены программы и известно время выполнения каждой программы на машине. Кроме того, наборы программ должны быть более или менее стабильными, а надобность в их выполнении должна возникать неоднократно, так как в противном случае не окупаются потери времени, в том числе и машинного времени, на составление планов.

Оперативное планирование проводится в динамике функционирования ЭВМ, непосредственно перед началом реализации поступившей в машину к данному моменту времени группы задач. Оно осуществляется с учетом сложившейся к этому моменту ситуации: занятости машины решением задач определенной приоритетности, наличием очередей для задач каждого приоритета, состоянием машины в смысле ее работоспособности. С учетом этих факторов оперативно составляется план реализации поступившего пакета задач, оптимальность которого оценивается величиной $T_{\text{зпп}}$. Оперативное планирование в сущности представляет собой диспетчирование, одно из основных требований к которому состоит в том, чтобы время на составление плана

(время на диспетчирование) было меньше выигрыша во времени реализации поступившего пакета задач, получаемого за счет проведения оперативного планирования. Оперативное планирование осуществляется при наличии данных о времени, необходимом на решение каждой задачи пакета на машине.

Известные методы мультипрограммного планирования различаются:

— по качеству решения задачи планирования: на методы, позволяющие получать строго оптимальный план в смысле выбранного критерия; методы, позволяющие получать субоптимальные решения; методы, позволяющие получать приближенные решения;

— по степени связности задач, для решения которых составляется план: на методы, применяемые для несвязанных задач, и методы для информационно связанных задач;

— по типу многомашинных или многопроцессорных ВС, для которых составляется план: на методы, ориентированные применительно к однородным системам (построенным на базе однотипных машин или процессоров), и методы, ориентированные применительно к неоднородным системам.

Применение методов, обеспечивающих получение строго оптимальных решений (метода целочисленного линейного программирования, метода динамического программирования и др.), как правило, связано с необходимостью выполнения большого объема вычислительной работы. По этой причине основная область их применения лежит вне динамики функционирования ЭВМ. Эти методы могут успешно применяться на отдельных этапах проектирования ЭВМ и ВС для оценки качества принимаемых конструкторских решений. В некоторых случаях, когда не накладывается жестких ограничений на время планирования и не нарушаются условия функционирования системы, эти методы могут быть положены в основу организации управления работой системы по оптимальному статическому плану.

По трудоемкости в лучшую сторону отличается метод активных расписаний, который основан на выборе оптимального расписания (т. е. оптимального плана реализации заданного набора программ) из числа активных расписаний. Активным называется такое расписание реализации ЗНП, при котором ни одна операция не начинается раньше, чем это предписано данным расписанием. Число активных расписаний значительно меньше числа всех возможных расписаний реализации ЗНП на данной системе, что облегчает решение задачи отыскания оптимального расписания. Метод активных расписаний путем последовательных оценок и приближений позволяет улучшить первоначально выбранный план реализации ЗНП. Отсутствие целенаправленного поиска оптимального решения, замененного методом проб и последовательных приближений, в общем случае не обеспечивает получения оптимального плана реализации ЗНП.

По затратам машинного времени на реализацию ЗНП и по требуемой емкости памяти метод активных расписаний имеет преимущество перед методом целочисленного линейного программирования.

Метод составления субоптимального плана с точки зрения трудоемкости и необходимой емкости памяти является более предпочтительным по сравнению с предыдущими методами. Недостаток его состоит в том, что он не позволяет получить строго оптимального плана. Он лишь обеспечивает получение одного из возможных решений, приближающегося к оптимальному.

Методы получения приближенных решений характеризуются тем, что при их использовании, хотя и не обеспечивается получение минимального значения $T_{зип}$, зато по своей трудоемкости они являются наиболее пригодными, особенно для систем, работающих в оперативном временном режиме.

Организация функционирования ЭВМ в режиме пакетной обработки под управлением ДОС ЕС. Как уже отмечалось, в мультипрограммном режиме работы дисковая операционная система обеспечивает возможность одновременного выполнения до трех независимых программ, каждая из которых размещается в своем разделе основной памяти. Для каждого из разделов может существовать свой входной поток заданий, т. е. с различных физических устройств ввода-вывода, распределенных между разделами основной памяти, одновременно могут поступать задания из трех независимых входных потоков.

Наибольшая эффективность функционирования системы достигается в случае, когда во всех разделах выполняются пакетированные задания, так как при этом простои основных устройств системы сводятся к минимуму. ДОС допускает различные варианты одновременного выполнения пакетированных заданий и одиночных программ, размещенных в разделах основной памяти так, как указано ниже.

Раздел	Вариант		
	1	2	3
Фоновый	Пакетирование		
F2	Пакетирование		Одиночная программа
F1	Пакетирование	Одиночная программа	

В исходном состоянии система настраивается на прием потока заданий в фоновом разделе, где, как правило, постоянно присутствует входной поток. Подключение к системе входных заданий или одиночных программ в разделах F2 и F1 производится опе-

ратором. Выполнение программ в режиме пакетной обработки осуществляется под управлением управляющей программы ДОС, причем одновременно могут выполняться одна фоновая программа и одна или две программы переднего плана. Поскольку ряд программ может претендовать на использование одного и того же ресурса (например, машинного времени процессора) в одно и то же время, между программами установлен приоритет, в соответствии с которым предпочтение при распределении ресурсов отдается программам первого раздела переднего плана. Самый низкий приоритет имеют фоновые программы. Следовательно, при мультипрограммной работе системы неизбежно образование очередей запросов на использование ресурсов. Когда тот или иной ресурс освобождается, он предоставляется программе из очереди старшего приоритета.

В случае прерывания (а оно наступает также и по окончании использования соответствующего ресурса) управление получает СУПЕРВИЗОР, который обрабатывает прерывание и передает управление программе с наивысшим приоритетом из числа программ, находящихся в состоянии готовности. Выполнение программы с высоким приоритетом прерывается, если появляется причина, препятствующая продолжению обработки (отсутствие очередной порции исходных данных, наличие ошибки в самой программе и т. д.). Продолжение ее выполнения возможно только после устранения этой причины. Прерывание выполнения программы с более низким приоритетом наступает всякий раз, когда устраняются причины, препятствующие продолжению (или началу) выполнения активной программы с более высоким приоритетом.

Система всегда настраивается на выполнение пакетированных заданий в фоновом разделе, т. е. после окончания очередного задания она требует поступления следующего задания. Оператор не может отменить пакетную обработку в фоновом разделе. В разделах F2 и F1 допускается обработка как пакетированных заданий, так и одиночных программ, причем при обработке пакетированных заданий соответствующий раздел переднего плана должен иметь емкость не менее 10К байт, а при обработке одиночных программ — не менее 2К байт. Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ принимает пакетированные задания для раздела F1 или F2 переднего плана после того, как оператор с пультовой пишущей машинки дал указание системе начать пакетную обработку в соответствующем разделе. После этого система будет требовать поступления следующих заданий в раздел F1 или F2 до тех пор, пока оператор не отменит пакетную обработку в этом разделе.

Включение в обработку одиночной программы осуществляется программой ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ после указания оператора, которое передается системе с пультовой пишущей машинки. После выполнения одиночной программы соответствующий раздел переднего плана становится свободным.

Для включения в обработку новой одиночной программы необходимо новое указание оператора.

Режим пакетной обработки, реализуемый под управлением ДОС, характеризуется тем, что все физические устройства ввода-вывода, подключенные к машине, распределяются между разделами. Следовательно, программа, выполняемая, например, в разделе F1 не может использовать устройства ввода-вывода, отведенные для других разделов. Программист в исходной программе указывает не конкретные физические устройства ввода-вывода, а логические устройства, и он знает лишь общее количество логических устройств, которые он может использовать. Распределение физических устройств ввода-вывода между разделами осуществляется оператором непосредственно перед выполнением программ. Поскольку при генерации ДОС для каждого раздела указывается количество логических устройств, которые могут использоваться программой этого раздела, то во время назначения логическому устройству данного раздела конкретного физического устройства одновременно происходит закрепление последнего за соответствующим разделом.

Одно и то же физическое устройство не может использоваться одновременно в разных разделах. Исключение из этого правила допускается для ВЗУ на магнитных дисках и для пишущей машинки. Различные участки одного и того же пакета дисков могут использоваться программами разных разделов.

Так как пакетная обработка требует для каждого раздела отдельных физических устройств ввода-вывода, организация такой обработки во всех разделах возможна лишь при наличии достаточно богатого набора этих устройств в комплекте машины (здесь уместно заметить, что раздел можно рассматривать не только как определенную область основной памяти, но и как совокупность области основной памяти и внешних устройств — УВВ и ВЗУ, с которыми работает программа данного раздела).

Организация функционирования ЭВМ в режиме пакетной обработки под управлением ОС ЕС. Пакетная обработка под управлением ОС возможна как в мультипрограммном режиме с фиксированным числом задач, так и в мультипрограммном режиме с переменным числом задач. В первом случае пакетированные задания формируются для фиксированных разделов основной памяти, во втором случае распределение пакетированных заданий между разделами осуществляется динамически. Управление выполнением заданий в каждом из этих случаев производится своей управляющей программой.

При пакетной обработке с фиксированным числом задач основная память разбивается на две области: системную и динамическую. Динамическая область, в свою очередь, разбивается на некоторое фиксированное число разделов (не более 52). Это число определяется во время генерации ОС или во время начальной загрузки. Для каждого раздела имеется свой входной поток за-

даний, однако в каждом разделе может находиться в данный момент лишь одна активная задача. Из общего количества разделов для выполнения заданий могут использоваться не более 15 разделов. Максимальное число одновременно выполняемых задач и подзадач не должно превышать 255, причем в это число входят как системные задачи и подзадачи, так и задачи и подзадачи абонентов.

В однопроцессорной ВС в каждый данный момент процессор занят выполнением какой-то одной команды из программы, соответствующей одной из реализуемых задач. Порядок выполнения задач, затребовавших ресурсы системы, устанавливается в соответствии с их приоритетом. Приоритет каждой задачи определяется разделом основной памяти, в котором она размещается. Переключение управления от задачи с более высоким приоритетом к задаче с более низким приоритетом осуществляется в случае, когда первая из них должна подождать окончания некоторого события, например, ввода необходимых исходных данных. Как только задача с высоким приоритетом оказывается в состоянии готовности возобновить работу, выполнение задачи низшего приоритета приостанавливается и управление возвращается к задаче высокого приоритета, выполнение которой продолжается до тех пор, пока другое событие не вынудит ее уступить управление.

Организация пакетной обработки в мультипрограммном режиме с переменным числом задач имеет различия по сравнению с режимом, когда число одновременно реализуемых задач постоянно. Различия эти особенно существенны в отношении использования основной памяти. Здесь также основная память разбивается на системную и динамическую области. Однако динамическая область, доступная для программ абонентов, разбивается не на разделы, а на зоны. Каждое задание, вызванное для выполнения, оперирует внутри определенной зоны. Пункт задания, в том числе и пакетированного, может ввести в обработку одну или несколько подзадач. Задача и порожденные ею подзадачи выполняются внутри одной и той же зоны и используют один и тот же ключ защиты памяти.

В рассматриваемом режиме работы может возникнуть такая ситуация, когда после назначения некоторой зоны для задания потребуется выделение дополнительной памяти этой зоне. Такая операция может быть проделана лишь в случае, если в системе предусмотрена процедура свертывания-развертывания, которая позволяет осуществлять временное динамическое распространение конкретного задания за пределы первоначально отведенной ему зоны. Для задания выделяется дополнительная нераспределенная память. Если же такой памяти нет, производится свертывание (перевод во вспомогательную или внешнюю память) другого задания (обычно низшего приоритета по отношению к первому заданию), его зона освобождается и используется первым заданием. После освобождения первым заданием дополнительной

области памяти она возвращается тому источнику, откуда была получена, т. е. либо в область нераспределенной памяти, либо свернутому заданию, которое вновь переводится в основную память (развертывается).

При обработке пакетированных заданий в мультипрограммном режиме работы достигается повышение пропускной способности системы за счет эффективного использования ресурсов. Для повышения пропускной способности важное значение имеет выбор приоритета заданий. Например, программы, требующие большого объема обработки, нецелесообразно наделять высоким приоритетом, так как они, монополюя процессор в течение длительного промежутка времени, могут существенно уменьшить общую пропускную способность. Снижение производительности системы происходит и в случае, когда несколько одновременно выполняемых программ используют информацию, хранимую в одном и том же внешнем запоминающем устройстве.

17.3. ПОРЯДОК ОБСЛУЖИВАНИЯ АБОНЕНТОВ ПРИ РАБОТЕ ЭВМ В РЕЖИМЕ РАЗДЕЛЕНИЯ ВРЕМЕНИ

В системах с разделением времени в типичном случае определенной программе абонента отводится некоторое машинное время (квант времени $\tau_{кв}$) системы, затем исполнение этой программы прерывается, и квант времени предоставляется программе другого абонента, затем программе третьего абонента и т. д. Далее в некоторые последующие моменты рабочие программы абонентов получают еще по одному кванту времени. Процесс продолжается до тех пор, пока каждая программа не будет выполнена полностью. Существует допустимый верхний предел числа активных абонентов, одновременно обращающихся к системе с запросами на обслуживание. Этот предел определяется производительностью ЭВМ, емкостью ее памяти, трудоемкостью реализуемых программ абонентов, допустимым временем ожидания каждой из этих программ, пропускной способностью каналов связи абонентов с ЭВМ.

В процессе работы ЭВМ в режиме деления времени обслуживаются различные группы абонентов, причем некоторым из них может предоставляться определенное преимущество при установлении очередности удовлетворения одновременно поступающих запросов. Это достигается введением системы приоритетов. Наивысший приоритет присваивается тем задачам (и соответствующим абонентам), которые должны решаться в первую очередь.

При установлении дисциплины (очередности, расписания, порядка) обслуживания запросов абонентов необходимо удовлетворить ряд требований, некоторые из которых являются противоречивыми:

— способность обслуживать срочные требования (запросы) и выполнять соответствующие программы абонентов в кратчайшее время;

— способность обслуживать требования низшего приоритета в приемлемые для абонентов сроки (во всяком случае в такие сроки, которые бы не дали повода абонентам отказаться от услуг системы);

— стремление наиболее полно загрузить центральный процессор полезной работой, т. е. выполнением программ абонентов. От того, какая принята дисциплина обслуживания требований абонентов, зависит частота переключения процессора от выполнения одной программы к другой, а значит и суммарная потеря времени на эти переключения;

— стремление уменьшить среднее время реакции СРВ и среднее число требований, ожидающих обслуживания;

— обеспечение относительной простоты реализации выбранного алгоритма определения очередности обслуживания абонентов.

Первые два требования являются взаимно противоречивыми, так как предоставление льготных условий срочным запросам осуществляется за счет запросов более низких приоритетов. И наоборот, стремление уменьшить среднее время обслуживания запросов низких приоритетов неизбежно связано (при прочих равных условиях) с необходимостью сокращения перечня запросов, относящихся к категории срочных, принадлежащих высшему приоритету. Противоречивыми являются также требование минимизации среднего времени реакции системы и среднего числа абонентов, ожидающих в очереди, с одной стороны, и требование учета степени важности и срочности выполнения отдельных программ, с другой стороны. В связи с этим при разработке алгоритма определения порядка обслуживания абонентов (иначе называемого алгоритмом диспетчирования) возникает задача нахождения компромиссного решения, удовлетворяющего в той или иной степени указанным требованиям.

Разработка оптимального алгоритма диспетчирования является сложной задачей исследования операций, требующая для своего решения привлечения методов математической статистики, теории очередей, а также учета ряда соображений инженерного характера. Оценка качества такого алгоритма производится обычно с помощью стоимостной функции, или функции штрафа за ожидание результатов вычислений.

В используемых в настоящее время СРВ нашли применение круговое (циклическое) и приоритетное обслуживание.

Круговое обслуживание представляет собой простейшую стратегию обслуживания поступивших требований, при котором предполагается, что для всех абонентов стоимость ожидания результатов выполнения их программ одинакова. Каждый абонент по очереди получает одинаковый квант времени. Если за

время одного кванта программа i -го абонента не выполняется, производится ее прерывание и переход к выполнению программы очередного абонента. При этом программа i -го абонента устанавливается в конец очереди. Формирование очереди обычно осуществляется в порядке поступления требований, т. е. по принципу «первый пришел — первый обслужен». Стратегия кругового обслуживания применяется в случае, когда для требований абонентов не устанавливается никакой дифференциации по степени их важности и срочности и не известно время на выполнение программ абонентов.

Приоритетное обслуживание представляет собой более высокую организацию обслуживания абонентов, учитывающую в большей степени их потребности.

Можно выделить следующие разновидности приоритетного обслуживания:

а) приоритетное обслуживание по принципу «чем короче программа, тем выше ее приоритет». Очередность обслуживания устанавливается в порядке возрастания величины τ_{oi} (τ_{oi} — время на обслуживание требований i -го абонента без учета потерь времени на ожидание обслуживания в очереди и на прерывания при выполнении соответствующей рабочей программы), т. е. в первую очередь выполняется наиболее «короткая» программа, требующая наименьшего объема вычислений. Если априорно распределение объема вычислений не известно, могут быть введены так называемые динамические приоритеты, которые присваиваются программам абонентов на основании предыдущих оценок программ, произведенных системой в процессе ее работы. Приоритетное обслуживание по указанному принципу применяется в случае, когда стоимость ожидания обслуживания для всех абонентов одинакова;

б) приоритетное обслуживание по так называемому правилу τ/c в порядке возрастания отношения τ_{oi}/c_i , где через c_i обозначена величина, определяющая стоимость ожидания обслуживания требования i -го абонента, отнесенная к единице времени;

в) приоритетное обслуживание по принципу «чем важнее требование, тем выше его приоритет», т. е. здесь при установлении очередности обслуживания абонентов вообще не принимается во внимание распределение объема вычислительной работы между реализуемыми программами. За основу выбирают степень важности и срочности требований абонентов.

При использовании приоритетного обслуживания абоненты по тому или иному признаку разбиваются на несколько групп, каждой из которых присваивается свой уровень приоритета. Каждому уровню приоритета соответствует своя очередь, упорядоченная в порядке поступления требований этого уровня. Приоритет требований убывает с увеличением номера уровня. Требование, которое должно обслуживаться следующим, выбирается из очереди требований наивысшего приоритета.

Приоритетное обслуживание может осуществляться либо без прерывания кванта времени, либо с прерыванием. При обслуживании без прерывания кванта переход к обслуживанию очередного требования осуществляется только в конце текущего кванта времени, поэтому возможна ситуация, когда в данный момент обслуживается требование не самого высокого приоритета.

При обслуживании с прерыванием кванта в любой момент времени осуществляется выполнение программы, принадлежащей очереди с наивысшим приоритетом.

В этом случае возможны два варианта организации вычислений:

— повторение вычислений при возврате к прерванной программе, начиная с той операции, которая соответствует началу прерванного кванта; при этом теряется часть кванта, предшествующая моменту приоритетного прерывания;

— возобновление вычислений при возврате к прерванной программе, начиная с того места программы, где эти вычисления были прерваны.

Рассмотрим конкретные примеры организации очередности обслуживания требований (программ), применяемые в существующих СРВ.

Организация очередей в соответствии с алгоритмом FB_N (FB — начальные буквы английского словосочетания Foreground Background). Алгоритм разработан применительно к однопроцессорной СРВ. Существо его заключается в следующем. Из программ, имеющихся в системе, организуется ряд очередей, которым присваиваются номера от 1 до N . Вновь прибывающие требования на обслуживание (на выполнение программ) поступают в конец очереди № 1. Выполнение программ производится в такой последовательности. Сначала выполняется программа, стоящая первой в очереди № 1. Ей представляется квант времени $\tau_{кв}$. Если за время $\tau_{кв}$ выполнение этой программы заканчивается, результаты выводятся на печать или вообще каким-то образом регистрируются и к исполнению принимается следующая программа из очереди № 1. В противном случае незавершенная программа отправляется в конец очереди № 2 и т. д. Так продолжается до тех пор, пока в очереди № 1 не останется ни одной программы. Затем таким же образом производится выполнение программы из очереди № 2 и т. д.

Первая программа из i -й очереди поступает на выполнение только в случае, когда нет очередей с номерами $i - 1, i - 2, \dots, 2, 1$. Программы из последней, N -й очереди выполняются без прерываний по принципу «первая поступила — первая обслужена».

Организация очередей в соответствии с алгоритмом $K + FB_N$, представляющим собой сочетание кругового (циклического) алгоритма и алгоритма FB_N . Алгоритм $K + FB_N$ использовался в системе с разделением времени, построенной на базе ЭВМ

AN/FSQ-32 фирмы IBM (США). В соответствии с этим алгоритмом из программ, находящихся в системе, образуются три очереди, а вновь прибывающие программы поступают в конец первой очереди. Из первой очереди программы выбираются для выполнения по принципу кругового обслуживания, причем каждая из них проходит не более трех циклов обслуживания. В каждом цикле программе предоставляется один квант времени ($\tau_{\text{кв}} = 600$ мс). Программы, на выполнение которых трех квантов оказалось недостаточным, из первой очереди отправляются в конец второй очереди.

Во второй очереди, где также используется принцип кругового обслуживания, программы выполняются в течение одного цикла, причем каждая из них получает от 1 до 6 квантов времени (здесь также $\tau_{\text{кв}} = 600$ мс). Незавершенные программы отправляются в конец третьей очереди, где они обслуживаются за один цикл. За время цикла каждой программе предоставляется до 20 квантов по 600 мс. После окончания каждого кванта времени выполнение программ из второй и третьей очередей может быть прервано, если за это время появились программы в первой очереди.

Выполнение в СРВ программ абонентов по частям с предоставлением каждой из них определенного количества квантов времени связано с заметным снижением производительности (эффективного быстродействия) центрального процессора за счет непроизводительных передач информации между оперативной памятью и ВЗУ, поскольку все активные программы, т. е. программы, требующие машинного времени, не могут быть размещены в ОЗУ. При прочих равных условиях потери времени, обусловленные обменом информацией между ОЗУ и ВЗУ, возрастают с уменьшением величины кванта, так как увеличивается число смен программ.

Один из возможных способов уменьшения числа непроизводительных передач информации между ОЗУ и ВЗУ состоит в том, что при квантовании приоритетного обслуживания требований каждому уровню приоритета ставится в соответствие своя величина кванта времени, причем с увеличением номера уровня величина кванта возрастает. В процессе обслуживания требований осуществляется перемещение их из одной очереди в другую. Если выполнение некоторой программы при предоставлении ей очередного кванта не завершено, соответствующее этой программе требование перемещается в конец очереди, принадлежащей следующему уровню приоритета с большим номером. Следовательно, требования с большим временем обслуживания получают большие значения квантов. Передвижение требования из одной очереди в другую, связанное с увеличением суммарного времени ожидания обслуживания, происходит до определенного момента. Если время ожидания требования в очереди превышает некоторую величину T , то это требование перемещается в конец очереди, принадлежащей следующему уровню приоритета с меньшим номером.

Очевидно, что такой способ сокращения числа непроизводительных передач информации неприменим в случае, когда реализуется приоритетное обслуживание по принципу «чем важнее требование, тем выше его приоритет».

Можно добиться уменьшения времени обслуживания требований за счет сокращения простоев процессора при обмене информацией между ступенями памяти СВВ путем регулирования величины кванта времени. При уменьшении числа требований величина кванта увеличивается, следовательно, уменьшается количество непроизводительных передач информации и сокращается время на обслуживание требований.

17.4. ОРГАНИЗАЦИЯ ФУНКЦИОНИРОВАНИЯ ЭВМ ЕДИНОЙ СИСТЕМЫ В РЕЖИМЕ РАЗДЕЛЕНИЯ ВРЕМЕНИ

В СВВ абоненты, использующие периферийные устройства ввода-вывода, или терминалы (например, аппаратуру абонентских пунктов), передают в систему свои формализованные запросы на обслуживание по линиям связи, которые образуют информационную магистраль абонентов. В мультиплексном канале информационная магистраль используется попеременно несколькими одновременно работающими абонентами. Реализуется временной принцип разделения, когда связь с каждым абонентом осуществляется однократно за цикл опроса всей группы активных абонентов. Этим обеспечивается более эффективное использование быстродействующей (электронной) аппаратуры мультиплексного канала. В то время, когда один терминал связан с информационной магистралью для передачи или приема информации, другие функционирующие терминалы выполняют действия, не требующие привлечения средств мультиплексного канала (например, печать ранее принятой информации). Количество одновременно работающих терминалов определяется пропускной способностью информационной магистрали, т. е. суммарная скорость передачи информации по всем активным терминалам не должна превышать пропускной способности магистрали.

В режиме разделения времени операционной системой ЭВМ решаются задачи двух типов: управление обмена информацией между абонентами и машиной и управление выполнением программ в самой машине. Требования по гибкости и адаптируемости к операционной системе, обеспечивающей такой режим работы, становятся особенно жесткими, так как состав активных абонентов, характер и приоритетность их запросов на решение задач могут значительно изменяться.

В отличие от режима пакетной обработки, когда возможно неоперативное планирование выполнения рабочих программ, в ре-

жиме разделения времени нельзя заранее предусмотреть ни времени поступления запросов, ни требуемых системных ресурсов на их удовлетворение. Тем не менее необходимо обеспечить такое обслуживание абонентов, при котором у каждого из них создается впечатление, будто он один работает с машиной. Следовательно, управляющая программа операционной системы должна осуществлять динамическое распределение ресурсов СРВ, которое постоянно меняется в зависимости от плотности запросов и их приоритетов.

С точки зрения учета машинного времени и организации функционирования СРВ независимой единицей работы является задание. Задания описываются с помощью управляющих предложений, включающих несколько типов операторов. Этими операторами определяются конкретные характеристики задания: опознаватели задания; его приоритет; временные и другие условия выполнения; рабочие программы, подлежащие выполнению; названия и характеристики используемых или образуемых массивов информации; количество и типы устройств ввода-вывода, требуемых для выполнения задания.

Для режима разделения времени важной процедурой является хронометрирование времени центрального процессора, которое используется некоторым заданием, выполняемым по запросу абонента. Хронометрирование времени основано на использовании интервального таймера. Оно необходимо для своевременного переключения процессора на выполнение другой программы, реализация которой предусмотрена по запросу другого абонента в соответствии с принятой дисциплиной обслуживания абонентов, а также для определения общего времени, затрачиваемого на обслуживание каждого запроса.

Системы с разделением времени на базе ЕС ЭВМ создаются с таким расчетом, чтобы при их использовании в определенные промежутки времени мог преобладать какой-то один режим функционирования; режим разделения времени, режим пакетной обработки, режим пакетной обработки с ограниченным временем ответа, режим запрос-ответ, разговорный режим.

В отличие от обычного режима пакетной обработки, когда время выдачи результатов решения задачи может быть отнесено, вообще говоря, сколь угодно далеко, в режиме пакетной обработки с ограниченным временем ответа это время определяется временем поступления задачи, но разница между ними существенно превышает время, затрачиваемое на решение этой задачи.

В разговорном режиме число абонентов, имеющих прямой доступ к центральному процессору системы, ограничен. В их распоряжение предоставляется один язык программирования. При работе в этом режиме программы абонентов постоянно хранятся в основной памяти системы. Центральный процессор играет роль либо активного партнера, либо посредника во взаимодействии абонентов друг с другом.

Система с разделением времени работает в реальном масштабе времени, если для решаемых по запросам абонентов задач удовлетворяется следующее временное соотношение:

$$\tau_i < \tau_i^{\text{доп}} \quad (i = \overline{1, m}),$$

где m — количество задач;

τ_i — временная характеристика i -й задачи;

$\tau_i^{\text{доп}}$ — предельно допустимое значение τ_i .

Для организации работы СРВ в реальном масштабе времени используется соответствующая управляющая программа операционной системы.

ПОСТРОЕНИЕ МНОГОПРОЦЕССОРНЫХ И МНОГОМАШИННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА БАЗЕ ЕС ЭВМ

18.1. ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ МНОГОПРОЦЕССОРНЫХ И МНОГОМАШИННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Многопроцессорная (многомашинная) вычислительная система (ВС) представляет собой совокупность взаимосвязанных и согласованно действующих однотипных или неоднотипных процессоров (машин), различных устройств, а также программных средств (математического обеспечения ВС), обеспечивающих автоматизацию процессов приема исходной информации, ее обработки и выдачи результатов обработки потребителям или на аппаратуру регистрации и документирования. В настоящей главе речь будет идти лишь о многопроцессорных и многомашинных ВС, поэтому для краткости слова «многопроцессорная», «многомашинная» в дальнейшем будем, как правило, опускать и использовать их только по мере необходимости.

Создание ВС — наиболее реальный путь разрешения противоречия между непрерывно растущими потребностями в быстродействующих и надежных средствах вычислений и пределом технических возможностей ЭВМ на данном этапе их развития. Вычислительные системы представляют собой логическое следствие эволюции двух общих тенденций, проявляющихся при создании ЭВМ, — стремления к модульной конструкции и стремления к организации параллельной работы различных устройств машины. Первая тенденция связана с повышением надежности и гибкости машины, а вторая — с повышением ее производительности (под гибкостью машины или системы понимается способность ее перестраивать свою работу в соответствии с изменениями условий эксплуатации, перечня решаемых задач, плотности потоков заявок на решение этих задач, сохраняя при этом некоторый уровень эффективности; гибкость ВС обеспечивается ее способностью к оперативному перераспределению объема вычислительной работы между элементами системы).

Вычислительные системы относятся к категории сложных систем. Наиболее существенными признаками сложных систем принято считать следующие:

— наличие единой цели функционирования для всей системы (например, обеспечение максимума пропускной способности ВС);

— сложность функции, реализуемой системой и направленной на достижение заданной цели функционирования;

— большое количество взаимно связанных и взаимодействующих элементов, составляющих систему;

— возможность деления системы на подсистемы, цели функционирования которых подчинены общей цели функционирования всей системы;

— иерархическая структура связей подсистем и иерархия критериев качества функционирования системы;

— наличие взаимодействия с внешней средой и функционирование в условиях воздействия случайных факторов;

— наличие управления в системе и высокая степень его автоматизации;

— устойчивость к воздействию внешних и внутренних возмущающих факторов (помех) и наличие самоорганизации;

— надежность системы в целом, построенной из ненадежных элементов.

Разумеется, в различных конкретных системах эти признаки представлены не в одинаковой степени.

Одной из характерных черт сложных систем вообще и вычислительных систем в частности является отсутствие единого критерия эффективности системы и наличие нескольких более или менее равнозначных критериев, каждый из которых может стать главным, преобладающим в зависимости от назначения, характера решаемых задач и состояния системы.

Современные ВС строятся на базе мультипрограммных машин, т. е. таких машин, в которых реализован принцип мультипрограммирования. В связи с этим и сами системы можно назвать мультипрограммными. Они могут и должны работать в различных мультипрограммных режимах.

Создание мультипрограммных ВС преследует две главных цели: 1) свести к минимуму время на выполнение некоторого объема вычислительных работ; 2) предоставить максимум удобств абонентам, в частности освободить их от неудобств и затруднений, связанных с необходимостью совместной и одновременной реализации нескольких программ.

Для выполнения этих целей необходимо, чтобы мультипрограммные ВС удовлетворяли ряду требований. Основные из них следующие:

— возможность использования алгоритмических языков, обеспечивающих простоту и удобство при подготовке рабочих программ, реализуемых системой;

— независимость работы различных абонентов при подготовке своих программ.

— обеспечение защиты программы данного абонента от нежелательного вмешательства со стороны программ или команд других абонентов;

— обеспечение гибкого распределения памяти и машинного

времени ВС. Управление распределением памяти и времени должно производиться исходя из потребностей реализуемых программ, в частности, исходя из заранее установленной приоритетности для этих программ;

— автоматическое управление процессом выполнения различных программ (общесистемных управляющих и обслуживающих программ, рабочих программ абонентов);

— обеспечение доступа абонентов к стандартным программам и подпрограммам, необходимым при выполнении своих рабочих программ.

Классификация ВС. Укажем прежде всего на различия между многопроцессорными и многомашиными вычислительными системами. Многопроцессорная система строится либо на базе одной многопроцессорной машины, либо на базе нескольких однотипных или неоднотипных процессоров. Многомашинальная ВС строится на базе нескольких машин, каждая из которых может быть как однопроцессорной, так и многопроцессорной. В связи с этим многомашинальные системы, как правило, являются более сложными по сравнению с многопроцессорными.

Существует достаточно большое количество различных признаков, по которым можно классифицировать ВС: по назначению, по типу процессоров или машин системы, по способу управления элементами ВС, по степени разобщенности основных элементов системы и т. д. На рис. 18.1 приведена классификационная схема ВС.

По назначению ВС делятся на универсальные (УВС) и специализированные (СВС). Универсальные системы предназначены для решения широкого круга задач различного характера, перечень которых заранее (до создания системы) не ограничен. Специализированные системы служат для решения определенного круга задач. Специализация СВС может определяться не только характером решаемых задач и ограниченностью их перечня, но и наличием в системе элементов специального назначения, например, таких, как устройства сопряжения с радиотехническими станциями, являющимися источниками информации, устройства отображения результатов обработки информации (светопланы, световые табло и др.).

По типу машин или процессоров, из которых комплектуется ВС, системы могут быть однородными, включающими однотипные

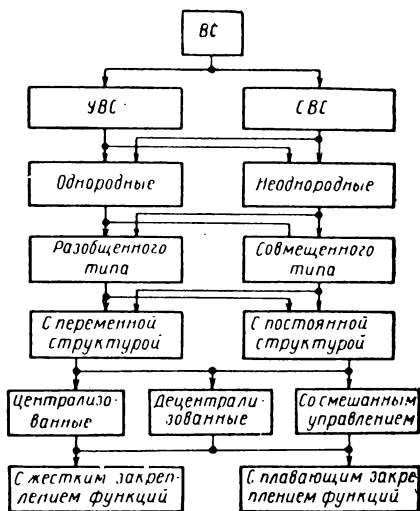


Рис. 18.1. Классификация вычислительных систем

машины (процессоры), и неоднородными, составленными из разнотипных машин (процессоров).

По степени территориальной разобщенности основных элементов многомашинные ВС разделяются на два типа. К первому типу (системы совмещенного типа) относятся системы, в которых время передачи информации от одной машины к другой пренебрежимо мало по сравнению с временем на решение задачи на одной из машин. Передача информации по линиям связи между машинами осуществляется, как правило, в параллельном коде.

Ко второму типу (ВС разобщенного типа) относятся системы, в которых время передачи информации по каналам связи между машинами соизмеримо со временем решения задач машинами системы и должно учитываться при исследовании процесса функционирования ВС. Передача информации между машинами обычно производится в последовательном коде.

Многопроцессорные системы, как правило, строятся совмещенного типа.

По постоянству структуры системы разделяются на ВС с постоянной и ВС с переменной структурой. Под структурой ВС понимается состав системы и схемы функциональных и управляющих связей между ее элементами. Две машины системы называются функционально связанными, если от одной из них к другой возможна передача функциональной информации (программ, исходных и промежуточных данных решаемых задач). Если передача функциональной информации осуществляется только в одну сторону, то ВС называются ориентированными по функциональным связям, если в обе стороны — неориентированными. Могут быть и частично ориентированные системы, в которых присутствуют как ориентированные, так и неориентированные подсистемы.

Две машины называются связанными по управлению, если от одной из них (управляющей) к другой (управляемой) поступает управляющая информация, необходимая для правильного функционирования ВС как единой системы. Если связи по управлению всегда ориентированы в одну сторону, система называется направленной, в противном случае — ненаправленной.

Каналы связи для передачи этих двух видов информации (функциональной и управляющей) могут быть как совмещенными, так и отдельными.

Система относится к типу ВС с постоянной структурой, если в процессе ее функционирования состав функциональных и управляющих связей и их ориентация остаются неизменными. В противном случае система называется ВС с переменной структурой.

По степени централизации управления ВС разделяются на централизованные, децентрализованные и системы со смешанным управлением.

В централизованных ВС осуществлена полная централизация управления элементами системы, т. е. все функции управления

сосредоточены в одном элементе, в качестве которого используется одна из машин (процессоров), называемая машиной-директором. Машина-директор координирует загрузку машин системы и их взаимодействие в процессе решения задач.

В децентрализованных системах функции управления распределены между главными и периферийными центрами (элементами). В этом случае структура средств управления обычно является иерархической, причем может быть несколько уровней управления. Например, при двух уровнях управления система разбивается на группы взаимодействующих элементов с локальным управлением в пределах каждой группы (это первый уровень управления), на втором уровне управления с помощью некоторого управляющего элемента обеспечивается взаимодействие выделенных групп элементов системы.

В системах со смешанным управлением в некоторой пропорции реализованы принципы централизованного и децентрализованного управления.

Централизованные ВС называются также одноуровневыми, а децентрализованные системы с иерархической структурой средств управления — многоуровневыми.

По принципу закрепления вычислительных функций за отдельными машинами (процессорами) различают системы с жестким и системы с плавающим закреплением функций. Более распространенным является принцип плавающего закрепления вычислительных функций за отдельными машинами. Жесткое закрепление, хотя и позволяет использовать для хранения программ надежно работающие ПЗУ, приводит к значительному уменьшению гибкости системы. В связи с этим системы с жестким закреплением функций за вычислительными средствами строятся редко, они могут быть только специализированными.

Имеются и другие признаки, по которым можно классифицировать ВС, например, принцип действия машин системы (цифровые и аналого-цифровые ВС), наличие или отсутствие общего поля оперативной памяти и т. д.

Уровни комплексирования машин Единой системы. В структуре моделей ЕС ЭВМ и в их математическом обеспечении заложена возможность объединения нескольких ЭВМ в единую вычислительную систему, называемую мультисистемой. Основные цели организации таких ВС состоят в достижении необходимой вычислительной мощности (производительности) за счет более эффективного использования вычислительных средств, в повышении надежности как за счет дублирования, так и за счет возможности быстрой локализации и устранения неисправностей, в улучшении использования аппаратуры ЭВМ путем перераспределения нагрузок между отдельными устройствами.

Мультисистема строится из логически независимых компонентов: процессоров, устройств оперативной памяти, мультиплексных и селекторных каналов, ВЗУ, устройств управления ВЗУ,

устройств ввода-вывода, устройств управления УВВ. Логическая независимость процессоров системы определяется возможностью их независимого функционирования. Для остальных компонентов эта независимость определяется возможностью их функционального подсоединения к одному или к нескольким процессорам ВС. На базе моделей ЕС ЭВМ предполагается в первую очередь построение ВС, состоящих из двух процессоров.

Связь между машинами (процессорами) мультисистемы может осуществляться на уровне любого из ее логически независимых компонентов с помощью специальных мультисистемных средств. В зависимости от этого различают следующие уровни комплексирования вычислительных средств;

- на уровне каналов с помощью адаптера канал — канал;
- на уровне устройств управления ВЗУ или УВВ с помощью двухканального переключателя;
- на уровне ВЗУ или УВВ с помощью переключателя на два направления;
- на уровне аппаратуры передачи данных с помощью специальных устройств управления дистанционной связью;
- на уровне общей оперативной памяти (при использовании многоходовых блоков памяти);
- на уровне процессоров с помощью средств прямого управления.

Естественно, что в зависимости от используемого уровня комплексирования, т. е. от типа связи между моделями машин Единой системы, эффективность функционирования ВС будет различной. Наиболее совершенной будет ВС с полными связями, в которой реализованы все уровни комплексирования. Однако мультисистема с полными связями получается весьма сложной как по своей структуре, так и по необходимому математическому обеспечению. Все модели ЕС ЭВМ обеспечивают построение мультисистем с комплексированием на уровне каналов и (или) общих внешних устройств (ВЗУ или УВВ). Оборудование оперативной памяти моделей ЕС-1030 и ЕС-1050 может распределяться между процессорами мультисистемы, образуя общее поле памяти.

Необходимо отметить, что высокая степень адаптивности ВС достигается не только наличием развитой системы связи между ее логически независимыми компонентами, наличием соответствующих устройств обмена информацией, но и иерархическим построением запоминающих устройств.

В системах, построенных на базе ЕС ЭВМ, устанавливается следующая иерархия памяти:

- память процессоров — это сверхоперативная память небольшого объема (матрица общих регистров);
- основная оперативная память — это быстродействующая память модульной структуры с независимым обращением к модулям;

— большая оперативная память — это сравнительно недорогая память модульной структуры на ферритовых сердечниках со средним быстродействием и с независимым обращением к модулям;

— внешняя память — это ВЗУ большого объема на магнитных дисках (реже барабанах) с достаточно малым временем выборки информации;

— архивная память — это ВЗУ практически неограниченного объема на магнитных лентах, магнитных картах.

При такой иерархии памяти информация может быть размещена по ее уровням следующим образом:

— в основной оперативной памяти: основные компоненты управляющей программы операционной системы, исполняемые рабочие программы и необходимые исходные данные по решаемым задачам, обслуживающие программы;

— в большой оперативной памяти: обслуживающие программы операционной системы, прерванные рабочие программы и исходные данные к ним, программы обслуживания ввода-вывода;

— во внешней памяти на магнитных дисках (барабанах): копии основных компонентов управляющей программы операционной системы, диагностические программы, очереди задач, стандартные программы;

— в архивной памяти ВС: копия операционной системы, типовые программы, архивные данные, библиотека стандартных программ.

В мультисистеме основная цель, которая преследуется при организации связи между ее компонентами на любом уровне, заключается в обеспечении использования общих информационных массивов и программ взаимодействующими процессорами системы. Наиболее гибкой и быстрой является связь процессоров через общее поле оперативной памяти. Такая связь позволяет увеличить вычислительную мощность мультисистемы и эффективность использования оперативной памяти, а также получить наиболее организованную мультисистему, работающую под управлением единой операционной системы.

Мультисистемная связь двух процессоров по прямому управлению (комплексирование на уровне процессоров) предназначена преимущественно для передачи между процессорами сигналов внешних прерываний и управляющей информации с целью синхронизации в мультисистеме единого вычислительного процесса.

Требования к мультипроцессорным ВС. При построении и организации функционирования многопроцессорных систем необходимо выполнить ряд требований, основные из которых следующие.

1. Процессоры должны быть функционально идентичны, т. е. взаимозаменяемы по своим функциям. СУПЕРВИЗОР операционной системы при распределении задач между процессорами

должен принимать во внимание занятость процессоров, их возможности по производительности, но никак не то, может ли данный процессор решать данную задачу или нет.

2. Система должна обеспечивать мультипрограммную реализацию независимых рабочих программ (или пакетов программ), каждая из которых может представлять собой комплекс зависимых программ, выполняемых совместно.

3. Система должна быть в работоспособном состоянии до тех пор, пока хотя бы один процессор системы продолжает работать. Из этого общего требования вытекает ряд частных требований, а именно:

— СУПЕРВИЗОР должен быть постоянно связан со всеми процессорами; во всяком случае он не должен быть постоянно связан только с каким-то одним процессором или с группой процессоров;

— каждый процессор должен иметь возможность начать операцию ввода-вывода через любой из каналов;

— в каждом процессоре должна быть заложена возможность реакции на всевозможные прерывания. Прерывания, проявляющиеся внутри данного процессора, обычно обрабатываются этим же процессором;

— рабочие программы должны быть составлены так, чтобы правильность и точность их выполнения не зависели от того, какие и сколько процессоров выполняют эти программы.

4. Система должна строиться таким образом, чтобы допускалось двумя и более процессорами одновременное выполнение одной и той же последовательности команд, используя различные группы данных. Такая необходимость возникает при обработке больших массивов информации или исходных данных. В этом случае более простой и выгодной является не сегментация программ, а сегментация данных: весь массив исходных данных разбивается на части, каждая часть обрабатывается на отдельном процессоре, но для всех частей необходима одна и та же последовательность команд, которая в единственном экземпляре хранится в оперативной памяти системы.

5. В системе должна быть исключена возможность нескольким процессорам одновременно изменять в оперативной памяти ВС одну и ту же величину. Для этого предусматривается временная блокировка обращений процессоров к избранным ячейкам памяти или блокировка входа более чем одного процессора в избранные части программы.

6. В системе должно обеспечиваться быстрое выполнение срочных программ. Для этого организуется приоритетная система, реализуемая управляющей программой операционной системы.

7. СУПЕРВИЗОР должен быть всегда заблокирован, т. е. в любой момент времени только один процессор может его выполнять.

Режимы работы ВС. В вычислительных системах, построенных на базе ЕС ЭВМ и укомплектованных техническими средствами для телеобработки информации и соответствующим математическим обеспечением, предусматриваются различные режимы работы, такие как однопрограммный режим, режим пакетной обработки, режимы телеобработки.

При работе ВС в однопрограммном режиме каждая машина системы (или каждый процессор) работает в однопрограммном режиме с последовательным выполнением программ без участия абонентов (в режиме последовательной обработки). Естественно, что при работе в таком режиме возможности системы используются далеко не полностью.

В режиме пакетной обработки осуществляется параллельное выполнение машинами (процессорами) системы пакетированных рабочих программ. Программы, включенные в один и тот же пакет, могут выполняться разными машинами системы. Распределение пакетированных программ между машинами производится как заранее, не в динамике функционирования системы, что характерно для неоперативного планирования мультипрограммной работы ВС, так и в динамике работы ВС, непосредственно перед началом реализации поступившей в систему к данному моменту времени группы задач (это характерно для оперативного планирования). Основная цель организации режима пакетной обработки ВС остается той же, которая была указана применительно к отдельной машине, а именно: обеспечение минимальной затраты машинного времени на выполнение заданного набора программ за счет повышения коэффициента использования каждого устройства машин ВС и, кроме того, за счет одновременной реализации программ набора на всех машинах системы.

Режимы телеобработки являются наиболее сложными, требующими соответствующего технического оснащения и математического обеспечения. В вычислительных системах предусматривается реализация всех видов телеобработки, указанных в гл. 1 (см. п. 1.5).

Выбор того или иного способа организации вычислительного процесса ВС (режима работы) зависит от степени однородности центральных вычислительных средств системы, способа закрепления функций за ЭВМ ВС, уровня комплексирования вычислительных средств системы, способа реализации прерывания программ, способа распределения памяти и внешних устройств, способа организации управления в системе, наличия или отсутствия резерва времени.

Как указывалось выше, в однородных системах центральные вычислительные средства включают только однотипные ЭВМ. Степень однородности ВС предопределяет прежде всего выбор того или иного метода планирования мультипрограммной работы вычислительных средств системы. Естественно, что в однородных системах решение задачи планирования мультипрограммной

работы осуществляется значительно проще, чем в неоднородных системах.

В системах с жестким закреплением функций за ЭВМ организация функционирования производится с меньшими потерями времени на планирование работы ВС. Это объясняется главным образом тем, что возникающие в процессе функционирования системы ситуации являются более или менее типовыми, для которых планирование мультипрограммной работы ВС может выполняться заблаговременно. С большими потерями времени на работу операционной системы и более сложной оказывается процедура организации работы ВС с плавающим закреплением функций за ЭВМ.

Уровень комплексирования вычислительных средств системы в значительной степени определяет ее гибкость и затраты времени на обмен информацией между машинами ВС. Это, в свою очередь, сказывается на выборе способа организации совместной работы машин системы. Если, например, комплексирование осуществляется на уровне канал—канал или на уровне внешних запоминающих устройств, когда на обмен информацией между ЭВМ затрачивается сравнительно большое время, целесообразно выбрать такой метод планирования мультипрограммной работы ВС и так организовать процесс ее функционирования, чтобы сократить до минимума число обменов информацией.

В современных мультипрограммных ЭВМ используется сочетание программных и аппаратных средств прерывания.

Обычно аппаратная система прерывания фиксирует сигналы прерывания от устройств ввода-вывода. В такой системе прерывание исполняемой рабочей программы производится за счет комплекса аппаратных средств, которые без предварительного обращения в память ЭВМ для вызова программы прерывания останавливают вычисления и с помощью микропрограммного аппарата производят операции по записи результатов решения и начальных команд.

При использовании программной системы прерывания с помощью специальной схемы определения приоритета осуществляется сравнение приоритетов исполняемой программы и поступившей. Если приоритет поступившей программы выше, чем исполняемой, вырабатывается сигнал на прерывание, который воспринимается устройством управления процессора. Устройство управления обращается в память машины за соответствующей программой прерывания, при выполнении которой производится остановка вычислений, запись результатов решения и команд, с которых следует продолжать дальнейшее решение задачи, обращение в память для вызова поступившей программы с более высоким приоритетом и т. д. Разумеется, это не единственная схема работы программной системы прерывания.

Чем больше удельный вес аппаратных средств в общей программно-аппаратной системе прерывания, тем, при прочих равных условиях, меньше затрачивается времени на работу этой системы,

т. е. на переключение ЭВМ с решения одной задачи на решение другой. Следовательно, при организации вычислительного процесса необходимо принимать во внимание эти затраты, иначе говоря, учитывать структуру системы прерывания. Если в этой системе преобладают программные средства, предпочтительнее в ВС иметь меньшее количество различных урвней приоритета как для задач (и соответствующих программ), так и для устройств.

Как известно, в мультипрограммных ЭВМ ВС выделяются два основных способа распределения памяти и внешних устройств: постоянное распределение и динамическое. При постоянном распределении память и внешние устройства распределяются между программами при их вводе. При динамическом распределении выделение этих средств программам осуществляется при включении их в фазу выполнения, поэтому заранее не производится присвоения действительных адресов памяти, определения адресов различных устройств, необходимых данной программе. Динамическое распределение является более распространенным, так как оно обеспечивает наибольшую гибкость и экономичность в расходовании средств для каждой выполняемой программы.

Естественно, что при организации работы ЭВМ ВС, т. е. при разработке математического обеспечения, должен учитываться реализуемый способ распределения памяти и внешних устройств.

Организация вычислительного процесса в ВС существенно зависит от степени централизации управления в системе. Особенности и различия в организации функционирования централизованных и децентрализованных систем телеобработки рассматриваются в работе [22].

Что касается наличия или отсутствия резерва времени, то этот фактор сказывается прежде всего на выборе метода планирования мультипрограммной работы вычислительных средств системы. При отсутствии резерва времени выбираются такие методы, которые не требуют сколько-нибудь значительных затрат машинного времени системы на составление плана ее работы, хотя получаемый при этом план не является оптимальным в смысле минимума времени на реализацию заданного набора задач.

18.2. СТРУКТУРЫ МНОГОМАШИНЫХ И МНОГОПРОЦЕССОРНЫХ ВС НА БАЗЕ ЭВМ ЕДИНОЙ СИСТЕМЫ

Основными факторами, оказывающими влияние на структуру многомашинных и многопроцессорных вычислительных систем, являются следующие:

— режимы работы, которые должны быть реализованы в системе. Если, например, не предусматривается работа системы в таких режимах, которые связаны с обслуживанием удаленных абонентов (режим разделения времени, режим запрос-ответ и др.), то структура ВС будет существенно проще. Реализация же подоб-

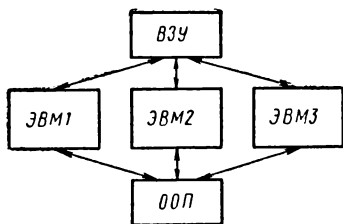


Рис. 18.2. Вычислительная система с общей памятью

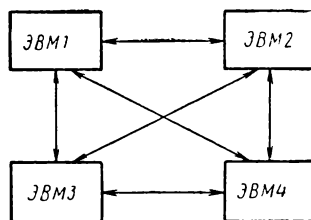


Рис. 18.3. Децентрализованная ВС

ных режимов сопряжена с необходимостью иметь разветвленную сеть внешних абонентских пунктов, аппаратуру передачи данных, мультиплексоры передачи данных, линии связи большой протяженности;

- способ объединения машин (процессоров) в системе или способ обмена информацией между машинами (уровень комплексования ЕС ЭВМ);

- принятый уровень централизации управления ВС;

- назначение ВС и характер решаемых задач (поскольку в зависимости от этого системы будут универсальными или специализированными, отличающимися по своей структуре в той или иной степени).

В настоящее время распространенным является вариант структуры ВС с общей оперативной памятью (ООП) и общими внешними запоминающими устройствами (рис. 18.2). В таких системах программы решаемых задач, исходная информация и промежуточные данные хранятся в общей системной памяти (ООП и ВЗУ), которая доступна всем ЭВМ ВС. Обмен информацией между машинами осуществляется через ООП. Вычислительные системы с общей памятью могут быть как централизованными, так и децентрализованными. Обычно они являются системами совмещенного типа.

На рис. 18.3 приведен пример структуры децентрализованной ВС без общих запоминающих устройств. ЭВМ в системе связаны по принципу «каждая с каждой». Каждая машина ВС имеет свои автономные средства для хранения информации — ОЗУ и ВЗУ. Обмен информацией между машинами осуществляется через автономную оперативную память ЭВМ. Такие системы могут быть как совмещенного, так и разобщенного типа.

Более совершенная структура системы приведена на рис. 18.4. Это ВС с общей памятью (оперативной и внешней) и с непосредственной связью между машинами, т. е. принцип связи «каждая с каждой» здесь также реализован. Кроме общей оперативной памяти, каждая машина имеет автономное ОЗУ. Обмен информацией между машинами осуществляется либо напрямую через свои

ОЗУ, либо через ОПП с помощью коммутатора. По степени централизации управления такие системы могут быть централизованными и со смешанным управлением.

В децентрализованной ВС с иерархической структурой средств управления (рис. 18.5) можно выделить несколько уровней управления, в данном случае два уровня. На нижнем (первом) уровне функции управления сосредоточены в управляющих ЭВМ вычислительных систем ВС1, ВС2, ВС3 каждая из которых объединяет некоторое количество вычислительных средств. Каждая из этих систем по степени централизации управления является централизованной ВС, структура которой может быть самой различной, например, такой, как приведена на рис. 18.4. Взаимодействие вычислительных систем ВС1, ВС2, ВС3 (а точнее, управляющих ЭВМ этих систем) производится с помощью управляющей ЭВМ вычислительной системы ВС4. Это высший (второй) уровень управления.

Рассмотренные структуры систем являются типовыми для универсальных ВС. В существующих в настоящее время УВС в той или иной мере реализованы описанные структурные принципы.

Структуры специализированных вычислительных систем (СВС) отличаются в той или иной степени от структур УВС в зависимости от характера решаемых задач. Отметим наиболее характерные особенности некоторых проектов структур СВС, предназначенных для реализации массового распараллеливания частей вычислительного процесса.

Система Унгера предназначена для решения задач типа распознавания зрительных образцов. Отличительная особенность таких задач состоит в том, что при их решении необходимо выполнять ограниченный перечень одних и тех же операций над большими массивами чисел, задаваемых в виде матриц. Результаты получаются также в виде матриц, причем для образования данного результата используются числа, находящиеся в соседних ячейках матрицы.

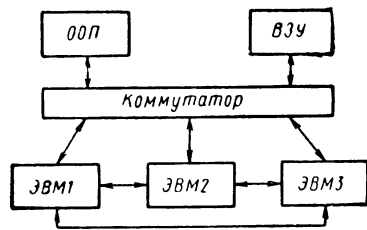


Рис. 18.4. Вычислительная система с общей памятью и непосредственной связью между машинами

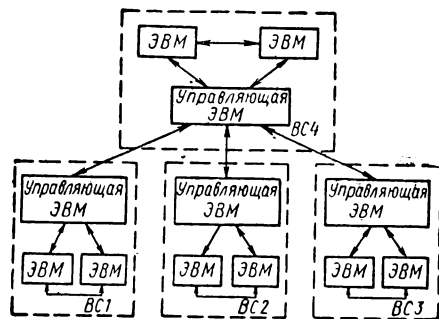


Рис. 18.5. Децентрализованная ВС с иерархической структурой средств управления

Основными элементами проекта системы Унгера являются центральное устройство управления (это ЭВМ с памятью большой емкости) и большое число одинаковых логических модулей, расположенных в узлах однородной сетки. Таким образом, модули образуют однородную по функциональным связям структуру, каждый модуль которой соединен с четырьмя такими же соседними модулями и, кроме того, с центральным устройством управления и внешним источником информации. Число модулей может изменяться от нескольких сотен до нескольких десятков тысяч без изменения логической структуры системы. Конструктивно каждый модуль представляет собой одноразрядный двоичный сумматор накапливающего типа с соответствующими логическими элементами и одноразрядными запоминающими ячейками, обеспечивающими выполнение заданного набора элементарных операций. В ходе решения задач все логические модули одновременно выполняют одну и ту же операцию.

Область применения СВС подобного типа ограничивается небольшим кругом задач, допускающих при их решении представление вычислительного процесса в виде большого числа **одинаковых** параллельных ветвей. Ограничения эти усугубляются такими особенностями системы, как невозможность исключения из выполнения операции каких-либо модулей СВС (все модули должны быть активными), малая разрядность сумматоров модулей, небольшой набор операций, выполняемых модулями.

Система Соломон предназначена для решения задач, допускающих одновременное выполнение одной и той же операции над множеством чисел (обращение матриц, решение систем линейных уравнений, вычисление корреляционных функций и др.). При решении этих задач вычислительный процесс может быть организован также в виде большого числа одинаковых параллельных ветвей, но в отличие от задач распознавания зрительных образов необходимо выполнять гораздо больший набор операций с многоразрядными числами и, кроме того, необходима память большей емкости для хранения исходных данных, промежуточных и окончательных результатов.

В конструктивном отношении система Соломон представляет собой набор из 1024 однородных и одинаково соединенных модулей, образующих сетку. Все модули связаны с центральным устройством управления, каждый из них состоит из арифметического устройства последовательного действия и двух блоков ОЗУ емкостью на 2048 бит каждый. В один и тот же момент все модули могут выполнять одну операцию над числами, хранящимися в их ячейках ОЗУ с одними и теми же адресами. Но не все модули должны быть активными, часть из них может оставаться в пассивном состоянии, что определяется командами центрального устройства управления. По сравнению с проектом системы Унгера система Соломон имеет более высокую производительность главным образом за счет экономии времени при обра-

щении к ОЗУ, хотя это достигается существенным усложнением системы.

Система Холланда предназначена для решения задач, допускающих представление вычислительного процесса в виде большого числа как одинаковых, так и различных параллельных ветвей. В такой системе одновременно может выполняться не одна, а множество различных программ. Система Холланда представляет собой однородную сетку из одинаковых модулей, каждый из которых выполняет не только функции выполнения операций, но и функции управления ходом вычислений. В каждом такте вычислений число активных модулей может быть произвольным, что позволяет одновременно выполнять несколько различных программ. Модули достаточно просты, так как емкость их памяти невелика, а набор выполняемых операций ограничен.

Указанные СВС могут быть названы проблемно-ориентированными, так как они предназначены для решения задач определенного класса или ориентированы на решение определенных проблем. Это позволяет упростить конструкцию основных элементов СВС и связей между ними.

Необходимо отметить, что с помощью проблемно-ориентированных СВС задача повышения производительности вычислительных средств решается только при реализации того класса задач, на который эти системы рассчитаны. С появлением новой проблемы, нового класса задач необходимо разрабатывать новую СВС, поэтому создание проблемно ориентированных систем — далеко не лучший путь решения задачи повышения производительности вычислительных средств.

До сих пор рассматривались структуры многомашинных и многопроцессорных систем, в которых не предусматривалось оперативное и непосредственно взаимодействие удаленных абонентов с ВС. Ниже приводятся варианты структур многопроцессорных систем с разделением времени.

Структура многопроцессорной СРВ определяется рядом факторов; наиболее существенными из них являются: число процессоров в системе, организация связи терминальной аппаратуры (периферийных абонентских пунктов) с основной оперативной памятью системы, организация связи между основными элементами СРВ, форма эксплуатации системы.

В многопроцессорных СРВ, в отличие от однопроцессорных, имеются более широкие возможности по организации различных дублирующих связей и включению дополнительных элементов, обеспечивающих повышение надежности функционирования системы

Структура многопроцессорной СРВ может быть той или иной в зависимости от организации симплексной или мультиплексной связи между абонентскими пунктами и процессорами системы. Такие связи характерны для многопроцессорных СРВ с иерархической структурой.

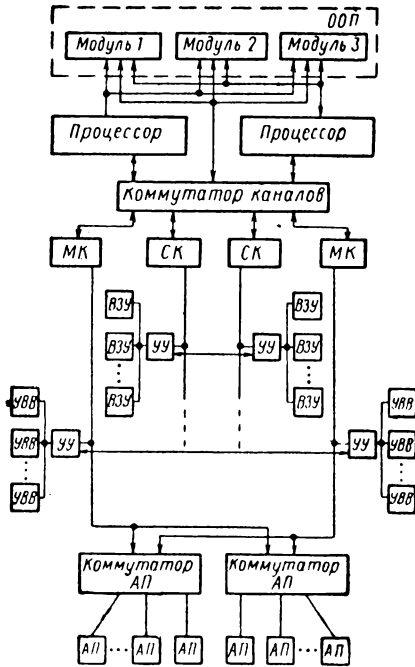


Рис. 18.6. Двухпроцессорная СРВ

характеризуется тем, что в ней связь между элементами из различных уровней (например, между абонентскими пунктами $АП_{11}, АП_{12} \dots, АП_{1n}$, с одной стороны, и $МК_1$, с другой,

Один из возможных вариантов структуры двухпроцессорной СРВ приведен на рис. 18.6. В системе имеется общее поле оперативной памяти. Кроме того, устройства управления ВЗУ и УВВ связаны между собой с помощью двойных переключателей. Абонентские пункты АП связаны с мультиплексными каналами МК через коммутаторы абонентских пунктов, функции которых могут выполнять устройства управления АП.

Иерархические структуры многопроцессорных СРВ изображены на рис. 18.7 и 18.8, где обозначено: $П_1, П_2 \dots$ — процессоры; $К_1, К_2 \dots$ — коммутаторы каналов; $МК_1, МК_2 \dots$ — мультиплексные каналы; $КТ_1, КТ_2 \dots$ — коммутаторы терминалов (абонентских пунктов). Система с симплексными связями (рис. 18.7)

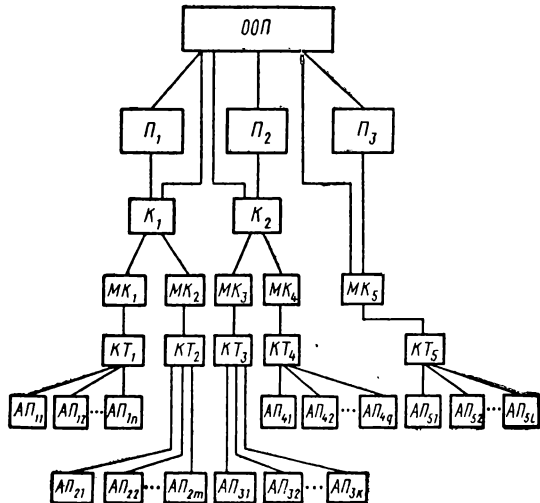


Рис. 18.7. Иерархическая структура многопроцессорной СРВ с симплексными связями

или между $МК_1, МК_2$, с одной стороны, и процессором $П_1$, с другой, может быть осуществлена только одним способом. Каждый коммутатор здесь осуществляет одноканальную связь между двумя группами элементов, которые он соединяет.

В системе с мультиплексными связями (рис. 18.8) связь между элементами может осуществляться различными путями. Например, информация от процессора $П_1$ к абонентскому пункту $АП_{12}$ может быть передана либо через канал $МК_1$, либо через канал $МК_2$. В таких системах за счет введения структурной избыточности существенно повышается надежность функционирования СРВ.

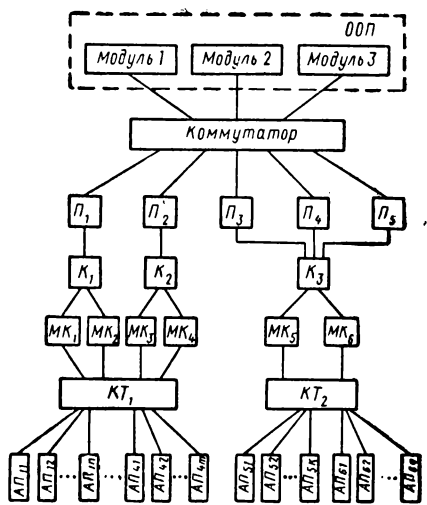


Рис. 18.8. Иерархическая структура многопроцессорной СРВ с мультиплексными связями

18.3. МНОГОПРОЦЕССОРНАЯ СИСТЕМА С ОБЩИМ ПОЛЕМ ОПЕРАТИВНОЙ ПАМЯТИ

Организация общего поля оперативной памяти. Общее поле оперативной памяти для двух и более процессоров, входящих в многопроцессорную систему, может быть реализовано различными способами. При выборе того или иного способа необходимо учитывать такие факторы, как количество процессоров и модулей (блоков) оперативной памяти, объединяемых в систему, техническую базу ВС, обеспечение неизменности (или снижения не ниже допустимого уровня) общей пропускной способности оперативной памяти после объединения обычных однопроцессорных систем в многопроцессорную. Последний фактор указывает на то обстоятельство, что в многопроцессорной системе имеют место потери времени на разрешение конфликтов при одновременном обращении двух или более процессоров к одному и тому же блоку оперативной памяти. Вследствие этого снижается общая пропускная способность оперативной памяти, т. е. уменьшается среднее суммарное число обращений к ней, сопровождаемое выборкой или записью информации.

В многопроцессорной системе возможны две схемы организации обращения к общей оперативной памяти (рис. 18.9):

вариант № 1 (рис. 18.9, а): очереди запросов на обращение к оперативной памяти организуются на входе каждого блока памяти. Каждый очередной запрос анализируется процессором и

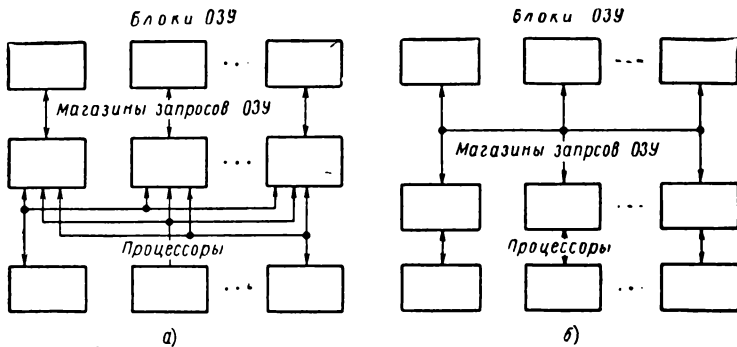


Рис. 18.9. Схемы организации обращения к общей оперативной памяти в многопроцессорной системе:

а — очереди запросов на обращение организуются на входе каждого блока памяти; *б* — очереди запросов организуются на выходе каждого процессора

направляется в магазин запросов соответствующего блока памяти. Удовлетворение запросов производится в порядке их приоритетов;

вариант № 2 (рис. 18.9; б): очереди запросов организуются на выходе каждого процессора. По мере освобождения блоков памяти запросы посылаются на обслуживание, причем право на доступ к памяти получает запрос из магазина того процессора, который имеет наивысший приоритет.

Преимуществами первого варианта схемы являются возможность организации асинхронной работы блоков ОЗУ и возможность реализации широкого диапазона приоритетов. Следует, однако, иметь в виду, что построение системы приоритетов в широком диапазоне их изменения связано с определенными трудностями, так как, с одной стороны, в каждый данный момент в вычислительной системе, как правило, решаются задачи одного или близких уровней приоритетов и, с другой стороны, само понятие приоритета является условным, приоритетность того или иного запроса в динамике работы ВС может изменяться. Последнее связано с тем, что:

- приоритет запроса к ОЗУ на запись обычно выше, чем приоритет на считывание;

- приоритет на обращение за операндом выше, чем приоритет на обращение за командой;

- при одновременном обращении в один блок ОЗУ предпочтение отдается тому запросу, который пропустил цикл обращения;

- при прочих равных условиях преимуществом пользуется запрос, неудовлетворение которого вызывает простаивание соответствующего устройства.

К недостаткам первого варианта следует отнести сложность схемной реализации, обусловленную наличием асинхронных связей с процессорами и необходимостью приведения в соответствие

асинхронно выбираемых команд и операндов, усложнение структуры устройства управления ОЗУ, невозможность активного вмешательства и управления очередью запросов к блоку ОЗУ.

Второй вариант по своей схемной реализации является более простым главным образом потому, что все конфликты разрешаются не на уровне быстродействующих процессоров, как это имеет место в первом варианте, а на уровне сравнительно медленно действующих блоков памяти. Можно указать и другие достоинства этого варианта такие, как возможность управления очередью, изменением приоритета запросов, более четко выраженный модульный принцип построения с возможностью наращивания числа модулей. Вследствие этого второй вариант является предпочтительным при выборе схемы организации обращения к общей оперативной памяти в многопроцессорной системе.

По производительности общей оперативной памяти, т. е. по частоте обращения к ней, сопровождаемого записью или считыванием информации, оба варианта равноценны. Различия заключаются в средствах достижения этой производительности. В первом варианте она достигается увеличением оборудования и количества информационных шин между процессорами и блоками ОЗУ.

В двухпроцессорной вычислительной системе на базе модели ЕС-1050 принята такая организация общей оперативной памяти, в которой используются двухвходовые блоки ОЗУ. Каждый вход обслуживает один из двух процессоров. Конфликты между запросами процессоров на обращение к ОЗУ разрешаются в самих устройствах оперативной памяти.

Режимы работы двухпроцессорной системы на базе моделей ЕС-1030 и ЕС-1050. Для такой системы предусматриваются следующие три режима работы: «Модель» (моносистема), «Расчлененная система» и «Мультисистема».

В первых двух режимах процессоры ВС работают независимо друг от друга. В режиме «Модель» однопроцессорные моносистемы, составляющие мультисистему, работают так, как если бы мультисистемные средства не были установлены, причем работа организуется однопроцессорной управляющей программой. Режим «Расчлененная система» обеспечивает мультипроцессорной управляющей программе доступ к любому из двух процессоров. В этом режиме исключается прямое управление между процессорами (блокируется действие команд прямого управления, используемых при комплексировании на уровне процессоров).

В режиме «Мультисистема» осуществляется объединенная работа двух процессоров и разделенных между ними ОЗУ, ВЗУ, УВВ под управлением общей мультипроцессорной управляющей программы. При работе в этом режиме разрешены пять специальных мультипроцессорных сигналов, которые могут вырабатываться каждым процессором: СООБЩЕНИЕ ОБ ОШИБКЕ, СБРОС СИСТЕМЫ, ВНЕШНИЙ ПУСК, УЧЕТ ПРЕРЫВАНИЙ ВВОДА-ВЫВОДА И ВЫЗОВ СИСТЕМЫ. Первый

из них формируется аппаратно при появлении неисправности в любом процессоре, остальные четыре сигнала выдаются при выполнении команды ПРЯМАЯ ЗАПИСЬ.

Сигнал СООБЩЕНИЕ ОБ ОШИБКЕ выдается в случае отключения питания или появления машинной ошибки. Он поступает в принимающий (исправный) процессор в качестве одного из внешних сигналов, вызывая в нем прерывание. Благодаря этому прерыванию мультипроцессорный СУПЕРВИЗОР может определить наличие ошибки в передающем процессоре, если она препятствует нормальному выходу на прерывание этого процессора. По сигналу СБРОС СИСТЕМЫ производится общий сброс принимающего процессора и всех устройств, доступных этому процессору, а по сигналу ВНЕШНИЙ ПУСК — пуск принимающего процессора. Сигнал УЧЕТ ПРЕРЫВАНИЙ ВВОДА-ВЫВОДА выдается исправным процессором при выполнении им инструкции ПРЯМАЯ ЗАПИСЬ и воспринимается специальным блоком учета прерываний. При выходе из строя одного из процессоров его каналы ввода-вывода продолжают работу и могут выставить сигналы прерывания, которые должны быть обслужены, так как в противном случае канал не освободит соответствующее устройство ввода-вывода и оно окажется недоступным для каналов исправного процессора. Сигнал ВЫЗОВ СИСТЕМЫ вызывает внешнее прерывание в принимающем процессоре, в результате обработки которого осуществляется передача управления к определенной части мультипроцессорного СУПЕРВИЗОРА.

При работе ВС с общим полем оперативной памяти прямоадресуемая область памяти для одного из двух процессоров смещается с целью исключения возможности наложения ее на прямоадресуемую область памяти другого процессора. Прямоадресуемая область — это постоянно распределенная область в оперативной памяти, в которую записывается вся необходимая информация о состоянии процессора и каналов ввода-вывода при возникновении прерываний и машинной ошибке. Для модели ЕС-1050 эта область в немультисистемном режиме работы определена всегда адресами 0—4095.

Организация функционирования ВС в режимах «Мульти-система» и «Расчлененная работа» осуществляется посредством развитой операционной системы ОС-3, основными функциональными особенностями которой являются:

- использование двух процессоров как единого ресурса;
- проведение операций ввода-вывода для задачи, решаемой одним процессором, с помощью другого процессора;
- обработка сигналов по аппаратным неисправностям с помощью обоих процессоров.

Реализация этих функциональных особенностей ОС-3 основана на применении средств прямого управления: команды ПРЯМАЯ ЗАПИСЬ (ПРЯМОЕ ЧТЕНИЕ), сигнал МАШИННАЯ

ОШИБКА, команды ПРОВЕРИТЬ И УСТАНОВИТЬ. По команде ПРЯМОЕ ЧТЕНИЕ, поступающей от передающего процессора, в принимающем процессоре производится один из следующих видов обработки:

- переключение задач;
- анализ необходимости и возможности начала операции ввода-вывода и проведение этой операции.

Многие подпрограммы управляющей программы операционной системы не являются многоходовыми, поэтому программными средствами приходится блокировать вход в такие подпрограммы одного процессора, пока другой процессор не закончит выполнение этой подпрограммы. Блокирование осуществляется посредством команды ПРОВЕРИТЬ И УСТАНОВИТЬ.

Требования к отдельным устройствам многопроцессорной системы. Рассмотрим требования к оперативной памяти, процессору, системе защиты памяти и синхронизации устройств.

Требования к оперативной памяти. Емкость оперативной памяти должна быть достаточной для хранения наиболее активной части операционной системы (т. е. тех программ ОС, которые наиболее часто реализуются в процессе функционирования ВС), а также рабочих программ, выполняемых в данный промежуток времени, и необходимых исходных данных.

Число независимых блоков основной оперативной памяти (ООП) можно выбрать из условия получения требуемой производительности памяти, измеряемой допустимой частотой обращения к ней, по формуле [27, 36]

$$N \geq \frac{N_{\text{п}} f_{\text{п}}}{f_{\text{доп}} k_{\text{доп}}} \left(1 + \frac{1}{\alpha} \right),$$

- где $N_{\text{п}}$ — число процессоров в системе;
 $f_{\text{п}}$ — средняя частота обращений к оперативной памяти со стороны процессора (величина, определяемая производительностью процессора, наличием регистровой памяти и т. п.);
 $N_{\text{п}} f_{\text{п}}$ — общая интенсивность запросов процессоров к памяти;
 $f_{\text{доп}}$ — допустимая частота обращений к блоку памяти;
 $k_{\text{доп}}$ — допустимый коэффициент снижения максимальной производительности оперативной памяти;
 α — коэффициент активности информации. Это среднее число обращений к памяти на единицу введенной информации.

Если, например, $N_{\text{п}} = 2$, $k_{\text{доп}} = 1$, $\alpha = 2$ и, кроме того, $f_{\text{п}} = f_{\text{доп}}$, то $N \geq 3$.

Большая оперативная память должна отвечать требованиям по надежности работы. С целью повышения ее надежности целесообразно введение помехоустойчивого кодирования, допускающего по крайней мере исправление одиночной ошибки на каждый байт запоминаемой информации.

Требование к процессору. В процессорах мультисистемы должна быть предусмотрена работа с расширенной адресацией памяти и внешних устройств. Каждый процессор должен иметь в оперативной памяти свою область отображения состояния прерывания. Возможна и такая организация работы ВС, когда один из ее процессоров выделяется для обслуживания внешних прерываний и обращений к СУПЕРВИЗОРУ. Все остальные процессоры маскируются от внешних прерываний, а также от прерываний по вводу-выводу, за исключением прерываний, вызванных каналами, в которых данный процессор начал ввод-вывод.

Система защиты памяти. Известно, что в системе машин IBM/360, являющихся прототипами моделей ЕС ЭВМ, организация защиты памяти обеспечивает достаточно эффективную защиту как от непредусмотренной записи, так и от неразрешенного считывания. Недостаток такой системы защиты в многопроцессорной ВС заключается в том, что в случае, когда в ООП будет находиться значительное число различных программ, соизмеримое с числом имеющихся ключей, вполне вероятно наличие нескольких из этих программ с одинаковыми ключами. Кроме того, такая система защиты памяти из-за ограниченного набора ключей не обеспечивает защиту от преднамеренных попыток проникновения в чужую область памяти.

В многопроцессорной системе защиту памяти можно строить по принципу определения правомочности обращения исполняемой программы в данную область памяти. Например, можно ограничить исполняемую программу только емкостью своей виртуальной памяти, запрещая все обращения вне ее. Если же необходимо обращение к информации другой программы, производится расширение виртуальной памяти с включением новых страниц. Для определенных привилегированных программ может быть предоставлена возможность обращения к любой области памяти.

Синхронизация устройств. В многопроцессорной системе может быть использован либо асинхронный принцип работы отдельных устройств, либо синхронный. В случае асинхронного принципа достигается максимальное использование устройств системы. Однако это делается за счет существенного усложнения системы, так как разные длительности циклов работы устройств и разные временные задержки в цепях соединений устройств приводят к необходимости установки развязывающих схем на входе каждого из них.

Реализация синхронного принципа значительно проще, поэтому он является более предпочтительным в ВС, хотя его применение и связано с некоторым снижением производительности системы по сравнению с производительностью системы, в которой реализован асинхронный принцип работы ее устройств.

В многопроцессорной системе с применением синхронного принципа используются общие синхронизирующие тактовые сигналы для всех устройств. При этом достигается упрощение схем

соединений за счет разделения по времени работы общих информационных, управляющих и адресных шин. Возможна также компенсация нерегулируемых временных задержек и использование простой регулировки при стыковке устройств. Частота синхронизации определяется максимальной разрешающей способностью устройств и шин передачи информации.

18.4. СТРУКТУРА ОПЕРАТИВНОЙ ПАМЯТИ ДВУХПРОЦЕССОРНОЙ СИСТЕМЫ

Рассмотрим вариант структуры основной оперативной памяти двухпроцессорной системы, построенной на базе моделей ЕС-1030 или ЕС-1050.

Основная оперативная память (ООП) многопроцессорной системы и устройство управления памятью (УУП) строятся по модульному принципу. Количество модулей определяется числом процессоров в системе, а также числом селекторных и мультиплексных каналов, осуществляющих обмен информацией между ООП и внешними устройствами.

Каждый модуль УУП имеет свой позиционный приоритет, т. е. такой приоритет, значение которого не изменяется в процессе работы. Процессоры также имеют позиционные приоритеты, поэтому при исполнении программ (задач) с одинаковым приоритетом предпочтение отдается процессору с наивысшим позиционным приоритетом. В процессе работы может появиться необходимость введения рабочего приоритета исполняемой программы. В этом случае запросы из устройств меньшего позиционного приоритета могут претендовать на более срочное исполнение, чем запросы устройств более высокого позиционного приоритета.

Все блоки ООП сгруппированы по четыре блока в группе (рис. 18.10), причем каждая группа связана со своей шиной обращения. Запросы к каждой группе блоков ООП от модулей УУП производятся практически одновременно. Запросы же внутри группы чередуются со сдвигом во времени. Право на обра-

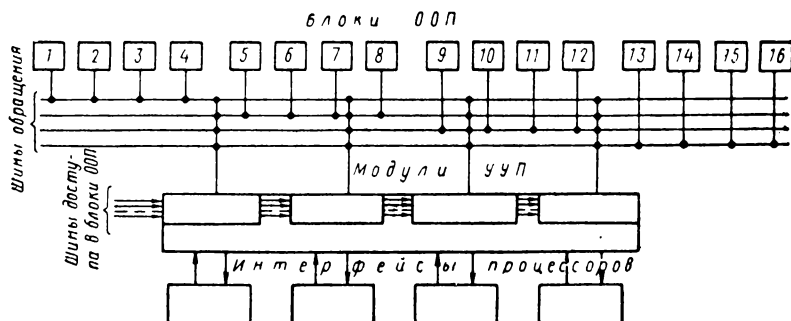


Рис. 18.10. Структура основной оперативной памяти системы

шение получает тот модуль УУП, который содержит наиболее срочный запрос в данный блок памяти. (уровень позиционного приоритета модулей УУП, приведенных на рис. 18.10, уменьшается слева направо).

Кроме шин обращения, имеются шины разрешения доступа в ООП (короче: шины доступа), которые последовательно обходят все модули УУП в порядке уменьшения их уровня позиционного приоритета. Количество шин доступа определяется числом блоков памяти. Если к данному блоку памяти обращение возможно (он исправен и не занят), сигнал на соответствующей шине доступа отсутствует. Если, например, i -й модуль УУП содержит запрос в этот блок памяти, он выдает сигнал по шине доступа на следующий $(i + 1)$ -й модуль УУП, что означает занятость рассматриваемого блока памяти для $(i + 1)$ -го модуля УУП. Если же i -й модуль УУП не содержит запроса в данный блок памяти, он не выдает сигнала на соответствующую шину доступа, и $(i + 1)$ -й модуль УУП получает разрешение на обращение к этому блоку ООП.

Запросы от модулей УУП к блокам ООП посылаются в виде запросных слов, сопровождаемых сигналом по отдельной линии. После удовлетворения запроса этот сигнал сопровождения возвращается в соответствующий модуль УУП также по отдельной линии, что гарантирует от поступления информации в другие процессоры или потери контроля над последовательностью выбираемых слов.

Передача информации в процессор и обратно осуществляется также по групповым шинам с разделением их во времени. Приведение в соответствие выбираемых операндов и команд осуществляется с помощью внутренней адресации оперативных регистров процессоров.

Динамический приоритет реализуется путем передачи сигналов запросов различной степени срочности в направлении, обратном позиционному приоритету. Эти сигналы передаются по трем шинам: по первой шине передается сигнал абсолютного запроса, по второй шине — сигнал аварийного запроса и по третьей — сигнал срочного запроса. Обычные запросы реализуются в пределах модуля УУП.

Схема одного модуля УУП приведена на рис. 18.11. Она имеет магазин адресов запросов (МАЗ) со своим устройством управления (УУ), матричный коммутатор приоритетных шин (МКП) и матричный коммутатор обращений в ООП (МКО).

Магазин адресов запросов содержит ряд регистров и служит для приема, хранения и выдачи адреса в память в момент обращения. Адрес поступает на самый верхний из свободных нижних регистров и по мере удовлетворения запросов перемещается снизу вверх. При этом приоритет запроса возрастает. Устройство управления магазином, кроме схем управления приемом, выдачей и сдвигом адресов, содержит счетчик времени и дешифратор

номера приоритета. В зависимости от номера приоритета и времени ожидания устанавливается то или иное число запросов разной степени срочности. Количество степеней выбирается равным числу приоритетов, принятому для системы.

Матричный коммутатор приоритетов обеспечивает одновременное сравнение приоритетов всех запросов. Логика его работы следующая:

- запросы более высокой степени срочности, поступающие от модулей УУП более низкого позиционного приоритета, подавляют запросы меньшей степени срочности, находящиеся в данном магазине запросов;

- запросы одинаковой степени срочности не подавляются;

- разрешенные (не подавленные) запросы данного магазина поступают на коммутатор обращений;

- запросы, поступившие справа (от модулей более низкого позиционного приоритета), объединяются с запросами данного магазина и передаются налево, к модулю УУП более высокого позиционного приоритета.

Матричный коммутатор обращений обеспечивает выполнение следующих операций:

- сравнение разрешенных запросов данного магазина с сигналами разрешения обращения, поступающими слева, от модулей УПП более высокого позиционного приоритета. Наличие запроса и сигнала разрешения означает возможность обращения к блокам памяти для данного магазина;

- выдачу сигнала возможности обращения в магазин запросов, если имеет место ситуация, описанная выше, и, кроме того, появился сигнал окончания цикла обращения;

- объединение сигналов обращения данного магазина с сигналами разрешения, поступившими слева, и передача сигналов разрешения направо, к модулям УУП более низкого позиционного приоритета.

При такой организации выборки слов из оперативной памяти обеспечивается непрерывный анализ и сравнение запросов всех уровней срочности, а также возможность удовлетворения запросов в любой подходящей последовательности, возможность управления очередью запросов, выбор оптимальной скорости удовлетворения запросов.

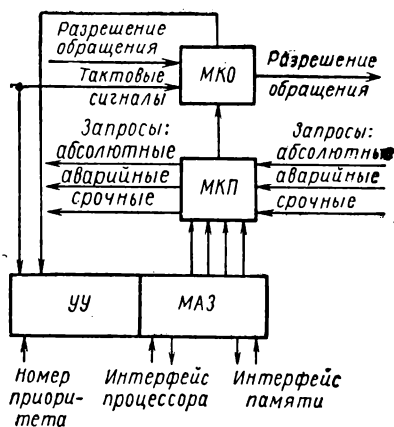


Рис. 18.11. Схема модуля устройства управления памятью

18.5. ОБМЕН ИНФОРМАЦИЕЙ МЕЖДУ МАШИНАМИ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

При функционировании электронных вычислительных машин ВС осуществляется обмен информацией между машинами, причем характер информации может быть самым различным — это управляющая информация (команды программ операционной системы), информация, характеризующая техническое состояние или занятость машин, исходные данные для решения задач, основные программы (или их части) решения задач и т. д. Техническое и программное решение всех вопросов по обеспечению обмена информацией между ЭВМ существенно зависит от того, к какому типу принадлежит рассматриваемая система — к системам разобщенного или совмещенного типа.

В системах разобщенного типа для передачи информации от одной машины к другой приходится использовать линии связи большой протяженности. При совместной работе ЭВМ с такими линиями связи возникает ряд специфических проблем, поэтому представляется целесообразным рассмотреть отдельно вопросы, связанные с организацией и обеспечением обмена информацией в системах разобщенного и совмещенного типов.

Обмен информацией в ВС разобщенного типа. В системах разобщенного типа, когда расстояния между элементами ВС таковы, что при оценке качества функционирования системы приходится принимать во внимание время, затрачиваемое на обмен информацией между машинами, необходимо обеспечить совместную работу ЭВМ с линиями связи.

Говоря о совместной работе ЭВМ системы и линий связи, нужно иметь в виду следующее:

— машины и линии связи обладают различными техническими параметрами: скорость обработки информации в ЭВМ во много тысяч раз превышает скорость передачи информации по линиям связи, т. е. отсутствует синхронность в работе ЭВМ и линий связи;

— информация по линии связи передается в последовательном коде, а современные ЭВМ, как правило, работают с информацией, представленной в параллельном коде. Поэтому возникает необходимость преобразования кода информации из параллельного в последовательный при вводе ее в линию связи и обратного преобразования при выводе из линии связи;

— в общем случае число линий связи, работающих на данную ЭВМ, и число ЭВМ, работающих на данную линию связи, может быть случайным. Кроме того, степень загрузки работающих линий связи и их пропускная способность могут быть различными.

Одной из важнейших задач в ВС разобщенного типа является задача обеспечения верности информации при передаче ее на многие десятки и сотни километров. Эта задача решается путем

использования различных помехоустойчивых кодов. На передающем пункте из передаваемых слов формируются так называемые сообщения, или кодограммы, состоящие из информационных, контрольных и служебных символов. Число контрольных символов в кодограмме и их позиции зависят от того, какой помехоустойчивый код используется для повышения верности передаваемой информации. На приемном пункте с помощью соответствующих технических средств, в частности декодирующих устройств, осуществляется раскодирование сообщений: проверка их верности (обнаружение и исправление или только обнаружение ошибок в сообщениях), выделение информационных символов и формирование из них слов. Естественно, что применение помехоустойчивых кодов с целью повышения верности информации, передаваемой от одной машины системы к другой, сопряжено с существенным увеличением времени на ее передачу.

Тот факт, что обработка информации осуществляется во много раз быстрее, чем передача ее по линиям связи, в современных ЭВМ часто используется, так как представляется возможным в промежутках между поступлениями информации производить ее обработку. Известно, что в зависимости от сложности алгоритма переработки информации на каждое поступающее слово приходится от нескольких десятков до нескольких сот и даже тысяч машинных операций. Конечно, не все время между поступлением разрядов слов может быть полностью использовано для обработки информации. Часть времени затрачивается на формирование слов из этих разрядов. Собственно обработка информации начинается только после формирования слов, чисел и целых сообщений. Кроме того, довольно значительные потери времени могут иметь место для обеспечения согласования работы ЭВМ и нескольких линий связи с различной загрузкой и различной пропускной способностью, а также при повторной передаче информации по линиям связи в тех случаях, когда это необходимо для обеспечения требуемой верности принимаемой информации.

Решение всех вопросов по обеспечению совместной работы ЭВМ ВС и линий связи может быть возложено как на аппаратуру системы передачи информации (СПИ), так и на сами вычислительные средства. Распределение функций обмена информацией между СПИ и ЭВМ может быть в различных соотношениях. Эти функции могут выполняться, например, только вычислительными машинами. Однако в этом случае на выполнение функций обмена затрачивается очень много машинного времени и емкости запоминающих устройств, поэтому может случиться, что производительности ЭВМ будет недостаточно для выполнения основной работы по обработке информации. Выполнение функций обмена только аппаратурой СПИ сопряжено с усложнением этой системы. В этом случае в состав СПИ может входить специализированная ЭВМ обмена.

Примером ЭВМ, специализированной для обмена информацией, является машина IBM-7741 (США). Эта машина выполняет такие функции, как [6]:

- преобразование сигналов, поступающих из линий связи, в сигналы, необходимые для работы машины, и наоборот;
- устранение искажений сигналов;
- проверка и коррекция принимаемой информации;
- выделение из принимаемых сообщений (кодограмм) информационных символов и сборка из них слов, программный анализ сообщений;
- прием и передача сообщений переменной длины;
- преобразование последовательного кода информации в параллельный и обратно;
- преобразование кодов принимаемых сообщений в код ЭВМ ВС;
- сопряжение между собой многих линий связи с различными пропускными способностями;
- коммутация сообщений, поступающих из одной линии связи в другую;
- установление очередности обработки сообщений, соблюдение приоритетных правил;
- регистрация загрузки линий связи;
- проверка линий связи и изоляция неисправных линий, проверка всей аппаратуры передачи и приема информации;
- указание и регистрация времени суток.

В машине IBM-7741 выполнение этих функций обеспечивается аппаратными и программными методами. В состав машины входит линейный адаптер и собственно процессор. С помощью линейного адаптера осуществляется преобразование информации: при приеме из линий связи коды преобразуются к виду, приемлемому для процессора, и, кроме того, сигналы освобождаются от искажений, при передаче неискаженные сигналы преобразуются в сигналы связи. Программными методами выполняются такие операции, как коммутация сообщений, анализ сообщений, проверка и коррекция принимаемой информации и др.

Обмен информацией в ВС совмещенного типа. В ВС совмещенного типа отсутствуют линии связи большой протяженности между машинами, поскольку ЭВМ расположены в непосредственной близости друг от друга. Обмен информацией между машинами непосредственно или через общую оперативную память осуществляется в параллельном коде, без потерь времени на преобразование кодов — из параллельного в последовательный и наоборот. Все это благоприятствует достижению весьма высокой степени применения принципов модульности и параллельного выполнения вычислений в различных режимах работы системы.

К числу наиболее серьезных вопросов, которые должны решаться при организации обмена информацией между машинами в системе совмещенного типа, относятся следующие:

— построение системы и организация вычислительного процесса таким образом, чтобы довести до минимума время на обмен информацией;

— обеспечение необходимой точности при обмене информацией между разнотипными машинами, отличающимися разрядностью и формой представления чисел;

— обеспечение обмена программами между машинами, различными по своим системам команд.

Следует отметить что все эти вопросы не являются специфическими только для ВС совмещенного типа. Они решаются и при построении и организации работы систем разобщенного типа. Однако примеров удачного их решения гораздо больше для совмещенных систем, поскольку они являются более распространенными.

В специализированных вычислительных системах (СВС), где круг решаемых задач ограничен и известен заранее, сокращение времени на обмен информацией между машинами достигается, помимо других путей, за счет уменьшения объема обмениваемой информации, что, в свою очередь, обеспечивается рациональной организацией вычислительного процесса. В универсальных вычислительных системах (УВС) такой путь тоже используется, хотя его реализация связана с необходимостью выполнения управляющей программой оптимального (или приемлемого) распределения рабочих программ между машинами применительно к каждому новому пакету этих программ.

В вычислительных системах с общей памятью, как специализированных, так и универсальных, для уменьшения времени на обмен информацией применяется модульная структура общей оперативной памяти. Каждый из модулей ООП на некоторое время может закрепляться за любой машиной системы. Это обеспечивает возможность быстрой и одновременной передачи больших массивов информации путем переключения одного или нескольких модулей ООП от одной машины к другой. Число модулей обычно больше числа машин в системе, так как в каждый момент времени за каждой машиной может быть закреплено более одного модуля.

В ВС с модульной структурой ООП большое значение имеет наличие эффективной, надежной, наращиваемой и экономичной системы коммутации данных (межмодульной коммутации).

Межмодульная коммутация может строиться в соответствии с одним из трех основных принципов [6].

Первый из них основан на пространственной селекции. Основной частью пространственного коммутатора является переключательная матрица, каждая строка которой жестко соединена с одной из ЭВМ системы, а каждый столбец — с одним из модулей ООП. Соединение i -й ЭВМ с j -м модулем ООП осуществляется с помощью управляемой вентильной схемы, находящейся в точке пересечения i -й строки с j -м столбцом переключательной матрицы.

Будучи в теоретическом отношении простым и понятным, такой принцип организации межмодульной коммутации имеет ряд конструктивных недостатков;

— большое число управляемых вентиляльных схем (оно равно произведению числа ЭВМ на число модулей и на число разрядов в обмениваемых словах), что отрицательно сказывается на надежности и стоимости коммутационного устройства;

— ограничение скорости переключения модулей и соответственно оперативности перестройки работы ВС при быстроменяющихся условиях ее функционирования за счет задержки кодовых сигналов в вентиляльных схемах переключательной матрицы;

— необходимость в изменении частоты синхронизации, возникающая при наращивании системы (импульсы синхронизации вырабатываются задающим генератором), поскольку скорость передачи данных через коммутирующее устройство зависит от его физических размеров подобно тому, как быстродействие ЗУ зависит от его емкости.

В связи с этим межмодульная коммутация, основанная на пространственной селекции, не применяется в тех ВС, в которых обмен информацией между модулями ООП должен осуществляться за время, сравнимое с длительностью машинных операций. Коммутаторы с пространственной селекцией применяются в некоторых американских вычислительных системах, в которых межмодульные соединения изменяются сравнительно редко и, кроме того, не предъявляется серьезных требований к гибкости системы.

Второй метод межмодульной коммутации основан на использовании принципа кодовой селекции. В соответствии с этим методом все машины постоянно соединены со всеми модулями ООП, т. е. вентиляльные схемы отсутствуют. Точно так же связаны друг с другом модули ООП. Однако действительный информационный обмен производится только между теми модулями, между которыми установлено соответствие кодов. На запрос машины отвечает только тот модуль, который указан в коде запроса. Для того чтобы избежать одновременного доступа к одному и тому же модулю со стороны более чем одной машины ВС, в таком коммутирующем устройстве необходима некоторая система установления очередности, которая должна анализировать приоритеты заявок. На каждом модуле должна быть отдельная схема адресации для каждой машины, что при наращивании ВС потребует дополнительного оборудования для каждого модуля.

Третий метод межмодульной коммутации основан на использовании принципа временной селекции. Все машины и все модули ООП подключены постоянно к общей информационной шине. Используется одна и та же схема адресации к каждому модулю для обслуживания запросов от всех машин ВС. Для обеспечения нужного направления потоков информации кодируются как модули, так и машины. В каждый данный момент времени доступ

к информационной шине имеет только одна точно определенная машина, а прием информации от этой машины производится только определенным модулем. В течение одного стандартного цикла обращения к модулям каждой машине системы предоставляется строго определенный интервал времени для связи с выбранным модулем. Во время одного цикла машина посылает запрос к модулю, а во время следующего цикла получает ответ от запрашиваемого модуля. Таким образом, здесь автоматически решается проблема конфликтов при одновременном обращении нескольких машин к одному и тому же модулю, а также проблема очередей.

Применение принципа временной селекции дает существенный выигрыш в отношении времени коммутации модулей и объема оборудования по сравнению с первыми двумя методами коммутации.

Выполнение вычислительных работ в ВС осуществляется с ошибками, которые, в зависимости от их источников, можно разделить на две большие группы:

- ошибки, не зависящие от технических параметров системы;
- ошибки, обусловленные несовершенством и ограниченностью средств вычислительной техники, или инструментальные ошибки.

К первой группе относятся:

- методологические ошибки, т. е. ошибки, обусловленные несовершенством математических моделей изучаемых проблем. Возникновение этих ошибок относится не к стадии вычислений, реализации полученных алгоритмов, а к стадии разработки теории рассматриваемой проблемы. Они характеризуют степень разработки теории изучаемых процессов, а не техническое совершенство и качество работы ВС;

- ошибки, обусловленные неточностью входной информации, исходных данных для решения задач;

- методические ошибки, обусловленные несовершенством выбранных методов решения заданных систем уравнений. Сюда относятся ошибки численных методов решения и аппроксимации трансцендентных функций и чисел, а также ошибки итеративных методов, требующих для полной сходимости решения бесконечного или очень большого числа итераций.

Основные виды инструментальных ошибок: ошибки округления, возникающие из-за ограниченности разрядной сетки машин ВС; ошибки, вызываемые отказами и сбоями (самоустраняющимися отказами) в различных элементах ВС, принимающих участие в вычислениях; ошибки, обусловленные сопряжением разнотипных ЭВМ системы, отличающихся разрядностью и формой представления чисел.

Решение задачи обеспечения обмена программами между машинами ВС определяется степенью их программной совместимости. По этому вопросу, а также по вопросу оценки ошибок, обусловленных сопряжением разнотипных ЭВМ, необходимый материал имеется в книге Ю. С. Голубева—Новожилова [6].

18.6. ПОКАЗАТЕЛИ КАЧЕСТВА ФУНКЦИОНИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Принадлежность вычислительной системы к тому или иному типу в соответствии с указанными выше классификационными признаками во многом определяет ее свойства. Например, система «Минск-222», построенная на базе ЭВМ «Минск-22», в соответствии с этими признаками является универсальной, однородной, совмещенного типа, с переменной структурой (поскольку программным способом осуществляется изменение числа машин системы, функционирующих в режиме взаимодействия, а также разбиение системы на подсистемы, каждая из которых объединяет две машины и более), централизованно-децентрализованная (машинадиректор отсутствует, но ее функции может выполнять любая машина системы, в то же время машины ВС могут использоваться автономно), с плавающим закреплением функций (иначе такая система называется равновесной в том смысле, что все машины с точки зрения выполняемых функций равноправны, имеют одинаковый вес), без общей оперативной памяти.

К числу свойств систем, не определяемых их классификационными признаками, относятся такие, как способность работы в различных режимах, возможность работы в реальном масштабе времени, адаптивность. Следовательно, под свойствами системы понимаются такие ее характеристики, которые, как правило, не выражаются количественно, не являются функционалами от процесса функционирования системы.

Показатель качества — это некоторая численная характеристика, являющаяся функционалом от процесса функционирования системы и определяющая одну из сторон качества этого процесса. Показатели качества ВС можно разбить на две группы. В первую группу включаются показатели, которые представляют собой вероятности (и их комбинации) нахождения системы в различных состояниях, и показатели, определяемые по формулам полной вероятности и Байеса. Во вторую группу входят показатели, являющиеся статистическими оценками некоторых сторон процесса функционирования системы, например, пропускная способность системы, временные характеристики качества обслуживания требований на решение задач (среднее время на обслуживание, среднее время ожидания обслуживания и др.), коэффициент загрузки элементов ВС.

Процесс функционирования ВС происходит в условиях воздействия случайных факторов, оказывающих большее или меньшее влияние на качество этого процесса. Влияние j -го фактора на i -е качество системы может быть оценено с помощью коэффициента [4]

$$K_{ij} = \frac{1 - P_i}{1 - P_{ij}},$$

где P_i — показатель i -го качества системы, вычисленный без учета влияния j -го фактора; P_{ij} — показатель i -го качества, вычисленный с учетом влияния j -го фактора.

Наиболее полно качество функционирования системы оценивается с помощью показателей эффективности ВС (W_i). Эффективность — это качество системы, отражающее степень соответствия ее своему назначению, техническое совершенство и экономическую целесообразность. Количественно эффективность системы оценивается через показатели эффективности, являющиеся функциями множества параметров входящих потоков требований (или параметров реализуемых алгоритмов) L_n , множества параметров системы и ее элементов L_c и множества параметров, характеризующих условия функционирования ВС, влияние внешних факторов на процесс функционирования L_y . Показатель эффективности определяется процессом функционирования системы, он является функционалом от этого процесса. В качестве показателей эффективности принимаются те показатели качества ВС, которые в наибольшей степени отражают степень соответствия системы своему назначению.

Основные требования, предъявляемые к показателям эффективности ВС: они должны выбираться исходя из интересов тех задач, которые решаются системой, т. е. они действительно должны количественно отражать степень соответствия системы своему назначению; они должны быть критичны к параметрам множеств L_n , L_c , L_y ; они должны быть по возможности простыми, имеющими четкий физический смысл; они должны быть эффективными в статистическом смысле, т. е. должны иметь сравнительно небольшую дисперсию; если ВС является подсистемой более сложной системы, то ее показатель эффективности должен рассматриваться как аргумент показателя эффективности этой сложной системы; показатели эффективности (если их выбор производится еще на этапе проектирования системы определенного назначения) должны обеспечивать относительную свободу выбора технических решений в рамках предъявленных требований.

Обычно не удается установить какой-то один достаточно общий показатель эффективности ВС, который бы удовлетворял этим требованиям. В связи с этим возникает проблема множественности показателей эффективности. От количества используемых для оценки эффективности ВС показателей зависит число параметров множеств L_n , L_c , L_y , участвующих в формировании их значений. С увеличением же числа параметров возрастает объем информации, характеризующей систему, т. е. возрастает определенность представления о системе. Однако наряду с этим существенно усложняется задача определения значений показателей эффективности и особенно законов их распределения, если это величины случайные.

В зависимости от назначения ВС и характера решаемых ею задач из всего перечня показателей, выбранных для оценки

эффективности системы, один принимается в качестве основного. Он должен в наибольшей степени характеризовать эффективность ВС.

Наиболее универсальной характеристикой качества функционирования ВС, которую можно принять в качестве основного показателя ее эффективности, является вероятность выполнения поставленных перед ней задач на требуемом уровне за определенное время τ и при заданных условиях функционирования — $P_{вз}(\tau)$. На величину этого показателя влияют параметры всех множеств L_r , L_c , L_y . Однако вычисление значений $P_{вз}(\tau)$ при оценке эффективности сложных ВС сопряжено с непреодолимыми трудностями, обусловленными большим количеством параметров, влияние которых необходимо учесть, и подчас отсутствием явно выраженных функциональных зависимостей, определяющих это влияние. Поэтому практически в качестве основного показателя эффективности ВС приходится выбирать такой показатель, который, с одной стороны, в отношении полноты оценки системы достаточно близко примыкает к показателю $P_{вз}(\tau)$ и, с другой стороны, может быть вычислен с помощью тех или иных методов.

Роль основных показателей эффективности ВС (в зависимости от ее назначения) могут выполнять такие показатели, как время на решение заданного набора задач (для систем, работающих в режиме пакетной обработки); время обслуживания абонента (для систем с разделением времени); относительная пропускная способность ВС — это отношение среднего числа обслуженных требований за некоторый интервал времени к среднему числу поступивших требований за тот же интервал; коэффициент задержки в обслуживании требований — это отношение среднего времени пребывания требования в системе к среднему времени, затрачиваемому на обслуживание требования (на решение задачи, соответствующей этому требованию).

Важное значение имеют и другие показатели, которые могут использоваться в качестве частных показателей эффективности ВС. К ним относятся: абсолютная пропускная способность (среднее число требований, обслуживаемых системой в единицу времени), время ожидания начала обслуживания и закон распределения этого времени, длина очереди требований, задаваемая законом распределения длины очереди, коэффициент загрузки системы (отношение времени занятости системы обслуживанием требований в некоторый период испытания T_n к разности $T_n - T_{нв}$, где $T_{нв}$ — непроизводительное время, расходуемое на профилактическое обслуживание ВС, на проведение функционального контроля, на восстановление ВС после отказов) и др.

При оценке качества ВС употребительными являются такие показатели, как достоверность выходной информации, надежность системы, помехозащищенность, экономичность.

Достоверность информации, получающейся на выходе системы в результате реализации заданного набора задач, может оцени-

ваться вероятностью того, что в выходной информации системы не содержится ошибки. Показатель достоверности относится к категории обобщенных показателей качества функционирования системы. Он носит системный характер, так как ошибки в выходной информации складываются из обычных ошибок, возникающих в ЭВМ при выполнении вычислений, и ошибок, появляющихся при передаче информации по каналам связи между машинами ВС, при сопряжении между собой машин с различными техническими параметрами, такими как разрядность и форма представления чисел. Кроме того, при последовательной информационной связи между машинами системы выходные ошибки предыдущей машины становятся входными ошибками последующей машины.

Надежность ВС — это способность системы сохранять свои свойства при заданных условиях работы в течение определенного промежутка времени. Количественной оценкой надежности более или менее простых вычислительных систем, содержащих элементы, отказ которых приводит к отказу всей системы, могут служить следующие показатели: $P_c(\tau)$ — вероятность безотказной работы системы за время τ при заданных условиях эксплуатации; T_c — наработка на отказ системы; $T_{вс}$ — среднее время восстановления системы. Необходимо подчеркнуть, что эти показатели имеют смысл именно для сравнительно простых систем, по отношению к которым применимо понятие «отказ».

Для многих более сложных вычислительных систем понятие «отказ системы» не имеет смысла. В таких системах отказы отдельных элементов приводят лишь к некоторому снижению их эффективности, а не к полной потере работоспособности ВС в целом. Например, в многоканальной вычислительной системе, рассматриваемой как система массового обслуживания, отказы отдельных каналов лишь изменяют в худшую сторону показатели, характеризующие качество функционирования ВС.

В качестве показателя надежности сложной вычислительной системы могут быть использованы [3]:

в) коэффициент снижения эффективности системы

$$K_w = \frac{W}{W_0},$$

показывающий, какую часть эффективности реальной системы (W) составляет от эффективности идеальной (в смысле надежности) системы (W_0);

б) величина разности

$$\Delta W_0 = W_0 - W,$$

показывающая, насколько снижается эффективность ВС вследствие отказов ее элементов по сравнению с эффективностью идеальной системы, элементы которой абсолютно надежны.

Величины K_w и ΔW_0 дают полное представление о влиянии параметров надежности элементов системы на эффективность ее

функционирования. Так, если значение $K_{\text{в}}$ достаточно велико или разность ΔW_0 достаточно мала, отказы элементов системы слабо влияют на ее эффективность, и поэтому могут оказаться нецелесообразными меры повышения надежности элементов ВС, поскольку они не оправдывают произведенных затрат. При малом значении $K_{\text{в}}$ или достаточно большой разности ΔW_0 такие меры необходимы, особенно в отношении элементов, отказы которых оказывают наибольшее влияние на эффективность системы.

Помехозащищенность системы — это способность системы сохранять свои свойства в течение определенного промежутка времени в условиях действия помех с известными характеристиками.

Количественная оценка помехозащищенности ВС может осуществляться с помощью следующих показателей:

а) коэффициента помехозащищенности системы

$$K_{\text{пз}} = \frac{W_{\text{п}}}{W_0},$$

показывающего, какую часть значение показателя эффективности системы, функционирующей в условиях действия помех с известными характеристиками ($W_{\text{п}}$), составляет от значения показателя эффективности системы, функционирующей в идеальных условиях (W_0), когда помехи отсутствуют;

б) абсолютной величины разности

$$\Delta W_{\text{пз}} = |W_0 - W_{\text{п}}|,$$

показывающей, насколько изменяется эффективность ВС под влиянием помех с известными характеристиками.

Готовность системы — это показатель, характеризующий качество и быстроту выполнения операций по подготовке системы к применению. В число подготовительных операций могут входить: включение элементов системы, измерение некоторых параметров системы, решение тестовых задач, переключение системы с одного режима работы на другой и др.

В качестве показателей, характеризующих готовность системы к применению, могут быть использованы:

а) вероятность того, что система будет приведена в готовность к выполнению своих функций за время, не превышающее допустимое τ ,

$$P_r(\tau) = P\{\tau_r < \tau\},$$

где τ_r — время выполнения всех подготовительных операций на заданном уровне;

б) среднее время подготовки системы к работе.

Экономичность системы оценивается затратами на ее разработку, создание и эксплуатацию. Количественно экономичность ВС может оцениваться с помощью показателя экономической

эффективности (C_3) — стоимости одной вычислительной операции, определяемой с учетом затрат на разработку, создание и эксплуатацию системы.

18.7. ЗАДАЧИ ИССЛЕДОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

На каждом из этапов создания ВС возникает ряд вопросов, решение которых может быть получено только в результате достаточно глубокого исследования системы.

Различают два основных класса задач исследования вычислительных систем:

а) *задачи анализа*, связанные с оценкой качества функционирования системы с известными параметрами, характеризующими систему и ее элементы, условия эксплуатации, потоки заявок на решение задач, изучением свойств и поведения системы в зависимости от ее структуры и значений параметров;

б) *задачи синтеза*, сводящиеся, как правило, к определению таких значений параметров системы и основных ее элементов, при которых показатели качества функционирования системы, обслуживающей заданные потоки заявок в определенных условиях эксплуатации, были бы не хуже требуемых.

На практике при модернизации ВС, при проектировании новых вычислительных систем и определении оптимальных режимов их работы задачи анализа часто сводятся к оценке возможных вариантов системы. Оценка этих вариантов осуществляется путем определения и сравнения значений показателей, характеризующих основные свойства системы. При выборе подходящего варианта системы необходимо обращать внимание не только на оптимальность значений показателей, но и на их стабильность при изменении параметров системы в определенных пределах.

Задачи анализа ВС могут формироваться в более частной постановке, преследуя цель проведения исследования отдельных сторон процесса функционирования системы. Отметим некоторые из этих задач.

1. Оценка различных вариантов структуры системы, проводимая с достаточно глубоким анализом влияния случайных факторов на динамику функционирования ее оборудования.

2. Согласование различных элементов системы по производительности. Для этого оценивается производительность элементов ВС и проводятся расчеты по выбору параметров элементов, обеспечивающих требуемую их производительность.

3. Определение оптимальных режимов работы системы при заданных условиях эксплуатации и известных параметрах потоков заявок на обслуживание. Оптимальный режим устанавливается по максимальному (минимальному) значению показателей, характеризующих основные свойства системы и вычисленных для различных режимов ее работы.

4. Оценка оптимальной степени централизации управления ВС. Решение этой задачи можно получить, анализируя результаты моделирования системы при различной степени централизации управления.

5. Определение оптимальной дисциплины обслуживания требований (заявок на решение задач) с учетом их приоритетности. Эта задача решается путем определения времени обслуживания требований различной приоритетности (или времени ожидания обслуживания) и сравнения этого времени для рассматриваемых дисциплин обслуживания.

6. Оценка целесообразности различных технических усовершенствований системы. При решении этой задачи зачастую приходится устанавливать, какой ценой достигнуто повышение качества системы (повышение ее производительности, увеличение пропускной способности, повышение надежности, помехозащитности, готовности и т. д.), обусловленное реализацией предлагаемых усовершенствований.

7. Оценка относительного выигрыша в эффективности системы от применения различных вариантов приоритетного обслуживания требований по сравнению с беспriorитетным обслуживанием, если оно допустимо. Для систем с потерями вследствие ожидания в очереди этот выигрыш можно оценить по величине уменьшения необходимой производительности ВС при введении приоритетной дисциплины обслуживания требований по сравнению с беспriorитетным обслуживанием (обслуживание в порядке поступления).

8. Оценка эффективности систем диспетчеризации, проводимая с учетом затрат на их реализацию, выражаемых в сопоставимых единицах (время счета, количество ячеек памяти для хранения программ).

Для решения перечисленных и других задач оценки качества функционирования ВС широко используется метод статистического моделирования, представляющий собой метод исследования сложных динамических процессов путем многократных реализаций на ЭВМ статистической модели изучаемого процесса и обработки большого объема вырабатываемой статистической информации, которая по своему характеру близка к информации, получаемой в ходе натурных испытаний. Под статистической моделью понимается совокупность соотношений (математических, логических), в наиболее полной форме воспроизводящих картину реальных процессов, имеющих место в рассматриваемых системах при некоторых значениях случайных возмущающих факторов.

При использовании метода статистического моделирования математическая модель исследуемого процесса преобразуется в так называемый моделирующий алгоритм (МА), представляемый в виде операторной схемы. Эта схема содержит последовательность операторов, каждый из которых изображает определенную группу элементарных действий. Запись МА в виде операторной схемы производится без учета системы команд и особенностей

ЭВМ, на которой будет реализован алгоритм. Учет этих особенностей осуществляется при составлении программы решения задачи на данной ЭВМ.

При исследовании ВС плодотворным оказывается сочетание аналитического метода и метода статистического моделирования. В этом случае аналитический метод используется для прикладной оценки системы или анализа отдельных ее подсистем, а метод статистического моделирования — для окончательного, полного анализа всей системы.

Общая задача синтеза ВС, в строгой ее постановке, это задача математической оптимизации, в результате решения которой с учетом заданных ограничений определяются такие значения параметров множеств L_c , L_y , которые бы обеспечивали обращение в максимум (или минимум) показатели (одного или нескольких) эффективности системы, обслуживающей потоки требований с известными параметрами. Иначе говоря, задача состоит в определении таких значений параметров множеств L_c , L_y , которые бы обеспечивали удовлетворение заданных требований по качеству функционирования системы, оцениваемых через показатели качества (главным образом, через показатели эффективности, поскольку они являются критериями выбора оптимального решения задачи синтеза ВС).

Однако в настоящее время методов решения общей задачи синтеза вычислительной системы нет, так как практически невозможно проводить оптимизацию по всем параметрам множеств L_c , L_y . В инженерной практике большое значение приобретают частные постановки общей задачи синтеза ВС.

Приведем примеры постановки частных задач синтеза вычислительной системы.

1. Заданы состав и структура системы, набор и алгоритмы решаемых задач, параметры потоков заявок на решение задач, условия эксплуатации ВС, требуемые значения показателей эффективности. Необходимо найти оптимальные значения параметров элементов системы.

2. Найти оптимальные значения параметров аппаратно-программной системы управления вычислительного комплекса заданной структуры, обслуживающего потоки требований с известными параметрами в определенных условиях эксплуатации. Целевая функция управления также задана. В результате решения этой задачи выбирается структура аппаратно-программной системы управления, устанавливается оптимальное соотношение между аппаратными и программными средствами управления, выбирается приемлемая (особенно для требований высшего приоритета) дисциплина обслуживания требований.

3. Определить оптимальные значения параметров выбранного режима профилактического обслуживания системы (длительность и периодичность обслуживания), обеспечивающие получение максимального значения коэффициента полезного времени ВС.

Параметры системы и ее элементов, а также параметры множества L_n известны.

В ходе реализации частных задач синтеза ВС решаются такие вопросы, как выработка требований к основным техническим параметрам ЭВМ (процессорам) системы или выбор наиболее подходящего типа ЭВМ из числа имеющихся образцов; выработка требований к системным запоминающим устройствам — к их структуре, емкости, времени обращения, скорости записи-считывания информации, а также требований к средствам защиты памяти; определение состава, структуры и требований к аппаратуре сопряжения системы с источниками и потребителями информации; разработка требований к основным параметрам аппаратно-программной системы функционального контроля вычислительного комплекса и др.

При этом, как правило, возможно множество решений. Задача состоит в том, чтобы из этого множества решений выбрать приемлемое или оптимальное решение.

Процесс принятия оптимального решения включает обычно следующие элементы: выбор показателей качества ВС, с помощью которых оцениваются результаты альтернативных способов действий; составление перечня возможных альтернативных способов действий (решений); определение соотношений, связывающих альтернативные решения со значениями показателей качества ВС; выявление одной, оптимальной альтернативы, при которой показатели качества принимаю экстремальные значения.

При решении задач синтеза ВС часто приходится прибегать к раздельной оптимизации подсистем вычислительной системы, т. е. к субоптимизации ВС.

18.8. ЭТАПЫ ПРОЕКТИРОВАНИЯ ВС И ИХ СОДЕРЖАНИЕ

Проектирование ВС представляет собой весьма сложный и трудоемкий процесс, выполнение которого под силу только достаточно большим коллективам высококвалифицированных научных работников различной специальности. Задание на проектирование выдает «Заказчик», в роли которого выступает учреждение или организация, ведающая вводом в строй и эксплуатацией вычислительных систем. Заказчиком может быть также организация, занимающаяся разработкой более крупного комплекса, в который проектируемая ВС входит как составная часть. В этом случае эта организация, на основании полученных от «Заказчика» общих требований к комплексу, составляет частное техническое задание (ЧТЗ) на разработку ВС.

Задание на проектирование ВС содержит общие требования к системе (включая технические, эксплуатационные и конструктивные требования), перечень и исходные алгоритмы решаемых задач (исходных алгоритмов в задании может и не быть, и тогда

разработка алгоритмов решаемых задач является одним из этапов проектирования ВС, выполняемых «Разработчиком»), данные по интенсивности потоков заявок на решение задач, их приоритетности, требования по времени решения задач.

«Разработчик» вычислительной системы — это различные научно-исследовательские институты (НИИ), конструкторские бюро (КБ) заводов или вузов, научно-исследовательские лаборатории, вообще научно-исследовательские организации, располагающие собственной достаточно мощной экспериментально-производственной базой или имеющие возможность при выполнении заказа привлекать соответствующие заводы к работам по созданию и испытанию опытных образцов аппаратуры ВС. «Разработчик», на основании заданных общих требований к ВС, составляет полные технические требования, осуществляет проектирование системы, ее изготовление и испытание.

Очевидно, что в ходе проектирования некоторой новой ВС должен использоваться накопленный опыт по их созданию. Это тем более необходимо, поскольку зачастую в аппаратном отношении новые системы представляют в значительной степени новые комбинации уже известных, освоенных промышленностью и проверенных в эксплуатации элементов, и лишь небольшое число элементов ВС являются радикально новыми конструкциями. Это замечание в известной мере справедливо и в отношении вопросов организации управления ВС, выбора дисциплины обслуживания требований (пользователей), организации технического обслуживания системы. В связи с этим в ходе проектирования ВС приходится заниматься экстраполированием и интерполированием имеющихся данных (полученных при эксплуатации подобных систем) с учетом новых и часто неизвестных обстоятельств. По этой же причине отдельные этапы проектирования системы, о которых речь идет ниже, могут отсутствовать или объединяться с другими этапами, т. е. последовательность проектирования ВС в каждом конкретном случае может изменяться, уточняться в большей или меньшей степени.

При проектировании системы приходится удовлетворять ряду требований, часть из которых может быть взаимно противоречивой. Поэтому должно быть найдено такое сочетание параметров системы и ее элементов, которое с точки зрения удовлетворения выбранным критериям качества ВС является оптимальным или приемлемым. Сам процесс проектирования в этом случае приобретает итеративный характер.

Основными этапами создания сложной системы вообще, вычислительной системы в частности, обычно являются следующие:

- разработка сначала общих, а затем и полных требований к системе;
- разработка аванпроекта системы;
- разработка эскизного проекта;

- разработка технического проекта;
- создание опытного образца ВС;
- проведение испытаний опытного образца;
- изготовление головного образца системы и ввод его в эксплуатацию;
- опытная эксплуатация и доработка головного образца;
- выпуск и ввод в эксплуатацию серийных образцов;
- модернизация системы.

В зависимости от того, в какой степени при проектировании ВС используются готовые или известные технические и методические решения, некоторые из этих этапов могут опускаться. Например, могут отсутствовать этапы разработки аванпроекта и эскизного проекта. В случае, когда проектируемая система является уникальной и дорогостоящей, создается один ее образец, который после испытаний, опытной эксплуатации и доработки принимается на постоянную эксплуатацию. Если проектируемая система не является уникальной, но достаточно дорогостояща, а на ввод в эксплуатацию серийных образцов даются сжатые сроки, функции головного образца системы может выполнять опытный образец.

Ниже при изложении содержания этапов проектирования ВС принимается во внимание, что достаточно полное и глубокое решение задач синтеза и анализа системы (поскольку, синтезируя систему, приходится анализировать, оценивать промежуточные и окончательные варианты ВС) аналитическими методами в большинстве случаев невозможно. Задача синтеза ВС сводится к решению задач построения математических моделей системы по выбранным структурным схемам, анализа этих моделей, оптимизации параметров ВС по выбранным критериям качества.

В математической модели оптимизации параметров системы присутствуют блоки анализа, оценки качества функционирования системы.

Остановимся на содержании отдельных этапов проектирования ВС.

Разработка технических требований к системе заключается в следующем:

- определяется назначение системы и выбираются критерии ее эффективности, т. е. предъявляются требования к показателям качества системы, в наибольшей мере характеризующим степень соответствия ее своему назначению (выбор критериев эффективности накладывает заметный отпечаток на все дальнейшее проектирование ВС);

- устанавливается перечень задач, которые должна решать система. По каждой задаче определяются параметры, необходимые разработчику системы (об информации, которая должна сопровождать каждую задачу, см. ниже);

- разрабатываются алгоритмы решаемых задач, что необходимо для постановки достаточно полных и точных требований к ВС;

— определяются связи системы с источниками и потребителями информации: число каналов связи, их тип, пропускная способность, помехозащищенность;

— предъявляются требования к техническим, конструктивным и эксплуатационным параметрам системы.

По каждой задаче, решаемой на ВС, должны быть определены следующие характеристики:

1. Уровень приоритета задачи.
2. Допустимое время ожидания до начала решения задачи от момента появления запроса на решение.
3. Допустимое время решения.
4. Ожидаемая частота решения задачи (плотность потока заявок на решение данной задачи) и закон распределения времени поступления запросов на ее решение.
5. Вид информации (аналоговая, дискретная), представляющей собой исходные данные для решения задачи.
6. Требуемая точность решения.
7. Допустимость потери информации (исходных данных, промежуточных результатов решения задачи) при поступлении запроса на решение задачи более высокого приоритета.
8. Возмущающие факторы, влияющие на решение задачи (влияние помех в линиях связи при передаче исходной информации или запроса, ошибки оператора или пользователя и др.).
9. Стабилизирующие факторы (возможность исправления и контроля информации).
10. Характеристики выходной информации, представляющей собой результаты решения задачи: вид ее представления (аналоговый, дискретный), форма регистрации и документирования, длительность хранения.

Задачи аванпроектирования ВС: разработка структурной схемы с детализацией до элементов ВС; уточнение алгоритмов решаемых задач; разработка аппаратно-программной системы управления (в первом приближении); выбор дисциплины обслуживания заявок на решение задач; определение мер по повышению надежности и точности решения задач; разработка предложений по математическому обеспечению ВС; предварительная оценка эффективности функционирования ВС выбранной структуры.

Для решения этих задач формализуется процесс функционирования системы выбранной структуры, составляется ее математическая модель, составляется и реализуется на универсальной ЭВМ моделирующий алгоритм.

Исследование системы на ее математической модели дает возможность получить информацию, на основе которой принимается далее решение о выборе наиболее рационального варианта системы.

Задачи эскизного проектирования ВС: уточнение структурной схемы системы; разработка структурных схем элементов системы, если эти элементы представляют собой радикально новую кон-

струкцию, или выбор типов элементов из числа имеющихся, если последние удовлетворяют поставленным требованиям; разработка принципиальных схем отдельных узлов и блоков ВС, построение макетов этих узлов и блоков, проведение экспериментальных исследований; уточнение и детализация вопросов, связанных с построением аппаратно-программной системы управления ВС, с организацией обслуживания запросов на решение задач; разработка и обоснование предложений по организации технического обслуживания системы; разработка математического обеспечения системы и отдельных ее машин (процессоров); дальнейшая проработка вопросов по обеспечению необходимой надежности работы ВС и точности решения задач; оценка эффективности функционирования ВС; общая оценка эффективности системы с учетом затрат на ее проектирование, производство и эксплуатацию.

Решение ряда вопросов эскизного проектирования ВС производится, как и в случае аванпроектирования, путем исследования математической модели системы. Моделирующий алгоритм процесса функционирования системы, составленный на этапе аванпроектирования, уточняется с учетом изменений в структуре системы и организации ее работы или составляется заново, если эти изменения слишком значительны.

Работа по составлению эскизного проекта системы, как, впрочем, и при составлении аванпроекта, технического проекта, ведется в тесном взаимодействии двух коллективов — разработчиков алгоритмов решаемых задач (алгоритмистов) и собственно проектировщиков ВС. Последние получают от первых информацию, которая, в частности, должна включать по каждой задаче следующие сведения: временную диаграмму решения задачи, включая обращения к внешним абонентам, для детерминированных процессов и законы распределения времени начала решения задачи и обращения к внешним абонентам для случайных процессов; требуемые интервалы времени для реализации алгоритма соответствующей задачи; необходимую точность вычислений промежуточных и окончательных результатов; статистику распределения основных команд при реализации алгоритма на универсальной ЭВМ определенного типа; наиболее употребительный перечень стандартных подпрограмм и средние данные по частоте обращения к ним; связность алгоритмов, оцениваемую объемом оперативной памяти, необходимой для хранения исходных данных, промежуточных и окончательных результатов; количество и разрядность констант; объем информации, выдаваемой на соответствующие внешние устройства (ВЗУ, печатающие, перфорирующие устройства) и в каналы связи внешним абонентам; условия допустимости прерывания реализации алгоритма и допустимые интервалы прерывания; требуемые надежность и достоверность решения задачи за определенное время; объем информации, которую необходимо хранить длительное время.

Задачи технического проектирования ВС: уточнение, детализация и дальнейшая проработка всех вопросов, которые решались на этапе эскизного проектирования; разработка принципиальных схем элементов, представляющих собой новые конструкции; полная конструктивная разработка системы; изготовление рабочих чертежей.

Перед изготовлением опытного образца системы производится детальное исследование принятого варианта методом моделирования. Анализ результатов моделирования позволяет вскрыть слабые места проекта, оценить согласованность отдельных элементов. Такое исследование дает возможность внести окончательные коррективы в проект, улучшить принятый вариант системы.

Основная задача этапа испытания опытного образца системы— это оценка соответствия фактических характеристик разработанной ВС заданным проектным характеристикам.

В результате испытаний системы выявляются: работоспособность, надежность и помехозащищенность технической части системы в заданных условиях эксплуатации; устойчивость функционирования в аварийных режимах; поведение системы при реализации альтернативных путей решения задач; соответствие фактического потребления энергии и других материальных ресурсов заданным; фактическое значение показателей качества системы и, в частности, показателей ее эффективности.

В ходе испытаний ВС выясняется необходимость и целесообразность проведения различных доработок и усовершенствований как технической части системы, так и ее математического обеспечения, что найдет отражение при создании головного образца системы или серийной партии.

18.9. ПУТИ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ И ЭФФЕКТИВНОСТИ ЭВМ И ВС

Превосходство по техническим параметрам и эксплуатационным характеристикам вновь создаваемых ЭВМ по сравнению с предшествующими образцами, как правило, достигается за счет улучшения элементной и технологической баз, новых структурных решений, создания принципиально новых периферийных устройств (особенно тех, которые участвуют в обеспечении дистанционного обмена информацией между абонентами и ЭВМ), совершенствования математического обеспечения. Повышение производительности и эффективности использования ЭВМ существенно зависит от улучшения таких характеристик, как надежность, быстродействие оперативной памяти и процессора, емкость оперативной памяти, скорость работы ВЗУ.

Повышение надежности ЭВМ достигается как совершенствованием традиционных способов (дублирование, резервирование, аппаратный и программный контроль, повышение надежности

элементной базы), так и реализацией новых способов. К числу последних относится применение корректирующих кодов, обеспечивающих обнаружение и исправление ошибок. Другой способ — использование блочного резервирования с самонастройкой. В случае отказа какого-нибудь блока осуществляется перераспределение функций блоков таким образом, чтобы обеспечивалось выполнение основных функций, т. е. уровень резервирования для основных функций выше, чем для второстепенных.

Проблема надежности ЭВМ кардинально решается структурным способом. Повышение надежности только элементной базы задачу не решает по следующей причине. Увеличение надежности элементов пропорционально повышению уровня их интеграции. При переходе к большим интегральным схемам надежность логического элемента возрастает на 1—3 порядка. Но, с другой стороны, при построении более сложных ЭВМ (а каждое последующее поколение ЭВМ значительно сложнее предыдущего) растет и число необходимых элементов на 1—3 порядка. В результате заметного выигрыша в надежности не получается.

Значительное повышение быстродействия основных устройств ЭВМ и машины в целом может быть достигнуто за счет использования быстродействующих элементов, работа которых основана на освоении новых физических принципов и эффектов. Перспективным в этом отношении является эффект Джозефсона, который может быть использован и в запоминающих устройствах, и в логических схемах.

Настоящей революцией в электронной вычислительной технике явилось бы создание недорогой оперативной памяти (ОП) большой емкости — порядка 10^{10} — 10^{12} байт *. От емкости и быстродействия ОП в наибольшей степени зависят характеристики ЭВМ. В настоящее время к ОП высокопроизводительных ЭВМ и ВС выдвигаются такие требования **: по емкости 10^8 бит, по времени цикла обращения 200 нс. В ближайшем будущем потребуется ОП емкостью 10^9 бит с циклом обращения 80—50 нс. Строить ОП с такими характеристиками на основе современной полупроводниковой технологии и схемотехники нецелесообразно из-за большой потребляемой мощности (около 50 кВт для ОП емкостью 10^8 бит), высокой стоимости (более 2 млн. руб.) и низкой по сравнению с процессором надежности**. Один из возможных путей решения проблемы — создание оптоэлектронной оперативной памяти, обладающей плотностью записи информации 10^5 — 10^7 бит/см² и относительной технологической простотой изготовления блоков хранения информации и адресной коммутации элементов ОП. Создание такой памяти сдерживается отсутствием в настоящее время соответствующих носителей информации.

* Глушков В. М. Проблемная ориентация и другие пути повышения эффективности ЭВМ. — «Вестник АН СССР» 1975, № 3, с. 18—24.

** Бурцев В. С. Перспективы создания ЭВМ высокой производительности. — «Вестник АН СССР», 1975, № 3, с. 25—31.

Противоречивость требований к быстродействию ЗУ и его большой емкости разрешается путем создания многоуровневых иерархических запоминающих систем с сегментной и страничной организацией памяти. Такая тенденция сохранится и в дальнейшем. Для согласования ОП и ВЗУ больших ЭВМ и ВС возникла необходимость в промежуточном ЗУ на 10^9 бит и временем цикла 1—5 мкс. В настоящее время такое ЗУ может быть реализовано на ферритовых сердечниках с временем цикла 2—3 мкс. В перспективе может быть создано оптоэлектронное ВЗУ, ВЗУ на цилиндрических магнитных доменах (последние позволяют на одной пластинке размером 10 см^2 разместить до 10^7 бит информации при темпе ее передачи 10^7 бит/с).

Развитие периферийной аппаратуры направлено на дальнейшее улучшение ее характеристик, на обеспечение упрощения общения человека с машиной.

В тесной взаимосвязи с развитием и совершенствованием технических средств ЭВМ и ВС развивается их математическое обеспечение: операционные системы, системы автоматизации программирования, системы контролирующих и диагностических тест-программ.

Основные направления развития операционных систем (ОС):
1. Углубление и расширение ориентации на пользователя, а не на программиста, обеспечение работы ЭВМ и ВС с удаленными абонентами в различных режимах, и особенно в диалоговом режиме, представляющем собой двустороннюю обратную связь ЭВМ (ВС) и пользователя в реальном масштабе времени. В диалоговом режиме обеспечивается непосредственное и эффективное взаимодействие системы и человека в процессе решения конкретных задач, при программировании и отладке задач, при управлении функционированием системы,

2. Дальнейшее развитие модульности структуры ОС.

3. Внедрение методов аппаратной реализации отдельных модулей ОС (в порядке перераспределения функций между аппаратными и программными средствами ЭВМ). К их числу относятся часто встречающиеся подпрограммы преобразования форматов данных, подпрограммы распределения оперативной памяти и др. Внедрение этих методов возможно благодаря достижениям интегральной технологии. Оно способствует повышению производительности ЭВМ и ВС.

4. Построение ОС по многоуровневому (иерархическому) принципу, в соответствии с которым модули ОС размещаются на различных уровнях так, что функции, выполняемые модулями на нижестоящих уровнях, являются дальнейшим развитием и детализацией функций модулей вышестоящих уровней. Такой принцип в сочетании с аппаратной реализацией модулей ОС нижних уровней дает возможность формулировать задачи ОС в широких категориях и, следовательно, использовать языки более высокого уровня. Это приводит к заметному уменьшению требуемого объема памяти для размещения модулей ОС.

5. Развитие самонастраивающихся, адаптивных ОС; создание супервизоров, обеспечивающих максимальную производительность ВС применительно к конкретным условиям работы, а в более далекой перспективе — способных предсказывать нежелательные ситуации и предотвращать их появление.

6. Разработка новых компонентов ОС, обеспечивающих сбор, обработку и анализ статистических данных о работе как отдельных модулей, так и всей системы в целом.

Первоочередными задачами в развитии математического обеспечения применительно к машинам и системам четвертого поколения являются следующие: усовершенствование реализаций принципов виртуальной памяти, используемых в ЭВМ третьего поколения; усовершенствование методов организации файлов; реализация совершенных методов обеспечения защиты массивов данных и программ; широкое применение аппаратных способов реализации ряда программных функций с использованием логических схем с памятью; автоматическое измерение ряда характеристик функционирования ВС, учет и регистрация показателей использования ресурсов системы каждым пользователем; разработка и отладка системы программного обеспечения в интерактивном режиме (в ЭВМ третьего поколения уже используются методы отладки программ в режиме диалога); разработка входного языка, который должен быть достаточно простым, доступным для изучения и эффективным при использовании в режиме диалога.

В связи с значительным увеличением сложности и стоимости математического обеспечения ЭВМ и ВС возникла проблема повышения его надежности. Решение этой проблемы будет способствовать повышению эффективности функционирования систем.

Одним из реальных путей повышения эффективности вычислительных средств является разумная специализация технико-математических комплексов (ТМК), определяемая их проблемной ориентацией. Специализация здесь затрагивает главным образом периферийное оборудование и математическое обеспечение (процессоры могут быть универсального типа). Она заключается в том, что вычислительные системы (т. е. ТМК) ориентируются на разные классы применений. Целесообразно создавать следующие типы проблемно-ориентированных ТМК:

— ТМК АСУ, для которого необходимо иметь специальный набор периферийных устройств, периферийные мини-ЭВМ для первичной обработки информации, развитые операционные системы, допускающие создание любых временных конфигураций элементов АСУ и эффективное управление их работой, а также управление распределенными банками данных;

— ТМК, ориентированный на обслуживание запросов удаленных абонентов в интерактивном режиме;

— ТМК, ориентированный на автоматизацию управления технологическими процессами, сбора и обработки данных, испытания сложных технических объектов; здесь специализация про-

является в наборе специальной аппаратуры (управляющие мини- и микро-ЭВМ, устройства сопряжения с объектами управления, интерфейсы с датчиками и исполнительными механизмами, устройства для обеспечения стандартной формы представления аналоговой информации) и в пакетах программ для первичной обработки данных;

— ТМК для подготовки данных и группового цифрового программного управления станками и другим оборудованием (здесь нужны специфические программно-языковые средства);

— ТМК, ориентированные на решение справочно-информационных задач; здесь важно иметь ЗУ большой емкости и обеспечить разумное сочетание машинной информации с системами микрофильмования;

— ТМК, ориентированные на автоматизацию проектно-конструкторских работ;

— ТМК, ориентированные на задачи автоматизации перевода и редактирования и др.

Новый, более совершенный этап в использовании электронной вычислительной техники представляет собой организация вычислительных сетей, или сетей ЭВМ. Вычислительная сеть—это совокупность независимых, связанных между собой ЭВМ (или ВС совмещенного типа), ориентированных на совместное использование таких общесетевых ресурсов, как машинное время, память, программы, массивы данных. В вычислительной сети осуществляется перемещение, хранение и переработка информации по запросам пользователей, имеющих доступ к сети и удаленных от центров обработки, где размещаются ЭВМ, на те или иные расстояния.

Для вычислительных сетей принципиальное значение имеют следующие обстоятельства:

— ЭВМ сети, находящиеся в разных центрах обработки, связываются между собой **автоматически**; именно в этом заключается основная сущность процессов, протекающих в сети;

— каждая ЭВМ сети должна быть приспособлена, с одной стороны, для работы в автономном режиме под управлением своей ОС, а, с другой стороны, должно обеспечиваться ее участие в работе всей сети совместно с другими ЭВМ под управлением ОС высшего уровня.

С появлением вычислительных сетей удастся разрешить две очень важные проблемы: проблему обеспечения в принципе неограниченного доступа к ЭВМ потребителям независимо от их территориального расположения и проблему оперативного перемещения больших массивов информации на любые расстояния, что позволяет своевременно получать данные, необходимые для принятия тех или иных решений.

ПРИЛОЖЕНИЕ

Т а б л и ц а 1

Внешние запоминающие устройства и устройства ввода-вывода ЕС ЭВМ

Шифр	ВЗУ и УВВ		Способ подключения к ЭВМ	Устройство управления			Каналы ЭВМ			Тип операционной системы
	Наименование	Шифр		Тип УУ	Количество на одно УУ	ЕС-1020	ЕС-1030	ЕС-1050		
ЕС-5010	Накопитель на магнитных лентах (НМЛ) . . .		ЕС-5511	Раздельный	8	МК, СК	МК, СК		ДОС	
ЕС-5013	НМЛ с автоматической сменной кассет		ЕС-5513	То же	8	МК, СК	МК, СК	МК, СК	ОС	
ЕС-5014	НМЛ		ЕС-5514	»	8	СК	МК, СК	МК, СК	ОС	
ЕС-5017	НМЛ		ЕС-5517	»	8	МК, СК	МК, СК	МК, СК	ДОС, ОС	
ЕС-5018	НМЛ		ЕС-5511	»	8	МК, СК	МК, СК	МК, СК	ДОС, ОС	
ЕС-5033	Накопитель на магнитном барабане (НМБ)		ЕС-5533	»	8	—	—	СК	ОС	
ЕС-5051	Накопитель на магнитных дисках (НМД) . . .		ЕС-5551	»	8	СК	МК, СК	МК, СК	ОС	
ЕС-5056	Накопитель на сменных магнитных дисках		ЕС-5551	»	8	СК	МК, СК	МК, СК	ДОС, ОС	
ЕС-5511	Устройство управления НМЛ		К каналу	Раздельный	8	МК, СК	МК, СК	МК, СК	ДОС, ОС	
ЕС-5513	Устройство управления НМЛ		То же	То же	8	МК, СК	МК, СК	МК, СК	ОС	
ЕС-5514	Устройство управления НМЛ		»	»	8	С	МК, СК	МК, СК	ОС	
ЕС-5533	Устройство управления НМБ		»	»	8	—	—	СК	ОС	
ЕС-6012	Устройство ввода с перфокарт		»	Совмещенный	1	МК, СК	МК, СК	МК, СК	ОС	

ЕС-6013	Устройство ввода с перфокарт	То же	То же	1	—	—	ОС
ЕС-601	Устройство ввода с перфокарт	»	»	1	—	—	ОС
ЕС-6022	Устройство ввода с перфоленг	»	»	1	МК, СК	МК, СК	ОС, ДОС
ЕС-7010	Устройство вывода на перфокарты	»	»	1	МК, СК	МК, СК	ДОС
ЕС-7012	Устройство вывода на перфокарты	»	»	1	МК, СК	МК, СК	ОС
ЕС-7022	Устройство вывода на перфоленг	»	»	1	МК, СК	МК, СК	ОС
ЕС-7030	Алфавитно-цифровое печатающее устройство	»	»	1	МК, СК	МК, СК	—
ЕС-7032	Алфавитно-цифровое печатающее устройство	»	»	1	МК, СК	МК, СК	—
ЕС-7051	Графическое регистрирующее устройство планшетного типа	»	»	1	МК	МК, СК	ОС
ЕС-7052	Графическое регистрирующее устройство рулонного типа	То же	То же	1	МК	МК	ОС
ЕС-7053	Графическое регистрирующее устройство рулонного типа	»	»	1	МК	МК	ОС
ЕС-7062	Выносной пульт для ввода-вывода алфавитно-цифровой информации на ЭЛТ	К каналу	К каналу	4	МК, СК	МК, СК	ДОС, ОС
ЕС-7064	Выносной пульт для ввода-вывода графической и алфавитно-цифровой информации на ЭЛТ	То же	То же	1	МК, СК	МК, СК	ДОС

ВЗУ и УВВ		Способ подключения к ЭВМ	Устройство управления		Каналы ЭВМ			Тип операционной системы
Шифр	Наименование		Тип УУ	Количество на одно УУ	ЕС-1020	ЕС-1030	ЕС-1050	
ЕС-7066	Выносной пульт для ввода-вывода алфавитно-цифровой информации на ЭЛТ	К 7566	Раздельный	16	МК, СК	МК, СК	МК, СК	ДОС, ОС
ЕС-7070	Пишущая машинка с блоком управления	К каналу	Совмещенный	1	МК, СК	МК, СК	МК, СУ	ДОС, ОС
ЕС-7566	Устройство группового управления выносными пультами	То же	Раздельный	16	МК, СК	МК, СК	МК, СК	ДОС, ОС
ЕС-8401	Мультиплексор передачи данных МПД-1	»	То же	15	МК, СК	МК, СК	МК	ДОС, ОС
ЕС-8402	Мультиплексор передачи данных МПД-2	»	»	176	МК, СК	МК, СК	МК	ДОС, ОС
ЕС-8501	Абонентский пункт с пишущей машинкой, ленточным перфоратором	Через ЕС-8401	»	15	МК, СК	МК, СК	МК	ДОС, ОС
ЕС-8563	Абонентский пункт с ЭЛТ	Через ЕС-8402	»	176	МК, СК	МК, СК	МК	ДОС, ОС
ЕС-8570	Абонентский пункт с пишущей машинкой	Через ЕС-8401 ЕС-8402	» »	15 176	МК, СК МК, СК	МК, СК МК, СК	МК МК	ДОС, ОС ДОС, ОС

Т а б л и ц а 2

**Минимальный состав технических средств,
необходимых для ДОС ЕС и ОС ЕС**

Наименование устройств	Шифр устройства	Количество устройств	Тип операционной системы
Процессор с основной памятью емкостью:			
64К байт		1	ДОС; ОС-1
128К байт		1	ОС-2
256К байт		1	ОС-3
Накопитель на сменных магнитных дисках	ЕС-5056	2 3 3	ДОС, ОС-1 ОС-2 ОС-3
Устройство управления для накопителей на магнитных дисках и барабанах	ЕС-5551	1	ДОС, ОС
Устройство ввода информации с перфокарт	ЕС-6012	1	ДОС, ОС
Устройство вывода информации на перфокарты	ЕС-7012	1	ДОС, ОС
Алфавитно-цифровое печатающее устройство	ЕС-7032	1	ДОС, ОС
Пишущая машинка с блоком управления	ЕС-7070	1	ДОС, ОС

Т а б л и ц а 3

Основные характеристики моделей ЭВМ Единой системы

Характеристики	Модели ЭВМ					
	ЕС-1010	ЕС-1021	ЕС-1020	ЕС-1030	ЕС-1040	ЕС-1050
Среднее быстродействие процессора (тыс. оп/с)	10	15	15—20	100	200—300	500
Разрядность процессора (байт)	1	1	1	4	8	8
Основная оперативная память:						
емкость (К байт)	8—64	16—64	64—256	128—512	256—1024	256—1204
время цикла (мкс)	1,0	2,0	2,0	1,25	1,35	1,25
время выборки (мкс)	0,5	1,0	1,0	0,75	0,45	0,8
разрядность (байт)	1	1	2	4	8	8
Емкость ВЗУ (базовый комплект):						
на МД (Мбайт) . . .	1·7,25	1·7,25	2·7,25	2·7,25	2·7,25	4·7,25
на МЛ (Мбайт) . . .	2·20		4·20	8·20	8·20	16·20
Мультиплексный канал:						
число подканалов	127	16	48—112	256	128—256	196
скорость в мультиплексном режиме (кбайт/с)	30	35	До 16	40	20—25	110—670

Характеристики	Модели ЭМВ					
	ЕС-1010	ЕС-1021	ЕС-1020	ЕС-1030	ЕС-1040	ЕС-1050
скорость в селек- торном режиме (кбайт/с)	140	220	До 100	300	180—720	180—720
число подключае- мых блоков управ- ления	32	10	8	8	10	40
Селекторные каналы: максимальное число каналов	1	2	2	3	6	6
скорость канала (кбайт/с)	50—250	250	200	600—900	1300	1300
число подключае- мых блоков управ- ления	32	10	8	8	10	8
Потребляемая мощность (кВА)	2,0	13	21	25	65	70
Занимаемая площадь (м ²)	25	50	50	110	150	230
Разработчик	ВНР	ЧССР	СССР	СССР	ГДР	СССР

Т а б л и ц а 4

Основные характеристики устройств ввода-вывода ЭВМ Единой системы

Наименование	Шифр	Основные характеристики
Устройство ввода с пер- фокарт	ЕС-6012	Скорость — 500 карт/мин, режим непре- рывный, ввод узкой стороной, механизм УВБК-600, $P_{\text{потр}} = 1$ кВт, габариты 1200×500×1220, масса — 300 кг
Устройство ввода с пер- фокарт	ЕС-6013	Скорость — 1000 карт/мин, режим старт- стопный, $P_{\text{потр}} = 1,5$ кВт, габариты 1000×700×1000, масса — 350 кг
Устройство ввода с пер- фокарт	ЕС-6014	Скорость — 2000 карт/мин в непрерыв- ном режиме, 1500 карт/мин в старт- стопном режиме, $P_{\text{потр}} = 3$ кВт, габари- ты 1450×780×1010, масса — 450 кг
Устройство ввода с пер- фокарт	ЕС-6111	Скорость — 50 колонок/с, режим старт- стопный с реверсом, $P_{\text{потр}} = 0,8$ кВт, габариты 600×300×260, масса — 150 кг (для абонентских пунктов)
Устройство ввода с пер- фокарт	ЕС-6022	Скорость — 1500 строк/с, режим старт- стопный (механизм FS-1501)
Устройство ввода с пер- фокалент	ЕС-6025	Скорость — 1500 строк/с, режим старт- стопный, $P_{\text{потр}} = 0,4$ кВт, габариты 1200×500×1320, масса — 200 кг

Наименование	Шифр	Основные характеристики
Устройство вывода на перфокарты	ЕС-7012	Скорость — 250 карт/мин (механизм ПЭМ-80), $P_{\text{потр}} = 2$ кВт, габариты $1500 \times 800 \times 1200$, масса — 350 кг
Устройство вывода на перфокарты	ЕС-7011	Скорость — 160 колонок/с, $P_{\text{потр}} = 1,5$ кВт, габариты $1300 \times 600 \times 1300$, масса — 400 кг (для абонентских пунктов)
Устройство вывода на перфоленту	ЕС-7022	Скорость — 150 строк/с (механизм ПЛ-150), $P_{\text{потр}} = 0,8$ кВт, габариты $1200 \times 500 \times 1245$, масса — 200 кг
Устройство вывода на перфоленту	ЕС-7023	Скорость — 200 строк/с, $P_{\text{потр}} = 0,5$ кВт, габариты $340 \times 280 \times 250$, масса — 40 кг (для абонентских пунктов)
Алфавитно-цифровое печатающее устройство	ЕС-7032	Скорость — 900 строк/мин (механизм АЦПУ-128-6), $P_{\text{потр}} = 3,5$ кВт, габариты $1420 \times 670 \times 1430$, масса — 500 кг
Устройство для вывода графиков	ЕС-7051	Вывод на планшет $1,0 \times 1,05$ м ² (механизм ДГУ-4), точность — 0,05 мм, скорость 80 мм/с, количество символов — 255, три цвета, $P_{\text{потр}} = 2$ кВт, габариты $1600 \times 1800 \times 900$, масса — 700 кг
Устройство для вывода графиков	ЕС-7052	Вывод на рулон 390×80 000 (механизм ДГУ-22), точность — 0,2 мм, скорость — 200 мм/с, количество символов — 64, три цвета, $P_{\text{потр}} = 1,2$ кВт, габариты $600 \times 1600 \times 1100$, масса — 400 кг
Устройство для вывода графиков	ЕС-7053	Вывод на рулон 800×20 000 (механизм ДГУ-21), точность — 0,1 мм, скорость — 150 мм/с, количество символов — 253, три цвета, $P_{\text{потр}} = 1,5$ кВт, габариты $600 \times 1600 \times 1100$
Выносной пульт с пишущей машинкой	ЕС-7070 ЕС-7571	Операторский пульт на основе ЭПМ «Консул-260», скорость — 10 зн/с, количество символов — 92
Выносной пульт для ввода-вывода алфавитно-цифровой информации на ЭЛТ	ЕС-7062	Количество знаков на экране — 960, количество символов — 64, экран 150×200 , $P_{\text{потр}} = 1,5$ кВт
Выносной пульт для ввода-вывода алфавитно-цифровой и графической информации на ЭЛТ	ЕС-7064	Количество знаков на экране — 1024, количество символов — 96, экран 250×250 , световой карандаш, габариты $1200 \times 1700 \times 750$ и $1500 \times 1100 \times 1200$ (УУ)

Основные характеристики абонентских пунктов ЕС ЭВМ

Наименование	Шифр	Основные характеристики
АП-1	ЕС-8501	Ввод-вывод с перфокарт и перфолент, вывод на ЭПМ, скорость — 200 бит/с, каналы связи: ТЛФ коммутируемые и некоммутируемые, Тлг некоммутируемые
АП-2	ЕС-8502	Ввод-вывод с перфоленты, скорость — 100 бит/с, каналы связи: Тлф коммутируемые и некоммутируемые, Тлг некоммутируемые
АП-3	ЕС-8503	Ввод-вывод с перфоленты, скорость — 600/1200 бит/с, каналы связи: Тлф коммутируемые и некоммутируемые
АП-4	ЕС-8504	Ввод-вывод с перфокарт и перфолент, вывод на ЭПМ и МЛ, скорость — 1200/2400 бит/с, каналы связи: Тлф некоммутируемые
АП-11	ЕС-8511	Вывод на АЦПУ, ввод и вывод с перфокарт, скорость — 600/1200, 1200/2400 бит/с, каналы связи: ТФЛ коммутируемые и некоммутируемые
АП-61	ЕС-8561	Имеет функциональную и алфавитно-цифровую клавиатуру, ЭЛТ емкостью 960 знаков, ЭПМ «Консул-260», скорость — 1200/2400 бит/с, каналы связи: Тлф некоммутируемые
АП-63	ЕС-8563	Имеет $8/16$ экранных пультов с функциональной и алфавитно-цифровой клавиатурой, ЭЛТ емкостью 480/240 знаков, ЭПМ «Консул-260», скорость — 1200/2400 бит/с, каналы связи: Тлф некоммутируемые
АП-70	ЕС-8570	Имеет функциональную и алфавитно-цифровую клавиатуру, ввод-вывод на ЭПМ «Консул-260», скорость — 100 бит/с, каналы связи: Тлф коммутируемые и некоммутируемые; Тлг некоммутируемые

Основные характеристики мультимплексов передачи данных ЕС ЭВМ

Наименование	Шифр	Основные характеристики
МПД-1А	ЕС-8400	Скорость передачи — 50, 70, 100, 1200, 2400, 4800 бит/с, типы линий связи: Тлг некоммутируемые, Тлг коммутируемые, Тлф некоммутируемые, физические линии до 10 км, количество одновременно работающих каналов связи — до 15. Типы АП: РТА-60, АП с ЭПМ «Консул», АП с ЭЛТ. Способ синхронизации — стартовый, код передачи — МТК-2, ГОСТ 13052-67, $P_{\text{потр}} = 2$ кВА
МПД-2	ЕС-8402	Скорость передачи — 50—4800 бит/с. Типы линий связи: Тлг некоммутируемые, Тлф коммутируемые и некоммутируемые. Количество одновременно работающих каналов связи — до 175. Способ обмена — двусторонний, полудуплексный

Наименование	Ш.ИФ	Основные характеристики
МПД-3	ЕС-8403	Скорость передачи: Тлг некоммутируемые — 50 бит/с, Тлф коммутируемые — 200, 600, 1200 бит/с, Тлф некоммутируемые — 2400, 4800 бит/с, широкополосный — до 48 000 бит/с. Количество одновременно работающих каналов связи — 4. Способ обмена — двусторонний, полудуплексный

Таблица 7

Графические знаки

Знак	Наименование	Знак	Наименование
[Квадратная скобка (левая)		Логическое <i>ИЛИ</i> (черта вертикальная)
.	Точка	.	Запятая
<	Меньше	%	Процент
(Круглая скобка (левая)	—	Подчеркивание
+	Плюс	>	Больше
!	Восклицательный знак	?	Вопросительный знак
&	Коммерческое <i>И</i>	^	Диакритический знак
]	Квадратная скобка (правая)	:	Двоеточие
⌘	Знак денежной единицы	#	Номер
*	Звездочка	@	Коммерческое ЭТ
)	Круглая скобка (правая)	'	Апостроф
;	Точка с запятой	=	Равно
∇	Логическое <i>НЕ</i>	"	Кавычки
-	Минус	—	Черта сверху
/	Дробная черта	{	Фигурная скобка (левая)
		}	Фигурная скобка (правая)

Стандартные управляющие символы

Позиция в кодовой таблице ДКОИ	Сокращенное обозначение символа		Наименование символа	
	русское	между- народное	русское	международное
0/0	ПУС	NUL	Пусто	Null
0/1	НЗ	SOH	Начало заголовка	Start of Heading
0/2	НТ	STX	Начало текста	Start of Text
0/3	КТ	ETX	Конец текста	End of Text
0/5	ГТ	HT	Горизонтальная табуляция	Horisontal Tabulation
0/7	ЗБ	DEL	Забой	Delete
0/В	ВТ	VT	Вертикальная та- буляция	Vertical Tabulation
0/С	ПФ	FF	Перевод формата	Form Feed
0/D	ВК	CR	Возврат каретки	Carriage Return
0/E	ВЫХ	SO	Выход	Shift—out
0/F	ВХ	SI	Вход	Shift—in
1/0	АР1	DLE	Авторегистр 1	Data Link Escape
1/1	СУ1	DC1	} Символы управления	Device Control
1/2	СУ2	DC2		
1/3	СУ3	DC3		
1/6	ВШ	BS	Возврат на шаг	Backspace
1/8	АН	CAN	Аннулирование	Cancel
1/9	КН	EM	Конец носителя	End of Medium
1/С	РФ	FS	Разделитель фай- лов	File Separation
1/D	РГ	GS	Разделитель групп	Group Separation
1/E	РЗ	RS	Разделитель запи- сей	Record Separation
1/F	РЭ	US	Разделитель эле- ментов	Unit Separation
2/5	ПС	LF	Перевод строки	Line Feed
2/6	КБ	ETB	Конец блока	End of Transmission Block

Позиция в кодовой таблице ДКОИ	Сокращенное обозначение символа		Наименование символа	
	русские	между- народное	русское	международное
2/7	AP2	ESC	Авторегистр 2	Escape
2/D	KTM	ENQ	Кто там?	Enquiry
2/E	ДА	ACK	Подтверждение	Acknowledge
2/F	ЗВ	BEL	Звонок	Bell
3/2	СИН	SYN	Синхронизация	Synchronous Idle
3/7	КП	EOT	Конец передачи	End of Transmission
3/C	СТП	STOP	Стоп	Stop
3/D	НЕТ	NAK	Отрицание	Negative Acknowledge
3/F	ЗМ	SUB	Замена	Substitute

Т а б л и ц а 9

Дополнительные управляющие символы

Позиция в кодовой таблице ДКОИ	Сокращенное обозначение символа		Наименование символа	
	русское	между- народное	русское	международное
0/4	ВЫП	PF	Выключение перфоратора	Punch off
0/6	СБ	LC	Строчные буквы	Lower Case
1/4	ВСТ	RES	Восстановление	Restore
1/5	НС	NL	Новая строка	New Line
1/7	ОЖД	IL	Ожидание	Idle
1/A	УУК	CC	Управление указателем	Cursor Control
2/0	ВЦФ	DS	Выбор цифры	Digit Select
2/1	НЗН	SOS	Начало значимости	Start of Significance
2/2	РП	FS	Разделитель полей	Field Separation
2/4	БК	BYP	Блокировка	Bypass
2/A	УР	SM	Установление режима	Set Mode
3/4	ВКП	PN	Включение перфоратора	Punch On
3/5	ОУЧ	RS	Останов устройства чтения	Reader Stop
3/6	ПБ	UC	Прописная буква	Upper Case

Коды операций

1. Команды формата RR

		Операции			
Младшая тетрада кода операции	передачи управления, переключения состояния	с фиксированной запятой, производимые над полными словами (32-разрядными операндами)	с плавающей запятой, производимые над длинными операндами	с плавающей запятой, производимые над короткими (32-разрядными) операндами	
XXXX 0000	0000 XXXX	0001 XXXX Загрузка положительного Load Positive*; LPR**	0010 XXXX Загрузка положительного Load Positive (Long); LPDR	0011 XXXX Загрузка положительного Load Positive (Short); LPER	
0001		Загрузка отрицательного Load Negative; LNR	Загрузка отрицательного Load Negative (Long); LNDR	Загрузка отрицательного Load Negative (Short); LNER	
0010		Загрузка и проверка—Load and Test; LTR	Загрузка и проверка—Load and Test (Long);—LTDR	Загрузка и проверка—Load and Test (Short); LTER	
0011		Загрузка дополнения—Load Complement; LCR	Загрузка дополнения—Load Complement (Long); LCDR	Загрузка дополнения—Load Complement (Short); LCSR	
0100	Установка маски программы—Set Program Mask; SPM	И And; NR	Деление пополам—Halve (Long); HDR	Деление пополам—Halve (Short); HER	
0101	Передача управления с вратом—Branch and Link; BALR	Сравнение логическое—Compare Logical; CLR			
0110	Передача управления по счетчику—Branch on Count; BCTR	ИЛИ Or; OR			
0111	Передача управления по условию—Branch on Condition; BCR	Исключающее ИЛИ—Exclusive Or; XR			

		Операции		
Младшая тетрада кода операции		с фиксированной запятой, производимые над полными словами (32-разрядными операциями)	с плавающей запятой, производимые над длинными данными (64-разрядными) операциями	с плавающей запятой, производимые над короткими (32-разрядными) операциями
1000	Установка ключа памяти Set Storage Key; SSK	Загрузка—Load; LR	Загрузка — Load (Long); LDR	Загрузка — Load (Short); LER
1001	Чтение ключа памяти—Insert Storage Key; ISK	Сравнение—Compare; CR	Сравнение — Compare (Long); CDR	Сравнение — Compare (Short); CER
1010	Обращение к супервизору Supervisor Call; SVC	Сложение—Add; AR	Сложение (с нормализацией)—Add Normalized (Long); ADR	Сложение (с нормализацией) Add Normalized (Short); AER
1011		Вычитание—Subtract; SR	Вычитание (с нормализацией)—Subtract Normalized (Long); SDR	Вычитание (с нормализацией) — Subtract Normalized (Short); SER
1100		Умножение—Multiply; MR	Умножение—Multiply (Long); MDR	Умножение — Multiply (Short); MER
1101		Деление—Divide; DR	Деление — Divide (Long); DDR	Деление — Divide (Short); DER
1110		Сложение логическое—Add Logical; ALR	Сложение (без нормализации)—Add Unnormalized (Long); AWR	Сложение (без нормализации)—Add Unnormalized (Short); AUR
1111		Вычитание логическое—Subtract Logical; SLR	Вычитание (без нормализации)—Subtract Unnormalized (Long); SWR	Вычитание (без нормализации)—Subtract Unnormalized (Short); SUR

2. Команды формата R X

Продолжение табл. 10

		Операции		
Младшая тетрада кода операции	с фиксированной запятой, производимые над полными словами (16-разрядными операндами) и передачи управления	с фиксированной запятой, производимые над полными словами (32-разрядными операндами)	с плавающей запятой, производимые над длинными (64-разрядными) операндами	с плавающей запятой, производимые над короткими (32-разрядными) операндами
↓ XXXX 0000	$\overbrace{0100 \text{ XXXX}}$ Запись в память—Store Halfword; SHN	0101 XXXX Запись в память—Store; ST	0110 XXXX Запись в память—Store (Long); STD	0111 XXXX Запись в память—Store (Short); STE
0001	Загрузка адреса — Load Address; LA			
0010	Запись символа в память Store Character; STC			
0011	Чтение символа — Insert Character; IC			
0100	Выполнение команды—Execute; EX	И And; N Сравнение логическое Compare Logical; CL		
0101	Передача управления с возвратом—Branch and Link; BAL			
0110	Передача управления по счетчику — Branch on Count; BCT	ИЛИ Or; O		
0111	Передача управления по условию—Branch on Condition; BC	Исключающее ИЛИ Exclusive Or; X		
1000	Загрузка полуслова—Load Halfword; LH	Загрузка—Load; L	Загрузка — Load (Long); LD	Загрузка — Load (Short); LE
1001	Сравнение полуслова—Compare Halfword; CH	Сравнение—Compare; C	Сравнение—Compare (Long); CD	Сравнение—Compare (Short); CE
1010	Сложение полуслова—Add Halfword; AH	Сложение—Add; A	Сложение (с нормализацией)—Add Normalized (Long); AD	Сложение (с нормализацией)—Add Normalized (Short); AE

Операции			
Младшая тетрада кода операции	с фиксированной запятой, производимые над полными словами (16-разрядными операндами) и передаче управления	с фиксированной запятой, производимые над полными словами (32-разрядными операндами)	с плавающей запятой, производимые над длинными (64-разрядными) операндами
1011	Вычитание полуслова—Subtract Halfword; SH	Вычитание—Subtract; S	Вычитание (с нормализацией)—Subtract Normalized (Short); SE
1100	Умножение полуслова—Multiply Halfword; MH	Умножение—Multiply; M	Умножение — Multiply (Short); ME
1101		Деление—Divide; D	Деление — Divide (Short); DE
1110	Преобразование в десятичную—Convert to Decimal; CVD	Сложение логическое—Add Logical; AL	Сложение (без нормализации)—Add Unnormalized (Short); AU
1111	Преобразование в двоичную—Convert to Binary; CVB	Вычитание логическое—Subtract Logical; SL	Вычитание (без нормализации)—Subtract Unnormalized (Short); SU

3. Команды формата RS, SI

Операции			
Младшая тетрада кода операции	передачи управления, переключения состояния и сдвига	с фиксированной запятой, логические и ввода-вывода	Операции
↓ XXXX 0000 0001	1000 XXXX Установка маски системы Set System Mask; SSM	1001 XXXX Запись в память групповая Store Multiply; STM Проверка по маске—Test under Mask; TM	1010 XXXX 1011 XXXX

Младшая тетрада кода операции	Операции	
	передачи управления, переключения состояния и сдвига	с фиксированной запятой, логические и ввода-вывода
0010	Загрузка ССП (слова состояния программы) — Load Program Status Word; LPSW Диагностика — Diagnose	Пересылка непосредственная — Move Immediate; MVI
0011	Прямая запись — Write Direct; WRD Прямое чтение — Read Direct; RDD	Проверка и установка — Test and Set; TS И непосредственное — And Immediate; NI Сравнение непосредственное — Compare Logical Immediate; CLI ИЛИ непосредственное Or Immediate; OI
0110	Передача управления по индексу «большее» — Branch on Index High; BXH	Исключающее ИЛИ непосредственное — Exclusive Or Immediate; XI
0111	Передача управления по индексу «меньше» или «равно» — Branch on Index Low or Equal; BXLE	Загрузка групповая — Load Multiply; LM
1000	Логический сдвиг слова вправо — Shift Right Single Logical; SRL	
1001	Логический сдвиг слова влево — Shift Left Single Logical; SLL	
1010	Арифметический сдвиг слова вправо — Shift Right Single Arithmetic; SRA	
1011	Арифметический сдвиг слова влево — Shift Left Single Arithmetic; SLA	

Операции	
Младшая тетрада кода операции	передачи управления, переключения состояния и сдвига
1100	Логический сдвиг двойного слова вправо—Shift Right Double Logical; SRDL
1101	Логический сдвиг двойного слова влево—Shift Left Double Logical; SLDL
1110	Арифметический сдвиг двойного слова вправо—Shift Right Double Arithmetic; SRDA
1111	Арифметический сдвиг двойного слова влево—Shift Left Double Arithmetic; SLDA
4. Команды формата SS	
Операции	
Младшая тетрада кода операции	логические и пересыльные над полями переменной длины
↓ XX XX 0000 0001 0010	Пуск ввода-вывода—Start I/O; SIO Отпрос ввода-вывода—Test I/O; TIO Останов ввода-вывода — Halt I/O; HIO Отпрос канала — Test Channel; TCH
↓ 1100 XXXX 0000 0001 0010	Пересылка цифр — Move Numerics; MVN Пересылка символов — Move Characters; MVC
	1111 XXXX Пересылка со сдвигом — Move with Offset; MVO Упаковка — Pack; PASC

Младшая тетрада кода операции		Операции	
		логические и пересалочные над полями переменной, длины	десятичные
0011		Пересылка зон — Move Zones; MVZ	Распаковка — Unpack; UNPK
0100		И	
0101		And; NC	
0110		Сравнение кодов — Compare Logical; CLC ИЛИ Ог; ОС	
0111		Исключающее ИЛИ Exclusive Or; XC	
1000			
1001			
1010			
1011			
1100		Перекодирование — Translate; TP	Очистка и сложение — Zero and Add; ZAP Сравнение десятичных чисел — Compare Decimal; CP Десятичное сложение — Add Decimal; AP Десятичное вычитание — Subtract Decimal; SP Десятичное умножение — Multiply Decimal; MP Десятичное деление — Divide Decimal; DP
1101		Перекодирование и проверка — Translate and Test; TRT	
1110		Редактирование — Edit; ED	
1111		Редактирование и отметка Edit and Mark; EDMK	

* Наименование операции, принятое в американской системе IBM-360.*

** Условное мнемоническое обозначение операции, применяемое при программировании на языке АССЕМБЛЕР.

Примечание. Указанные в таблице мнемонические обозначения операций приняты и в ЕС ЭВМ.

СПИСОК ЛИТЕРАТУРЫ

1. **Актуальные вопросы обеспечения надежности ЕС ЭВМ.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 112—121. Авт.: В. В. Веригин, Б. И. Ермолаев, Л. А. Кожарский, Б. П. Михайлин, И. Б. Михайлов, И. А. Попова, С. П. Соловьев, В. И. Яковлев.
2. **Булей Г.** Микропрограммирование. М., «Мир», 1973. 128 с.
3. **Бусленко Н. П.** Моделирование сложных систем. М., «Наука», 1968. 335 с.
4. **Васильев Б. В., Козлов Б. А., Ткаченко Л. Г.** Надежность и эффективность радиоэлектронных устройств. М., «Советское радио», 1964. 368 с.
5. **Вычислительная система IBM 360.** Принципы работы. М., «Советское радио», 1969. 440 с.
6. **Голубев—Новожилов Ю. С.** Многомашинные комплексы вычислительных средств. М., «Советское радио», 1967. 424 с.
7. **Данилочкин В. П., Пеледов Г. В.** Система обработки данных в реальном масштабе времени на базе ЕС ЭВМ.— «Вопросы радиоэлектроники». Серия ЭВТ, 1973, вып. 3, с. 33—42.
8. **Джермейн К.** Программирование на IBM 360. М., «Мир», 1971. 870 с.
9. **Дроздов Е. А., Комарницкий В. А., Пятибратов А. П.** Многопрограммные цифровые вычислительные машины. М., Воениздат, 1974. 406 с.
10. **Дроздов Е. А., Пятибратов А. П.** Основы построения и функционирования вычислительных систем. М., «Энергия», 1973. 368 с.
11. **Единая система ЭВМ.** Под ред. А. М. Ларионова. М., «Статистика», 1974. 135 с.
12. **Каган Б. М., Каневский М. М.** Цифровые вычислительные машины и системы. М., «Энергия», 1974. 679 с.
13. **Клячко Э. И.** Схемный и тестовый контроль автоматических цифровых вычислительных машин. М., «Советское радио», 1963. 192 с.
14. **Краймер Л. П.** Быстродействующие ферромагнитные запоминающие устройства. М.—Л., «Энергия», 1964. 371 с.
15. **Краймер Л. П.** Устройства хранения дискретной информации. Л., «Энергия», 1969. 312 с.
16. **Лапин В. С.** К вопросу о технической реализации каналов ввода-вывода ЕС ЭВМ.— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 3, с. 43—59 с ил.
17. **Логическая структура и принципы работы ЕС ЭВМ.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 3—18 с ил. Авт.: А. М. Ларионов, В. К. Левин, С. П. Соловьев, А. П. Заморин.
18. **Майоров С. А., Новиков Г. И.** Структура цифровых вычислительных машин. М., «Машиностроение», 1970. 480 с.
19. **Макурочкин В. Г.** Внешние запоминающие устройства ЕС ЭВМ.— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 3, с. 60—71 с ил.
20. **Основные свойства операционной системы ДОС ЕС ЭВМ.**— «Вопросы радиоэлектроники». Серия ЭВТ, 1973, вып. 3, с. 3—14. Авт.: Э. В. Ковалевич, М. С. Марголин, М. Г. Скоромник, Л. Т. Чупригина.
21. **Особенности структуры и функциональные характеристики ЭВМ ЕС — 1030.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 79—93 с ил.

Авт.: В. Л. Аджемян, Л. Х. Гаспарян, А. Т. Кучукян, И. Б. Мкртумян, Г. О. Патакян, М. А. Семерджян, Г. С. Хачатрян.

22. **Поспелов Д. А.** Введение в теорию вычислительных систем. М., «Советское радио», 1972. 280 с.

23. **Принципы системной организации ЭВМ Единой системы.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 55—65 с ил. Авт.: А. М. Ларионов, В. К. Левин, С. П. Соловьев, Е. Л. Брусиловский.

24. **Проектирование свойств эксплуатационной надежности для системы ИВМ—360.**— «Кибернетический сборник». Новая серия, вып. 1, «Мир», М., 1965, с. 137—161 с ил.

25. **Путинцев Н. Д.** Аппаратный контроль управляющих цифровых вычислительных машин. М., «Советское радио», 1966. 421 с.

26. **Селлерс Ф.** Методы обнаружения ошибок в работе ЭЦВМ. М., «Мир», 1972. 310 с.

27. **Система ввода-вывода ЕС ЭВМ. Принципы организации.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 30—39. Авт.: А. М. Ларионов, В. К. Левин, С. П. Соловьев, В. С. Лапин.

28. **Система математического обеспечения ЕС ЭВМ.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 19—29 с ил. Авт.: Э. В. Ковалевич, М. С. Марголин, Л. Д. Райков, Ж. П. Сватков, М. Р. Шура—Бура.

29. **Система математического обеспечения ЕС ЭВМ.** Под ред. А. М. Ларионова, М., «Статистика», 1974. 216 с.

30. **Система телеобработки данных ЕС ЭВМ.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 40—54 с ил. Авт.: В. С. Лапин, В. С. Антонов, Ю. Г. Данилевский, М. К. Сулим, В. И. Уткин.

31. **Справочник по полупроводниковым диодам, транзисторам и интегральным схемам.** Под общ. ред. Н. Н. Горюнова. Изд. 3-е, перераб. и доп., М., «Энергия», 1972. 568 с.

32. **Средства управления данными и перспективы их развития в ОС ЕС ЭВМ.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 23—32. Авт.: Е. Х. Еникеева, В. Г. Лесюк, Л. Д. Райков, Я. С. Шегидевич.

33. **Управляющая программа ОС ЕС ЭВМ.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 3, с. 15—22. Авт.: Л. Б. Лебедев, В. В. Наумов, Г. В. Пеледов, А. Г. Чекалов.

34. **Флорес А.** Программное обеспечение. М., «Мир», 1971, 519 с.

35. **Флорес А.** Организация вычислительных машин. М., «Мир», 1972, 428 с.

36. **Шаруненко Н. М.** Запоминающие устройства ЕС ЭВМ.— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 3, с. 72—95 с ил.

37. **Шелихов А. А., Селиванов Ю. П.** Вычислительные машины. Справочник. М., «Энергия», 1973. 216 с.

38. **Электронная вычислительная машина ЕС-1020.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 66—78 с ил. Авт.: В. В. Пржиялковский, Г. Д. Смирнов, Н. А. Мальцев, Р. М. Асатуров, В. П. Качков.

39. **Электронная вычислительная машина ЕС-1050.**— «Вопросы радиоэлектроники. Серия ЭВТ», 1973, вып. 1, с. 94—111 с ил. Авт.: В. С. Антонов, А. П. Заморин, В. С. Лапин, К. С. Ораевский, С. П. Соловьев, А. А. Шульгин.

40. **Электронные, управляющие (проблемы и суждения).** Газета «Правда», 11.07.1973.

ОГЛАВЛЕНИЕ

Предисловие	3
Введение	5

Глава 1

ПРИНЦИПЫ ПОСТРОЕНИЯ И ХАРАКТЕРИСТИКИ ЕС ЭВМ

1.1. Принципы построения ЭВМ Единой системы	11
1.2. Обобщенная схема ЭВМ Единой системы	13
1.3. Комплектация моделей ЭВМ Единой системы	17
1.4. Основные характеристики ЭВМ Единой системы	22
1.5. Математическое обеспечение ЕС ЭВМ Единой системы	28

Глава 2

КОДИРОВАНИЕ ДАННЫХ И КОМАНД

2.1. Системы счисления, применяемые в ЕС ЭВМ	38
2.2. Кодирование алфавитов входных языков	44
2.3. Форматы данных и расположение информации в основной оперативной памяти	50
2.4. Формы представления чисел	52
2.5. Машинные коды чисел	57
2.6. Структура набора операций процессора. Коды операций. Адресация операндов в командах	67

Глава 3

ПРИНЦИПЫ ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ ОПЕРАЦИЙ В ПРОЦЕССОРАХ МОДЕЛЕЙ

3.1. Общие сведения о структурах и особенностях реализации машинных алгоритмов операций	76
3.2. Арифметические операции над двоичными числами с фиксированной запятой	77
3.3. Арифметические операции над двоичными числами с плавающей запятой	112
3.4. Арифметические операции над десятичными числами	121
3.5. Логические операции и особенности их выполнения	128

Глава 4

ЭЛЕМЕНТНАЯ БАЗА И УНИФИКАЦИЯ КОНСТРУКЦИЙ ЭЛЕКТРОННОЙ АППАРАТУРЫ УСТРОЙСТВ ЭВМ

4.1. Типовые конструктивные модули	134
4.2. Основные комплексы интегральных схем, применяемых в моделях ЭВМ Единой системы	136
4.3. Полупроводниковые запоминающие элементы	144
4.4. Функциональные элементы	150
4.5. Типовые функциональные схемы	154

Глава 5

ОПЕРАТИВНАЯ И ПОСТОЯННАЯ ПАМЯТЬ

5.1. Принципы построения оперативной памяти типа 2,5D	160
5.2. Состав и назначение основных блоков оперативной памяти типа 2,5D для модели ЕС-1020	165
5.3. Устройство и работа составных частей оперативной памяти типа 2,5D для модели ЕС-1020	169
5.4. Функционирование оперативной памяти модели ЕС-1020	181
5.5. Оперативная память машины ЕС-1030	187
5.6. Оперативная память машины ЕС-1050	197
5.7. Постоянные запоминающие устройства	200

Глава 6

АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА ПРОЦЕССОРОВ

6.1. Особенности арифметико-логических устройств процессоров различных моделей	206
6.2. Структуры и основные информационные связи арифметико-логических устройств процессоров	209
6.3. Операционная часть арифметико-логического устройства процессора ЕС-2050	218
6.4. Операционная часть арифметико-логического устройства процессора ЕС-2030	226
6.5. Арифметико-логический блок процессора ЕС-2020	229
6.6. Особенности перемещения и преобразования операндов одинаковых операций в арифметико-логических устройствах различных моделей	236

Глава 7

ЦЕНТРАЛЬНОЕ УСТРОЙСТВО УПРАВЛЕНИЯ

7.1. Общие принципы управления работой машин Единой системы	247
7.2. Микропрограммное управление	258
7.3. Схемное управление	270
7.4. Система прерываний и приоритетов	281
7.5. Управление оперативной памятью	291
7.6. Системные средства управления	297
7.7. Пульт управления	301

Глава 8

ВЗАИМОДЕЙСТВИЕ ОСНОВНЫХ УСТРОЙСТВ ЭВМ

8.1. Выполнение пультовых процедур	312
8.2. Работа устройств при выполнении арифметических и логических операций	319
8.3. Работа устройств при выполнении вспомогательных операций	327

Глава 9

РАСПРЕДЕЛЕНИЕ И ЗАЩИТА ПАМЯТИ

9.1. Принципы распределения памяти	333
9.2. Блоки защиты	338
9.3. Память ключей защиты	343

Глава 10

ИНТЕРФЕЙС ВВОДА-ВЫВОДА

10.1. Основные положения и понятия	352
10.2. Состав и назначение линий интерфейса	356
10.3. Выполнение операций	363
10.4. Управляющая и служебная информация	372
10.5. Интерфейс УВУ—ВЗУ	379

Глава 11

ВНЕШНИЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЕС ЭВМ

11.1. Состав и характеристики ВЗУ	387
11.2. Накопители на магнитных лентах	390
11.3. Устройства управления накопителями на магнитных лентах	392
11.4. Накопители на магнитных дисках	397
11.5. Накопитель на магнитных картах	401
11.6. Устройство управления накопителями на магнитных дисках и картах	404
11.7. Накопитель на магнитном барабане. Устройство управления НМБ	407

Глава 12

УСТРОЙСТВА ВВОДА-ВЫВОДА И ПОДГОТОВКИ ДАННЫХ

12.1. Устройства ввода информации с перфокарт	411
12.2. Устройства вывода информации на перфокарты	418
12.3. Печатающие устройства	426
12.4. Пишущая машина с блоком управления	432
12.5. Выносные пульты с групповым устройством управления	438
12.6. Логический ретранслятор	441
12.7. Устройства подготовки данных	445

Глава 13

МУЛЬТИПЛЕКСНЫЕ И СЕЛЕКТОРНЫЕ КАНАЛЫ

13.1. Организация передач информации	448
13.2. Мультиплексный канал однородной структуры	454
13.3. Селекторные каналы	462
13.4. Мультиплексный канал с селекторными подканалами	474

Глава 14

ОПЕРАЦИОННЫЕ СИСТЕМЫ

14.1. Структура дисковой операционной системы и назначение отдельных ее компонентов. Основные понятия и определения	479
14.2. Управляющая программа ДОС ЕС. Программы ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА И СУПЕРВИЗОР	485
14.3. Управляющая программа ДОС ЕС. Программы УПРАВЛЕНИЕ ЗАДАНИЯМИ И ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ	493
14.4. Управляющая программа ДОС ЕС. Система управления вводом-выводом	499
14.5. Системные обслуживающие программы ДОС ЕС	501
14.6. Трансляторы. Генерация ДОС ЕС	509
14.7. Постановка задачи к решению на машине с помощью ДОС ЕС	512
14.8. Операционная система ЕС ЭВМ. Состав и функции ОС ЕС	514
14.9. Управляющая программа ОС ЕС	517

Глава 15

АППАРАТНЫЕ СРЕДСТВА ТЕЛЕОБРАБОТКИ ДАННЫХ

15.1. Общие сведения об организации телеобработки	526
15.2. Структуры звеньев передачи данных	528
15.3. Абонентские пункты	532
15.4. Мультиплексоры передачи данных	541
15.5. Аппаратура передачи данных	557
15.6. Форматы сообщений для полудуплексного режима	560

Глава 16

СРЕДСТВА АВТОМАТИЧЕСКОГО КОНТРОЛЯ РАБОТЫ И ДИАГНОСТИКИ НЕИСПРАВНОСТЕЙ ЭВМ

16.1. Общие сведения о системе обслуживания	563
16.2. Аппаратурные средства контроля	566
16.3. Программные средства технического обслуживания	573
16.4. Программно-аппаратная диагностика	578

Глава 17

ОРГАНИЗАЦИЯ ФУНКЦИОНИРОВАНИЯ ЕС ЭВМ В РАЗЛИЧНЫХ РЕЖИМАХ

17.1. Режимы работы ЭВМ Единой системы	580
17.2. Организация функционирования ЭВМ Единой системы в режиме пакетной обработки	585
17.3. Порядок обслуживания абонентов при работе ЭВМ в режиме разделения времени	593
17.4. Организация функционирования ЭВМ Единой системы в режиме разделения времени	598

Глава 18

ПОСТРОЕНИЕ МНОГОПРОЦЕССОРНЫХ И МНОГОМАШИНЫХ СИСТЕМ НА БАЗЕ ЕС ЭВМ

18.1. Общие принципы построения многопроцессорных и многомашиных вычислительных систем	601
18.2. Структуры многомашинных и многопроцессорных ВС на базе ЭВМ Единой системы	611
18.3. Многопроцессорная система с общим полем оперативной памяти	617
18.4. Структура оперативной памяти двухпроцессорной системы	623
18.5. Обмен информацией между машинами вычислительной системы	626
18.6. Показатели качества функционирования вычислительных систем	632
18.7. Задачи исследования вычислительных систем	637
18.8. Этапы проектирования ВС и их содержание	640
18.9. Пути повышения производительности и эффективности ЭВМ и ВС	645
Приложение	650
Список литературы	667

Евгений Афанасьевич ДРОЗДОВ
Владимир Александрович КОМАРНИЦКИЙ
Александр Петрович ПЯТИБРАТОВ

ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ ЕДИНОЙ СИСТЕМЫ

Редактор издательства инж. Т. В. АБИЗОВА
Технический редактор Н. Ф. ДЕМКИНА Корректор А. А. СНАСТИНА
Переплет художника Е. Г. ШУБЕНЦОВА

Сдано в набор 3/II 1976 г. Подписано к печати 23/VII 1976 г. Т-14304
Формат 60 × 90^{1/16} Бумага типографская № 2 Усл. печ. л. 42 Уч.-изд. л. 45,2
Тираж 47 000 экз. Заказ № 795 Цена 2 р. 50 к.

Издательство «Машиностроение», 107885, Москва, Б-78, 1-й Басманный пер., 3

Ленинградская типография № 6 Союзполиграфпрома
при Государственном комитете Совета Министров СССР
по делам издательств, полиграфии и книжной торговли
193144, Ленинград, С-144, ул. Монсеенко, 10