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

ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ ЕДИНОЙ СИСТЕМЫ

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

ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ ЕДИНОЙ СИСТЕМЫ

МОСКВА
«МАШИНОСТРОЕНИЕ»
1981

ББК 32.973
Д75
УДК 681.3

Авторы: *Е. А. Дроздов* (гл. 7—10, 12, 13 и п. 11.4, 11.7); *В. А. Комарницкий* (гл. 2—4, 6, 11, 15, 16, кроме п. 11.4 и 11.7, табл. 2—5 прил.); *А. П. Пятибратов* (введение, гл. 1, 5, 14, 17—19, табл. 1 прил.)

Рецензент проф. *Г. П. Молотков*
Редактор инж. *Е. Е. Скворцов*

Д72 **Дроздов Е. А., Комарницкий В. А., Пятибратов А. П.**
Электронные вычислительные машины Единой системы.— 2-е изд., перераб. и доп.— М.: Машиностроение, 1981.— 648 с., ил.

В пер.: 2 р. 60 к.

В книге изложены основы выполнения операций, принципы построения и организации функционирования ЭВМ Единой системы. Рассмотрены общие схемы машин, их элементная база; особенности построения и основные схемы блоков и устройств машин, а также унифицированного набора внешних устройств; структура и возможности математического обеспечения, средства автоматического контроля и вопросы организации централизованного технического обслуживания, способы и аппаратура организации совместной работы нескольких машин и работы в составе систем телеобработки данных.

Второе издание в основном базируется на рассмотрении моделей ЭВМ ЕС-1022, ЕС-1033 и ЕС-1050 (ЕС-1052). Книга предназначена для тех, кто готовится к участию в наладке, эксплуатации машин Единой системы, а также будет полезна специалистам, уже работающим на таких машинах.

Д $\frac{30502\text{-}033}{038(01)\text{-}81}$ 33-81. 2405000000 ББК 32.973
6Ф7.3

ИБ № 2575

Евгений Афанасьевич Дроздов,
Владимир Александрович Комарницкий,
Александр Петрович Пятибратов

Редактор **З. С. Баранова**
Художественный редактор **С. С. Водчиц**
Технические редакторы **Ф. П. Мельниченко** и **Е. П. Смирнова**
Корректоры **О. Е. Мишина** и **А. П. Озерова**
Оформление художника **С. И. Сергеева**

Сдано в набор 05.02.81. Подписано в печать 30.09.81. Т-23660. Формат 60 × 90$^1/_{16}$. Бумага типографская № 1. Гарнитура Таймс. Печать высокая. Усл. печ. л. 40,5. Уч.-изд. л. 44,15. Тираж 70000 экз. Заказ № 1655. Цена 2 р. 60 к.

Издательство «Машиностроение», 107076, Москва, Стромынский пер., 4. Ордена Октябрьской Революции, ордена Трудового Красного Знамени Ленинградское производственно-техническое объединение «Печатный Двор» имени А. М. Горького Союзполиграфпрома при Государственном комитете СССР по делам издательств, полиграфии и книжной торговли. 197136, Ленинград, П-136, Чкаловский пр., 15.

Оглавление

Предисловие. 7

Введение. 8

Глава 1

Принципы построения и характеристика ЕС ЭВМ. 13

1.1. Принципы построения ЭВМ Единой системы. 13
1.2. Логическая структура технических средств. 15
1.3. Основные характеристики ЭВМ Единой системы. 22
1.4. Система программного обеспечения ЕС ЭВМ. 27

Глава 2

Кодирование данных и команд. 41

2.1. Системы счисления, применяемые в ЕС ЭВМ. 41
2.2. Кодирование алфавитов входных языков. 47
2.3. Форматы данных и расположение информации в основной оперативной памяти. 52
2.4. Формы представления чисел. 54
2.5. Машинные коды чисел. 59
2.6. Структура набора операций процессора. Коды операций. Адресация операндов в командах. 68

Глава 3

Принципы выполнения арифметических и логических операций в процессорах моделей. 76

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. 103
3.4. Арифметические операции над десятичными числами. 110
3.5. Логические операции и особенности их выполнения. 118

Глава 4

Элементная база ЭВМ. 123

4.1. Типовые конструктивные модули. 123
4.2. Интегральные схемы. 124
4.3. Полупроводниковые запоминающие элементы. 135
4.4. Функциональные элементы. 143
4.5. Типовые функциональные схемы. 148
4.6. Интегральные схемы повышенной степени интеграции. 153
4.7. Универсальные арифметико-логические интегральные схемы. 158

Глава 5

Оперативная и постоянная память. 172

5.1. Принципы построения оперативной памяти типа 2,5D. 172
5.2. Состав и назначение основных блоков оперативной памяти ЭВМ ЕС-1022. 176

5.3. Оперативная память машины ЕС-1033. 182
5.4. Оперативная память старших моделей ЕС ЭВМ. 190
5.5. Постоянные запоминающие устройства. 193

Глава 6

Арифметико-логические устройства процессоров. 198

6.1. Особенности арифметико-логических устройств процессоров различных моделей. 198
6.2. Структуры и основные информационные связи арифметико-логических устройств процессоров. 201
6.3. Операционная часть арифметико-логического устройства процессора ЕС-2050. 211
6.4. Операционная часть арифметико-логического устройства процессора ЕС-2433. 220
6.5. Арифметико-логический блок процессора ЕС-2622. 222

Глава 7

Центральное устройство управления. 231

7.1. Общие принципы управления работой машин Единой системы. 231
7.2. Микропрограммное управление. 241
7.3. Схемное управление. 255
7.4. Система прерываний и приоритетов. 265
7.5. Управление оперативной памятью. 275
7.6. Системные средства управления. 280
7.7. Пульт управления. 284

Глава 8

Взаимодействие основных устройств ЭВМ. 295

8.1. Выполнение пультовых процедур. 295
8.2. Работа устройств при выполнении арифметических и логических операций. 302
8.3. Работа устройств при выполнении вспомогательных операций. 309
8.4. Взаимодействие устройств при микропрограммном управлении и магистральной структуре процессора. 313

Глава 9

Распределение и защита памяти. 326
9.1. Принципы распределения памяти. 326
9.2. Блоки защиты. 330
9.3. Память ключей защиты. 335

Глава 10

Интерфейс ввода-вывода. 344

10.1. Основные положения и понятия. 344
10.2. Состав и назначение линий интерфейса. 347
10.3. Выполнение операций. 354
10.4. Управляющая и служебная информация. 363

Глава 11

Внешние запоминающие устройства ЕС ЭВМ. 370

11.1. Состав и характеристики ВЗУ. 370
11.2. Накопители на магнитных лентах. 372
11.3. Устройства управления накопителями на магнитных лентах. 382
11.4. Интерфейс УВУ-НМЛ. 387
11.5. Накопители на магнитных дисках. 391
11.6. Устройства управления накопителями на магнитных дисках. 398
11.7. Интерфейс УВУ-НМД. 401

Глава 12

Устройства ввода-вывода и подготовки данных. 405

12.1. Устройства ввода информации с перфокарт. 405
12.2. Устройства вывода информации на перфокарты. 412
12.3. Печатающие устройства. 419
12.4. Пишущая машина с блоком управления. 424
12.5. Комплекс устройств отображения информации. 429
12.6. Логический ретранслятор. 433
12.7. Устройства подготовки данных. 436

Глава 13

Мультиплексные и селекторные каналы. 440

13.1. Организация передач информации. 440
13.2. Мультиплексный канал однородной структуры. 445
13.3. Селекторные каналы. 453
13.4. Мультиплексный канал с селекторными подканалами. 464
13.5. Каналы магистральной структуры. 468

Глава 14

Операционные системы. 476

14.1. Основные понятия и определения. 476
14.2. Дисковая операционная система ДОС ЕС. 478
14.3. Операционная система ОС ЕС. Состав и функции ОС ЕС. 492
14.4. Управляющая программа ОС ЕС. Управление данными. 495
14.5. Управляющая программа ОС ЕС. Управление заданиями. 499
14.6. Управляющая программа ОС ЕС. Управление задачами. 502
14.7. Трансляторы. Версии ОС ЕС. 507

Глава 15

Аппаратные средства телеобработки данных. 511

15.1. Общие сведения об организации телеобработки. 511
15.2. Структуры звеньев передачи данных. 513
15.3. Абонентские пункты. 517
15.4. Мультиплексоры передачи данных. 526
15.5. Аппаратура передачи данных. 544
15.6. Форматы сообщений для полудуплексного режима. 547

Глава 16

Средства автоматического контроля работы и диагностики неисправностей ЭВМ. 550

16.1. Общие сведения о системе обслуживания. 550
16.2. Аппаратные средства контроля. 552
16.3. Программные средства технического обслуживания. 560
16.4. Программно-аппаратная диагностика. 565

Глава 17

Организация функционирования ЕС ЭВМ в различных режимах. 568

17.1. Режимы работы ЭВМ Единой системы. 568
17.2. Дисциплины обслуживания требований. 574
17.3. Организация функционирования ЭВМ в режиме пакетной обработки. 585
17.4. Организация функционирования ЭВМ Единой системы в режиме разделения времени. 592

Глава 18

Построение многопроцессорных и многомашинных вычислительных систем на базе ЕС ЭВМ. 595

18.1. Общие принципы построения многопроцессорных и многомашинных вычислительных систем. 595
18.2. Структуры многомашинных и многопроцессорных ВС. 604
18.3. Многопроцессорная система с общим полем оперативной памяти. 608
18.4. Операционные системы многопроцессорных ВС. 614
18.5. Показатели качества и эффективности функционирования вычислительных систем. 619

Глава 19

Комплексное централизованное обслуживание средств ЕС ЭВМ. 624

19.1. Функции и принципы построения системы комплексного централизованного обслуживания средств ЕС ЭВМ. 624
19.2. Структура системы комплексного централизованного обслуживания средств ЕС ЭВМ. 627
19.3. Организация ввода в эксплуатацию и технического обслуживания ЕС ЭВМ. 630
19.4. Фонд программ и служба сопровождения. 633

Приложение. 636

Список литературы. 647

Предисловие

Первое издание книги, выпущенное в 1976 г., получило достаточно широкое распространение. Однако некоторые его положения устарели, так как на смену моделям ЕС-1020, ЕС-1030 и ЕС-1050 пришли более совершенные модели ЕС-1022, ЕС-1033 и ЕС-1052, практически завершающие гамму машин Единой системы (ЕС) первого стандартного ряда. Это обстоятельство было учтено авторами при подготовке второго издания книги. В то же время основным при построении материала книги, как и прежде, остается то, что все модели ЭВМ Единой системы имеют общую структурную организацию, общую конструктивную и элементную базу, являются программно-совместимыми.

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 в основном базируется на схемах и структурах машин ЕС-1022, ЕС-1033, ЕС-1050 (ЕС-1052); рассматриваются также вопросы, относящиеся ко всем моделям ЭВМ Единой системы, в том числе вопросы организации ввода и вывода информации, телеобработки, контроля, построения и функционирования вычислительных систем.

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 блоки и устройства. Следует иметь в виду, что эта книга не является справочником по моделям ЭВМ Единой системы, ее назначение — ввести читателя в круг идей и решений, принятых при создании ЕС ЭВМ, чтобы облегчить углубленное изучение некоторых конкретных вопросов или моделей ЭВМ.

Общая последовательность изложения материала по основным принципам построения и функционирования ЭВМ Единой системы, принятая в книге, отвечает сложившейся к настоящему времени методике изучения основ построения и применения технических средств вычислительных систем. Это, по мнению авторов, должно способствовать лучшему усвоению рассматриваемых идей и принципов, расширению и углублению полученных ранее знаний. При этом предполагается, что читатель уже знаком с общими принципами построения и применения цифровых вычислительных машин и систем, основами организации и использования средств их общего программного обеспечения.

Авторы выражают искреннюю признательность руководителям и исполнителям проектных работ по ЕС ЭВМ за полезные консультации и советы, рецензенту профессору Г. П. Молоткову и редактору Е. Е. Скворцову, в значительной степени способствовавшим отработке материала книги.

ВВЕДЕНИЕ

Научно-технический прогресс, развитие современной научной мысли, развитие практически всех отраслей народного хозяйства тесно связаны 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 программному 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, программным 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 являлась электронная лампа, остальные компоненты электронной аппаратуры — это обычные резисторы, конденсаторы, трансформаторы. 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 (телетайпы, 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 программное обеспечение. Несмотря на это, первое поколение ЭВМ заложило основы логического построения машин и продемонстрировало грандиозные возможности цифровой вычислительной техники.

На смену лампам 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 в аппаратурном комплексе 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 ступенью 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 совместная 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 использовать 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.

Программное 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 разделения 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 (десятки 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 программному обеспечению. Например, в режиме разделения 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 машинное время предоставлено одному ему.

При разработке машин третьего поколения применяют 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, изготовляемых 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 программное обеспечение. Более тесной становится связь структуры машины и ее программного обеспечения, особенно операционной системы.

В настоящее время в серийном производстве находятся ЭВМ третьего поколения, которые по сравнению с машинами первых поколений имеют улучшенные характеристики (по быстродействию на два-три порядка, по емкости оперативной памяти на два порядка, по надежности на три порядка). Одновременно на два-три 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 программного 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 использованы все современные достижения в области электронной вычислительной техники, технологии и конструирования ЭВМ, в области построения систем программного 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, предназначенные для решения широкого круга научно-технических, экономических и других задач. Важной областью применения ЕС ЭВМ является использование их в АСУ.

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 тем, 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 й 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 пульты со средствами наглядного отображения 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 машины.

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. Технико-экономические показатели зависят от таких факторов, как структура машины, ее технологичность, состав программного обеспечения. Улучшения 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 программного обеспечения ЕС ЭВМ. Проводится стандартизация инструкций, кодов, идет разработка подсистем программного обеспечения, ориентированных на широкие классы применений ЭВМ, — это и АСУ предприятий и объединений, отраслей министерств и сферы торговли, снабжения и другие области. В этой работе принимают деятельное участие все страны — разработчики ЕС ЭВМ. Активное участие социалистических стран в разработке и совершенствовании машин Единой системы и их программного обеспечения во многом способствует более полному решению задачи освоения и серийного выпуска современных средств электронной вычислительной техники.

Глава 1

ПРИНЦИПЫ ПОСТРОЕНИЯ И ХАРАКТЕРИСТИКА ЕС ЭВМ

1.1. Принципы построения ЭВМ Единой системы

Единая система ЭВМ представляет собой семейство совместимых между собой универсальных ЭВМ третьего поколения с достаточно широким диапазоном по производительности и единой логической структурой. Различные модели ЕС ЭВМ отличаются по производительности, максимальной емкости оперативной памяти и максимальному количеству подключаемых внешних устройств.

Все машины Единой системы относятся к категории многопрограммных (мультипрограммных), являющихся дальнейшим развитием однопрограммных машин, в которых каждая последующая программа может выполняться только после реализации предыдущей.

Основные отличия многопрограммных ЭВМ от однопрограммных заключаются в следующем:

1. Многопрограммные машины имеют более сложную логическую структуру, в них предусматриваются средства распределения памяти между выполняемыми программами, система защиты памяти, система прерывания и приоритетов, датчик времени. Эти средства и системы дают возможность реализовать принцип мультипрограммирования, суть которого заключается в том, что на одной и той же ЭВМ могут совместно выполняться несколько программ. При работе в многопрограммном (мультипрограммном) режиме в машине совместно выполняются разные команды одной и той же или различных независимых программ, которые хранятся в оперативной памяти. Причем в каждый данный момент процессор выполняет только одну какую-то команду. Цель применения мультипрограммного режима работы машины — повысить эффективность ее использования за счет организации параллельной работы ее основных устройств. Машины, в которых реализован такой режим работы, называются мультипрограммными, или многопрограммными.

2. Большинство многопрограммных ЭВМ строится таким образом, чтобы можно было организовать различные режимы работы, отличающиеся порядком обслуживания программ абонентов (пользователей) и степенью их воздействия на вычислительный процесс: режим пакетной обработки, режим работы с разделением времени и др..

3. Для многопрограммных машин характерно широкое применение модульных структур; это относится прежде всего к запоминающим

устройствам и устройствам ввода-вывода, состав которых определяется нуждами потребителя.

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 в различных кодах.

В основу построения ЕС ЭВМ положены следующие принципы.

1. Принцип мультипрограммирования, о котором говорилось выше.

2. Информационная и программная совместимость. Информационная совместимость выражена: в совместимости алфавитов и кодов, используемых для обмена информацией между устройствами машины; в структуре и формате сообщений в линиях связи, а также процедурах управления передачей данных; в структуре и разметке файлов на различных носителях (магнитных дисках, магнитных лентах, перфокартах, перфолентах). Программная совместимость — определяющее свойство ЕС ЭВМ — осуществлена на уровне машинных кодов, языке Ассемблера, языков высокого уровня. Поскольку все машины Единой системы имеют идентичные системы команд, системы прерываний, системы управления внешними устройствами, программа решения той или иной задачи составляется независимо от того, на какой именно конкретной модели она будет выполняться. Обработка информации по такой программе на разных моделях ЕС ЭВМ будет отличаться лишь временем ее выполнения. Совместимость ЕС ЭВМ позволяет: использовать имеющиеся заделы программ при переходе к работе на других машинах одного и того же ряда, существенно экономить в силах и средствах при разработке системы стандартных программ математического обеспечения (МО) за счет унификации этих программ для ЭВМ различной производительности, создать необходимые предпосылки для технической стандартизации машин при их дальнейшем совершенствовании.

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 процессором, в стандартизации конструктивного решения ряда узлов и блоков. Такой уровень стандартизации позволяет, с одной стороны, сэкономить в силах и средствах при разработке и серийном производстве ЭВМ 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 комплектации машин по количеству модулей оперативной памяти, ВЗУ, УВВ. Стандартизация связей внешних устройств с процессорами позволяет создавать многомашинные вычислительные системы различной конфигурации.

4. Широкое применение интегральной микроэлектроники. Все машины Единой системы строятся на основе интегральных микроэлектронных схем, что позволяет удешевить аппаратуру, повысить ее надежность, улучшить габаритно-массовые и эксплуатационные характеристики, в частности снизить потребляемую мощность, а также затраты на системы охлаждения, электросилового питания и оборудование помещений.

5. Возможность организации многоэтапной работы по созданию и совершенствованию машин Единой системы и их математического обеспечения.

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

1.2. Логическая структура технических средств

Технические средства ЕС ЭВМ представляют собой ряд моделей, включающих две группы устройств — центральные и периферийные (рис. 1.1).

Ряд моделей ЕС ЭВМ объединяет машины, имеющие общую логическую структуру и принципы работы. Под логической структурой ЭВМ понимают совокупность функциональных средств и принципов их взаимодействия. Однородность логической структуры всех моделей ЕС ЭВМ достигается тем, что предусмотренные в ней средства выполняют одинаковые функции и одинаково взаимодействуют между собой. Несмотря на общность логической структуры, различные модели ряда отличаются производительностью, емкостью памяти, конфигурацией и т. д. Это объясняется тем, что структура как логическое понятие определяет лишь общую концепцию построения и взаимодействия аппаратных средств и не накладывает жестких ограничений на конкретную техническую реализацию модели. Например, в одних моделях некоторые технические средства могут представлять отдельные конструкции, а в иных они могут быть интегрированы в оборудовании других средств.

Поскольку ЕС ЭВМ представляет собой непрерывно развивающуюся систему, в которой улучшаются качественные характеристики составных частей, растут технико-экономические показатели, совершенствуется номенклатура периферийных устройств, в данный ряд ЭВМ могут входить те или иные модели. Так, модели ЕС-1010, ЕС-1020, ЕС-1020А, ЕС-1030, ЕС-1040, ЕС-1050 и ЕС-1060 объединены в семейство моделей Ряд-1. К числу модифицированных образцов ЕС ЭВМ Ряд-1 принадлежат модели ЕС-1012, ЕС-1022, ЕС-1032, ЕС-1033, ЕС-1052. Более совершенные образцы моделей (ЕС-1015, ЕС-1025, ЕС-1035, ЕС-1045, ЕС-1055) составляют семейство моделей Ряд-2.

Устройства ЭВМ являются обычно функционально независимыми по преобразованию информации частями машины. В конструктивном отношении не все они представлены в виде автономной, самостоятельной части.

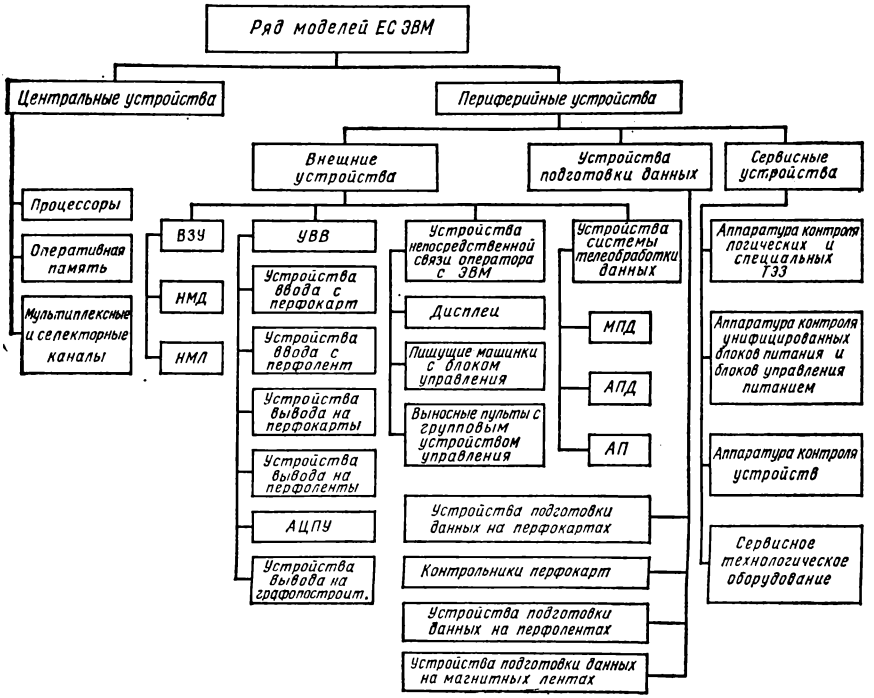


Рис. 1. 1. Структура технических средств ЕС ЭВМ

Центральные устройства ЭВМ. Эта группа устройств включает технические средства, определяющие основные технические характеристики машин. Это — процессоры, оперативная память, мультиплексные и селекторные каналы.

Процессор — главная часть ЭВМ, осуществляющая заданные программой преобразования информации, управление всем вычислительным процессом и взаимодействие устройств. В процессоре сосредоточены средства выполнения арифметических, логических и вспомогательных операций, управления выполнением заданной последовательности команд, обращения к оперативной памяти и инициирования обменов с внешними устройствами. Координация работы устройств, блоков и узлов машины производится с помощью синхронизирующих и управляющих сигналов, вырабатываемых в процессоре. Синхронизирующие сигналы обеспечивают совмещение срабатывания функциональных частей ЭВМ во времени,

а управляющие сигналы задают необходимую последовательность их срабатывания. В разных моделях ЕС ЭВМ схемы процессоров различны, чем и объясняется их неодинаковая производительность. Однако логические функции процессоров одинаковы, и результаты выполнения команд программы идентичны. Особенности конкретной структуры процессора каждой модели ЕС ЭВМ оговорены в технических характеристиках.

Основными функциональными частями процессора являются: устройство управления процессором, арифметико-логическое устройство (АЛУ), регистровая память и пульт управления. Устройство управления осуществляет все функции по управлению вычислительным процессом в соответствии с программой. В арифметико-логическом устройстве производится обработка двоичных и шестнадцатиричных полей фиксированной длины с фиксированной и плавающей запятой, десятичных полей (двоично-десятичных слов) переменной длины и логических полей как фиксированной, так и переменной длины. Регистровая память процессора представляет собой сверхоперативную память ЭВМ. Она состоит из регистров общего назначения (РОН), которые используются в качестве индексных регистров и регистров местной памяти при выполнении логических операций и арифметических операций с числами, представленными в естественной форме (с фиксированной запятой), и регистров с плавающей запятой (РПЗ), используемых при выполнении операций с числами, представленными в нормальной форме (с плавающей запятой). Пульт управления процессора предназначен для организации взаимодействия оператора или программиста с ЭВМ. Он содержит необходимые для этого средства и органы управления.

Оперативная память ЕС ЭВМ служит для приема, хранения и выдачи информации (операндов и команд), непосредственно используемой в процессе выполнения операций в АЛУ и устройстве управления процессора. В ней хранятся управляющие и рабочие программы, данные, необходимые для решения задач, промежуточные и окончательные результаты обработки информации. Оперативная память (ОП) ЕС ЭВМ делится на основную (ООП) и большую (БОП). Большая оперативная память является резервом основной. Она имеется только в старших моделях ЕС ЭВМ.

Для различных моделей ЕС ЭВМ ООП может иметь различные параметры: емкость, быстродействие, разрядность, количество логических блоков (модулей). Обращение к данному логическому блоку может осуществляться параллельно и независимо от других логических блоков.

Быстродействие ООП 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 или запись новой информации в ячейку. Время выборки — это время, затрачиваемое на дешифрацию адреса и выборку содержимого ячейки.

Оперативная память организована таким образом, что возможна адресация хранящейся в ней информации к каждому байту. За одно обращение к памяти в разных моделях выбирается одновременно различное количество байтов (два, четыре или восемь), представляющее

собой ширину выборки. Для адресации используется 24-разрядный двоичный адрес, что позволяет адресовать 16 777 216 байт.

В ООП высокопроизводительных моделей ЕС ЭВМ применяется так называемый принцип расслоения памяти. Это метод адресации, при котором смежные ячейки памяти располагаются в следующих по номеру модулях. Такое чередование адресов в параллельно работающих модулях ОПП позволяет осуществлять одновременное обращение к ним и, следовательно, сокращать эффективный цикл обращения.

Оперативная память ЕС ЭВМ снабжена средствами защиты, которые защищают содержимое ее ячеек от разрушения из-за ошибочного обращения к ним при работе в мультипрограммном режиме.

Подключение внешних устройств ЭВМ к центральным устройствам осуществляется с помощью специальных устройств обмена — каналов ввода-вывода. Весь объем работы по обмену информацией между внешними устройствами и ООП выполняет канал ввода-вывода; процессор лишь инициирует ввод-вывод одной командой. В функции процессора входит пуск и остановка канала, а также выявление непредвиденных в нем ситуаций. Программу ввода-вывода канал выполняет независимо и одновременно с работой процессора.

Имеются два типа каналов ввода-вывода — мультиплексный и селекторный. Они различаются по назначению, внутренней структуре и режимам работы.

Мультиплексный канал (МК) представляет собой логически самостоятельное функциональное устройство, осуществляющее организацию взаимообмена информацией между ООП машины и группой сравнительно медленнодействующих внешних устройств (устройств ввода-вывода с перфокарт, перфолент, аналого-цифровых печатающих устройств, графопостроителей, устройств телеобработки и т. д.). Он может работать как в монопольном, так и в мультиплексном режиме. В первом случае средства канала монополизируются в течение некоторого времени одним внешним устройством. Во втором случае МК работает по методу уплотнения передач путем разделения во времени действий по обслуживанию нескольких одновременно подключенных внешних устройств.

Мультиплексный канал разделяется на подканалы, число которых определяется числом внешних устройств, подключенных к нему. Подканал образуют средства, выделенные для работы с отдельным внешним устройством. В сущности это главным образом память канала, используемая для хранения различной информации, относящейся к данной операции ввода-вывода.

Подканалы в МК могут быть неразделенными или разделенными (селекторными). Неразделенный подканал связан и может использоваться только одним внешним устройством. Такие подканалы работают с внешними устройствами, не имеющими ограничений на одновременность передачи или приема данных. Разделенный подканал связан и может использоваться группой однотипных внешних устройств. Разделенные подканалы применяются для работы с внешними устройствами, имеющими общее устройство управления.

Селекторный канал (СК) — это логически самостоятельное функциональное устройство, осуществляющее организацию взаимообмена информа-

цией между ООП машины и группой сравнительно быстродействующих внешних устройств (внешних запоминающих устройств на магнитных дисках, лентах, барабанах). Селекторный канал имеет только один подканал и может работать только в монопольном режиме.

Подключение внешних устройств к каналам ввода-вывода осуществляется с помощью стандартной системы сопряжения — интерфейса ввода-вывода, который представляет собой унифицированную систему линий связи с определенными функциями и параметрами сигналов на этих линиях. Интерфейс ввода-вывода обеспечивает для всех внешних устройств одинаковые формат данных и последовательность сигналов.

Периферийные устройства ЕС ЭВМ. В состав этой группы устройств входят внешние устройства, устройства подготовки данных, сервисные устройства.

Внешние устройства — аппаратура для хранения больших массивов информации, для обеспечения ввода данных, необходимых при решении задач, а также для вывода результатов обработки информации, устройства непосредственной связи оператора с ЭВМ, устройства системы телеобработки данных.

В рамках ЕС ЭВМ для хранения больших массивов информации разработаны и используются накопители на магнитных лентах (НМЛ), накопители на магнитных дисках (НМД) и устройства управления этими накопителями, обеспечивающие подключение их к каналам ввода-вывода. Серийно выпускаемые НМЛ и НМД имеют широкий диапазон скоростей обмена информацией и различную плотность записи.

Устройства ввода служат для восприятия вводимой информации (исходных данных и программ), преобразования ее в электрические кодовые сигналы и передачи через интерфейс ввода-вывода к мультиплексному каналу. Для этого применяются устройства ввода с перфокарт и перфолент.

Устройства вывода преобразуют выводимый из машины числовой материал, представленный электрическими сигналами, и либо наносят его в виде системы пробивок на перфокарты и перфоленты, либо печатают результаты решения задач на бумажных лентах, или выдают в виде графиков, планов, карт. К ним относятся устройства вывода на перфокарты и перфоленты, аналого-цифровые печатающие устройства (АЦПУ), устройства вывода на графопостроители (графические регистрирующие устройства). Графические регистрирующие устройства (ГРУ) используют во многих отраслях, где требуется быстрое и точное выполнение чертежей, графиков, планов и т. д. Применяются два типа ГРУ — планшетные и рулонные. В каждом из них основными функциональными блоками являются электромеханический построитель графиков, блок управления и блок преобразования данных. С помощью этих блоков осуществляется подключение ГРУ к мультиплексному или селекторному каналу, передача информации из ООП и необходимое ее преобразование. Поступающая из оперативной памяти информация в конечном счете преобразуется в линейное перемещение пишущего узла в прямоугольной системе координат (в ГРУ планшетного типа) или в перемещение рулона и пишущего узла вдоль образующей барабана (в ГРУ рулонного типа).

К группе устройств, обеспечивающих непосредственную связь оператора с ЭВМ, относят пишущие машинки с блоком управления, дисплеи, выносные пульты с групповым устройством управления.

Пишущая машинка и блок управления с пультом образуют устройство, которое предназначено для организации непосредственного обмена информацией между оператором и ЭВМ через стандартный интерфейс ввода-вывода. Устройство подключается к мультиплексному или селекторному каналу моделей ЕС ЭВМ и может работать как в мультиплексном, так и в монопольном режимах. Основные функции устройства: ввод данных для управления процессом выполнения задач в процессоре (при вводе данных с помощью клавиатуры они одновременно печатаются на бумаге), вывод небольших массивов данных, вывод содержимого выбранных ячеек ООП, протоколирование процесса выполнения программы. Технические характеристики используемых пишущих машинок несколько различаются, но эти различия не являются принципиально существенными. Поэтому любую пишущую машинку ЕС ЭВМ можно применять в составе любой модели Единой системы.

Дисплеи представляют собой устройства ввода-вывода алфавитно-цифровой и графической информации на электронно-лучевую трубку (ЭЛТ). В их состав входят следующие функциональные блоки: управления, оперативной памяти, сопряжения с МК или СК, генератор знака и векторов, индикатор с ЭЛТ, алфавитно-цифровая и функциональная клавиатура, блок питания, световой карандаш с управлением, операторский пульт.

Работа дисплея осуществляется под управлением программы, хранящейся в оперативной памяти. С помощью светового карандаша можно перемещать и поворачивать изображение на экране ЭЛТ, изменять масштаб, стирать, вводить с клавиатуры текст в любое место экрана. Дисплей весьма удобен для оперативного редактирования наборов данных и внесения изменений непосредственно в программу в ходе решения задачи. С помощью функциональной клавиатуры дисплея можно запрограммировать выполнение специальных программ, связанных с решаемой задачей.

Дисплей можно использовать как в качестве локального терминала, когда операции ввода-вывода осуществляются посредством прямой связи между блоками управления дисплея и мультиплексным (или селекторным) каналом, так и в качестве удаленного терминала, когда дисплей передает и воспринимает информацию через линии связи и взаимодействует с каналами через мультиплексоры передачи данных.

Выносные пульты с групповым устройством управления применяются для обслуживания абонентов, удаленных на несколько сот метров от ЭВМ. Выносные пульты, каждый из которых конструктивно автономен и рассчитан на индивидуальное использование, подключаются к мультиплексному или селекторному каналу через стандартный интерфейс ввода-вывода. Работа по обмену алфавитно-цифровой информацией между ООП машины и выносными пультами осуществляется как в мультиплексном, так и в монопольном режимах.

Для машин третьего поколения, и в частности для ЕС ЭВМ, характерной чертой является возможность телеобработки данных. В составе ЕС ЭВМ для этого имеются аппаратные и программные средства.

Эти средства образуют систему телеобработки, представляющую собой комплекс взаимосвязанных и согласованно действующих технических и программных средств, обеспечивающих дистанционную обработку данных, поступающих в ЭВМ от удаленных абонентов.

В состав технических средств системы телеобработки входят ЭВМ (одна или несколько), являющаяся центральным органом управления и обработки информации, аппаратура передачи данных (АПД), устройства сопряжения ЭВМ с АПД, линии связи, аппаратура абонентских пунктов (АП).

В системах телеобработки, построенных на базе ЕС ЭВМ, функции устройства сопряжения ЭВМ с АПД выполняет мультиплексор передачи данных (МПД), который в сущности представляет собой многоканальный коммутатор. По отношению к ЭВМ он реализует функции устройства управления удаленными АП. Как и все устройства управления внешними устройствами (УВУ), МПД подключается к МК посредством интерфейса ввода-вывода.

Абонентские пункты могут комплектоваться аппаратурой ввода-вывода специализированного или универсального типа. Средства телеобработки на основе ЕС ЭВМ комплектуются универсальным терминальным оборудованием. Аппаратура АП включает телетайпы, АЦПУ, пишущие машинки, перфокарточные и перфоленточные устройства ввода-вывода, НМЛ, дисплеи.

Аппаратура передачи данных обеспечивает установление соединений между АП и МПД по некоммутируемым линиям связи, передачу по ним дискретной информации и повышение ее достоверности, если в этом есть необходимость. В состав АПД входят устройства преобразования сигналов телеграфного типа, устройства преобразования сигналов для телефонных каналов (модемы), вызывные устройства, устройства защиты от ошибок.

Устройства подготовки данных ЕС ЭВМ разделяются на три группы: перфокарточные, перфоленточные и использующие магнитные ленты.

Применяются несколько типов устройств подготовки данных на перфокартах (карточных перфораторов), отличающихся скоростью перфорирования, количеством печатных символов, потребляемой мощностью и т. д. Контроль правильности нанесения информации на перфокартах осуществляется путем дублирования перфокарт и сравнения их на контрольниках.

В устройствах подготовки данных на перфолентах предусматривается ряд режимов работы, в частности: режим подготовки данных — для нанесения информации на перфоленту с клавиатуры и печати ее на бланке, режим распечатки — для распечатки данных с перфоленты на бланк, режим реперфорации — для репродукции перфоленты, режим сравнения — для сравнения двух перфолент между собой и др.

Устройство подготовки данных на магнитной ленте работает в двух режимах: запись на магнитную ленту данных, вводимых оператором с помощью алфавитно-цифровой аппаратуры, и печать данных, считываемых с магнитной ленты. Программа работы устройства вводится с клавиатуры или с магнитной ленты.

Сервисные устройства ЕС ЭВМ предназначены для контроля, наладки, испытания и осуществления ремонта технических средств и их частей на месте эксплуатации. Совместно с ЗИП устройств сервисная аппаратура способствует обеспечению требуемого времени восстановления ЭВМ и повышению коэффициента их использования.

В состав сервисного оборудования входят: аппаратура контроля логических и специальных типовых элементов замены (ТЭЗ); аппаратура контроля унифицированных блоков питания и управления питанием; аппаратура контроля устройств ЭВМ; сервисное технологическое оборудование.

Контроль логических и специальных ТЭЗ осуществляется с помощью стендов и тестеров. Например, для контроля логических ТЭЗ серии 155, выполненных на интегральных схемах и имеющих выходной разъем с 48 контактами, применяется стенд ЕС-А102. Необходимые для проверки ТЭЗ сигналы вырабатываются генератором стенда и поступают на контрольные гнезда и коммутационное поле. Контроль производится в соответствии с таблицами проверки, разработанными для каждого типа ТЭЗ. Соответствие контролируемых параметров табличным значениям устанавливается визуально с помощью стандартных контрольно-измерительных приборов, подключаемых к стенду вручную.

Для контроля унифицированных блоков питания и блоков управления питанием используют стационарный стенд ЕС-А001. Стенд позволяет подключить к проверяемому блоку питания соответствующие питающие напряжения, нагрузку и контрольно-измерительные приборы. Стенд ЕС-А001 совместно с приставкой ЕС-А003 обеспечивает контроль нестабилизированных блоков питания и блоков управления питанием.

Переносной прибор ЕС-А703 применяют для проверки устройств ввода-вывода. Он обеспечивает индикацию потенциальных сигналов, поступающих из УВВ при их автономной или совместной с интерфейсом ввода-вывода работе; кроме того, прибор ЕС-А703 может передавать в УВВ сигналы, обеспечивающие различные режимы их работы.

Контроль накопителей на сменных магнитных дисках осуществляется с помощью устройства ЕС-А503, которое позволяет производить автономную проверку, профилактические работы, фиксацию неисправностей и наладку после устранения отказов или замены узлов накопителей.

Имеется ряд других приборов, которые применяются для контроля устройств ЭВМ. Сервисное технологическое оборудование предназначено для выполнения ремонта неисправных электронных узлов машины в процессе ее эксплуатации. Оно включает ремонтно-технологическое оснащение и инструменты: электропаяльники, фрезы для снятия лака, приспособления для удаления изоляции, съема микросхем, демонтажа накрутки, подготовки провода к монтажу и т. д.

1.3. Основные характеристики ЭВМ Единой системы

Характеристики ряда моделей машин Единой системы приведены в приложении (табл. 1). Кроме данных, содержащихся в этой таблице,

приведем некоторые дополнительные сведения о трех следующих моделях.

Модель ЕС-1022 предназначена для решения широкого круга научно-технических, информационно-логических и экономических задач малого и среднего объема в различных сферах применения. Возможности многостороннего использования обеспечиваются развитой системой команд, большим объемом основной оперативной памяти и внешних запоминающих устройств, а также наличием стандартного интерфейса для присоединения необходимого количества внешних устройств.

В состав ЭВМ ЕС-1022 входят следующие устройства:

процессор ЕС-2622 с оперативной памятью, мультиплексным каналом и двумя селекторными каналами;
пишущая машинка с блоком управления ЕС-7077;
накопитель на сменных магнитных дисках ЕС-5056;
устройство управления для НМД ЕС-5551;
накопитель на магнитной ленте ЕС-5017 – 4 шт.;
устройство управления для НМЛ ЕС-5517;
устройство ввода с перфокарт ЕС-6012;
устройство ввода с перфоленты ЕС-6022;
устройство вывода на перфоленту ЕС-7022;
устройство вывода на перфокарты ЕС-7010;
печатающее устройство ЕС-7032;
устройство подготовки данных на перфокартах ЕС-9011 – 2 шт.;
устройство подготовки данных на перфокартах ЕС-9020.

Расширение возможностей ЭВМ ЕС-1022 осуществляется путем установки дополнительных блоков оперативной памяти и увеличения количества внешних устройств и устройств подготовки данных.

В процессоре используется микропрограммный принцип управления обработкой информации. Микропрограммное управление является основным средством управления выполнением всех инструкций, операций с пульта, процедур обмена данными с устройствами ввода-вывода. Основной функциональной частью микропрограммного устройства управления является блок постоянной памяти, в котором хранится управляющая информация в виде набора микропрограмм. Необходимое преобразование информации производится под воздействием управляющих сигналов, вырабатываемых в результате дешифрации микрокоманд, составляющих микропрограммы и считываемых из постоянной памяти. Производительность процессора (по смеси «Гибсон-1») – 80 тыс. операций в 1 с.

В составе процессора модели имеются один мультиплексный канал и два селекторных.

Мультиплексный канал может работать в мультиплексном или монопольном режиме со скоростью передачи данных соответственно до 80 и 400 кбайт/с. Число подканалов 128; число подключаемых устройств УВУ до 18. Передача информации между внешними устройствами и каналом и между каналом и ООП осуществляется побайтно.

Селекторный канал работает со скоростью передачи данных до 500 кбайт/с, число подключаемых УВУ – до 10. Между селекторными каналами и ООП информация передается по четыре байта.

Оперативная память модели построена по системе 2,5D с циклом обращения 2 мкс и разрядностью 4 байт. Цикл обращения имеет два такта: чтение и запись (регенерация). Длительность каждого такта 1 мкс. Между тактом чтения и тактом записи могут выполняться действия без обращения к ООП. Длительность этих действий равна или кратна длительности машинного такта (0,5 мкс). Оперативная память включает следующие два логически самостоятельных типа памяти: основную и мультиплексную. В основной памяти хранятся программы, исходные данные, промежуточные и окончательные результаты обработки информации. Емкость такой памяти составляет 128К байт, ее можно расширять блоками по 128К байт до 512К байт. Мультиплексная память емкостью 2048 байт используется для хранения команд, с помощью которых осуществляется управление обменом данных в мультиплексном канале.

В составе процессора имеется локальная память. Она содержит поля регистров общего назначения, регистров с плавающей запятой и вспомогательной рабочей памяти АЛУ и УУ процессора. Емкость локальной памяти равна 512 байт.

Измерение интервалов времени осуществляется с помощью таймера (датчика времени), полный цикл которого равен 15,5 ч, а интервал прерывания программ — 20 мс (таймер работает от сети частотой 50 Гц). Обновление значения таймера возможно в конце выполнения каждой команды.

В процессоре модели ЕС-1022 применяется аппаратный контроль по модулю 2 (контроль на «нечет» побайтно), а также частично метод дублирования аппаратуры.

Модель ЕС-1033 имеет среднюю производительность и предназначена для решения широкого круга научно-технических, планово-экономических и специальных задач как автономно, так и в системах обработки информации, в средних и старших звеньях АСУ. Многоплановость использования модели обеспечивается универсальной системой команд, большим объемом оперативной памяти (256 или 512К байт) и возможностью подключения большого количества периферийной аппаратуры.

Быстродействие ЭВМ при решении научно-технических задач достигает 150—200 тыс. операций в 1 с, при решении экономических задач — около 130 тыс. операций в 1 с.

В основном исполнении ЭВМ ЕС-1033 выпускается в следующем составе:

процессор ЕС-2433;
оперативное запоминающее устройство ЕС-3207 (256 или 512К байт);
стойка питания ЕС-1033/С000;
щит распределительный ЕС-1030/С005;
устройство ввода с перфокарт ЕС-6012;
устройство ввода с перфоленты ЕС-6022;
накопитель на магнитной ленте ЕС-5017 — 4 шт.;
устройство управления для НМЛ ЕС-5517;
накопитель на сменных магнитных дисках ЕС-5056М — 4 шт.;
устройство управления для НМД ЕС-5551М;
устройство вывода на перфокарты ЕС-7010;
устройство вывода на перфоленту ЕС-7022;

печатающее устройство ЕС-7032;

пишущая машинка с блоком управления ЕС-7077;

устройство нанесения и расшифровки информации на перфокартах ЕС-9011 — 2 шт.;

устройство подготовки перфолент ЕС-9024.

В двух других исполнениях ЭВМ ЕС-1033 выпускается в расширенной комплектации: оперативная память увеличивается до 512К байт, возрастает количество поставляемых НМД, НМЛ, УВВ.

Процессор ЕС-2433 является основным ядром машины и имеет магистральную организацию, обеспечивающую равнозначные связи на уровне функциональных узлов. Это устройство с микропрограммным управлением всеми процедурами. Микропрограммы хранятся в блоках постоянной памяти процессора. В процессоре предусмотрен аппаратный контроль передачи данных и выполнения операций.

В составе процессора имеются один мультиплексный канал, три селекторных СК1, СК2, СК3 и один общий. Скорость передачи данных в мультиплексном канале до 300 кбайт/с в монопольном и до 50 кбайт/с в мультиплексном режимах. В селекторном канале передача информации осуществляется со скоростью до 800 кбайт/с. Каналы СК1, СК2 и СК3 предназначены для управления высокоскоростными устройствами ввода-вывода: СК1 — устройством управления магнитными дисками ЕС-5551М, СК2 — устройством управления магнитными лентами ЕС-5517, СК3 — резервный канал. Канал МК управляет работой сравнительно медленнодействующих устройств ввода-вывода: ЕС-7077, ЕС-6012, ЕС-7032, ЕС-7010, ЕС-7022, ЕС-6022. Общий канал предназначен для сопряжения центрального процессора и оперативной памяти с каналами. Связь его с центральным процессором производится по 36-разрядной магистрали, с оперативной памятью — через блок управления ОП и с каналами МК и СК — по внутренним магистралям. Связь МК и СК с устройствами УВУ осуществляется через стандартный интерфейс ввода-вывода.

Оперативная память ЕС-3207, построенная по системе 2,5D, имеет модульную структуру. В ее состав входят восемь модулей памяти, емкость каждого модуля равна 64К байт. В качестве запоминающих элементов используются кольцевые ферритовые сердечники типа 5ВТ, электронная часть ОП выполнена на интегральных схемах и дискретных компонентах. Длительность цикла обращения равна 1,2 мкс, время выборки информации — 0,7 мкс. Запись информации в ОП может производиться полным словом или по байтам, считывание — всегда полным словом. Управление обращением к оперативной памяти осуществляется блоком управления оперативной памятью (БУОП), входящим в состав процессора. С помощью этого блока производится активизация ОП, задание режима записи и считывания, выдача адреса обращения. Запросы каналов и центрального процессора к ОП обслуживаются по принципу «первый пришел — первый обслужен», а в случае одновременного поступления запросов от центрального процессора и канала предпочтение отдается каналу. Запрос обслуживается немедленно, если ОП свободна, в противном случае запрос ждет.

Подобно тому, как это сделано в ОП модели ЕС-1022, оперативная

память ЕС-3207 состоит из двух логически самостоятельных типов памяти (основной и мультиплексной), назначение которых аналогично.

Измерение относительного времени осуществляется с помощью таймера. Полный цикл его работы равен 15,5 ч.

Модель ЕС-1052 – высокопроизводительная ЭВМ (средняя производительность 700 тыс. операций в 1 с), предназначенная для использования в крупных вычислительных центрах и высших звеньях АСУ для решения научно-технических, планово-экономических и информационных задач большего объема и высокой степени сложности.

В состав основного комплекта ЭВМ ЕС-1052 входят:

процессор ЕС-5052;

оперативное запоминающее устройство ЕС-3206 – 2 шт.;

мультиплексный канал ЕС-4012.02;

селекторный канал ЕС-4035.04 с двумя функциональными каналами (СК1, СК2) и адаптером канал-канал ЕС-4060;

пульт управления ЕС-1501;

распределительный шкаф ЕС-0853;

накопитель на магнитной ленте ЕС-5017 – 8 шт.;

устройство управления накопителями на магнитной ленте ЕС-5517;

накопитель на сменных магнитных дисках ЕС-5061 – 9 шт.;

устройство управления накопителями на магнитных дисках ЕС-5568;

устройство ввода с перфокарт ЕС-6019 – 2 шт.;

устройство ввода с перфоленты ЕС-6022 – 2 шт.;

устройство вывода на перфокарты ЕС-7010;

устройство вывода на перфоленту ЕС-7022 – 2 шт.;

пишущая машинка с блоком управления ЕС-7077 – 2 шт.;

печатающее устройство ЕС-7032 – 2 шт.;

комплекс устройств группового управления дисплеями ЕС-7906;

устройство подготовки данных на магнитной ленте ЕС-9002 – 2 шт.;

устройство подготовки данных на перфокартах ЕС-9011 – 2 шт.;

устройство подготовки данных на перфоленте ЕС-9024.

Расширение состава машины возможно за счет следующих устройств: графического регистрирующего устройства планшетного типа ЕС-7051, узкоформатного (ЕС-7052) и широкоформатного (ЕС-7053) графических регистрирующих устройств барабанного типа, устройства ввода-вывода алфавитно-цифровой и графической информации на ЭЛТ ЕС-7064, выносного пульта ЕС-7066.

В процессоре для получения высокой производительности реализован принцип схемного управления и пришита структура с несколькими уровнями совмещения процесса выполнения программы. Используется универсальный набор команд. Разрядность арифметического устройства – 64 двоичных разряда.

Мультиплексный канал в своем составе имеет 224 мультиплексных и два селекторных подканала. Пропускная способность канала достигает 425 кбайт/с, максимальная пропускная способность мультиплексного подканала составляет 110 кбайт/с, селекторного подканала – 180 кбайт/с.

Количество интерфейсов ввода-вывода – 3. Разрядность информации обмена с оперативной памятью – 8 байт, с внешними устройствами – 1 байт.

Селекторный канал ЕС-4035.04 организует обмен информацией между ОЗУ и ВЗУ в монопольном режиме передачи данных. В устройстве ЕС-4035.04 имеется два канала – СК1 и СК2. Максимальная пропускная способность каждого из них – 1,25 Мбайт/с. Адресация ВЗУ – от 0 до 255. Разрядность информации обмена с ОЗУ и ВЗУ – та же, что и для МК.

Устройство ЕС-3206 используется в качестве основной оперативной памяти. Емкость устройства – 1024К байт, длина слова – 16 байт, количество автономных блоков памяти – 2 (емкость одного блока – 512К байт). Время выборки информации из блока памяти – 650 нс, время цикла обращения – 1,25 мкс.

В машине используется развитая система программно-аппаратного контроля. В оперативной памяти предусмотрены обнаружение многократных ошибок и корректировка одиночных ошибок.

Все ЭВМ ЕС – многопрограммные. Поэтому перечисленные выше модели имеют также средства распределения и защиты памяти, приоритетные системы прерывания, датчики времени. Для обеспечения возможности построения вычислительных систем в машинах предусматриваются мультисистемные средства внешних связей, системные средства управления межпроцессорными и межмашинными обменами информацией, формирования единого общесистемного времени и т. п. Особенности построения основных устройств машин и устройств, расширяющих возможности их использования, будут рассмотрены в следующих главах.

1.4. Система программного обеспечения ЕС ЭВМ

Система программного обеспечения (СПО) ЕС ЭВМ представляет собой комплекс программных средств, предназначенных для повышения эффективности использования машины, облегчения ее эксплуатации и снижения трудоемкости подготовки программ пользователей. Она обеспечивает решение широкого класса научно-технических, экономико-статистических, управленческих и других задач, а также построение на базе ЕС ЭВМ различного рода вычислительных систем, автоматизированных систем управления и обработки информации. СПО выполняет роль посредника между пользователями и ЭВМ, обеспечивая для них удобный уровень общения с машиной.

Основные функции СПО ЕС ЭВМ:

автоматическое управление вычислительным процессом, 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, главным образом мультипрограммных режимов ее работы, за счет автоматически выполняемого рационального распределения ресурсов системы (машинного времени процессора, емкости памяти, устройств ввода-вывода, системных программных модулей, стандартных программ) при одновременном решении нескольких

задач, наконец, за счет минимального вмешательства оператора или программиста в вычислительный процесс;

обеспечение взаимодействия вычислительной системы и оператора в формах, удобных для оператора;

обеспечение сокращения времени на подготовку задач к решению на ЭВМ с помощью системы автоматизации программирования, являющейся составной частью СПО ЕС ЭВМ;

обеспечение контроля и надежности функционирования ЭВМ с помощью наладочных, контролирующих и диагностических программ.

В сущности, система программного обеспечения ЕС ЭВМ относится к так называемому общему программному обеспечению, которое, выполняя вышеуказанные функции, решает задачу применения ЭВМ как некоторой универсальной системы обработки информации. Кроме общего, имеется специальное программное обеспечение, представляющее собой комплекс программ, которые добавляются к общему программному обеспечению и решают задачу применения ЭВМ как некоторой специализированной системы обработки информации.

Общее программное обеспечение (рис. 1.2) состоит из четырех частей [45]: операционных систем (ОС), набора пакетов прикладных программ (ППП), комплекса программ технического обслуживания (КПТО) и системы документации на СПО ЕС ЭВМ.

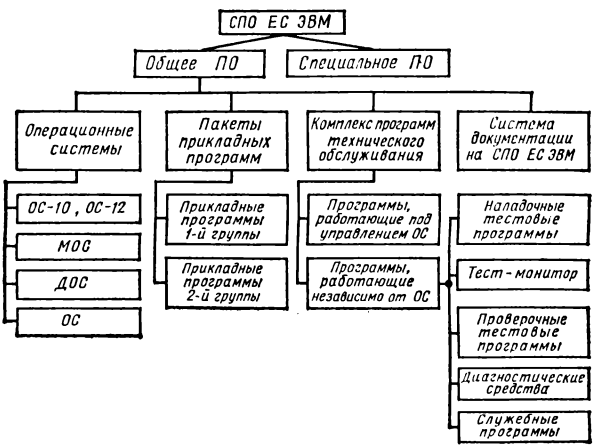


Рис. 1.2. Структура системы программного обеспечения ЕС ЭВМ

Операционные системы и комплекс программ технического обслуживания вместе с документацией на СПО составляют базовое программное обеспечение (БПО) ЕС ЭВМ. Пакеты прикладных программ входят в систему прикладного программного обеспечения.

Операционные системы ОС объединяют средства организации и управления процессом функционирования ЭВМ в различных режимах. Перечень и объем операций, выполняемых операционной системой, различны для различных типов ОС. В настоящее время используются четыре типа операционных систем:

ОС-10 и ОС-12 для моделей ЕС-1010 и их усовершенствованных модификаций ЕС-1012 разработки ВНР;

МОС (малая операционная система) для моделей ЕС-1021 и их последующих модификаций разработки ЧССР;

ДОС ЕС (дисковая операционная система) для всех других моделей ЕС ЭВМ в малой конфигурации;

ОС ЕС для всех моделей ЕС ЭВМ (кроме ЕС-1010 и ЕС-1021) в средней и расширенной конфигурациях.

Подробно структура и функции операционных систем будут рассмотрены в гл. 14.

В наибольшей степени требованиям, которые предъявляются к операционным системам, отвечает ОС ЕС. Главные из них состоят в следующем.

ОС должна быть способной к расширению своих функциональных возможностей, к изменению и пополнению новыми программными компонентами, быть структурно гибкой. Это обусловливается 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, в частности терминальная аппаратура. Обеспечение расширения функциональных возможностей ОС с целью применения 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 (на управлении определенной части вычислительного процесса), выполняемой в вычислительной системе. При этом один и тот же модуль может использоваться для нескольких различных программ. Модули формируются и отлаживаются независимо один от другого. Благодаря модульной структуре программы ОС легко поддаются изменению либо заменой модулей, либо подключением новых, либо перестановкой модулей, определяющих процесс обработки информации.

ОС должна обладать свойством адаптируемости, приспособления к изменению условий функционирования вычислительной системы (см. гл. 18).

ОС должна обеспечивать предоставление абонентам максимальных удобств в сочетании с эффективным использованием ЭВМ.

ОС должна способствовать сведению к минимуму простоев оборудования (особенно процессора), связанного с вмешательством оператора или программиста в вычислительный процесс.

Потери времени на работу самой операционной системы должны быть минимальными. Эти потери существенно зависят от соотношения между программной и аппаратной частями ОС (строго говоря, операционная система представляет собой программно-аппаратный комплекс средств организации и управления процессом функционирования ЭВМ). В общем случае чем больше удельный вес аппаратной части, тем сложнее ЭВМ, но тем меньше затрачивается времени на работу ОС, и наоборот. В аппаратную 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 ОС 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. Уровень развития интегральной микроэлектроники в настоящее время обусловил тенденцию повышения удельного веса аппаратной части операционной системы с целью сокращения времени на работу ее компонентов и, следовательно, повышения производительности ЭВМ.

В составе операционных систем ЕС ЭВМ можно выделить следующие наиболее важные функциональные группы:

программы начального пуска, управляющие операцией начального ввода информации в ООП машины и осуществляющие настройку ОС на заданный режим работы;

программы управления выполнением потока заданий, диспетчеризации вычислений, организации прерываний, планирования и управления мультипрограммной работой в различных режимах;

программы управления вводом-выводом информации, обменом информацией между ОП машины и УВВ и ВЗУ, между ЭВМ и удаленными абонентами;

программы управления процессом контроля работоспособности и надежности функционирования ЭВМ.

В состав операционных систем ЕС ЭВМ включены средства, обеспечивающие снижение трудоемкости подготовительной работы при решении задач на ЭВМ. Эти средства составляют систему автоматизации программирования (САП). Система автоматизации программирования — это часть СПО, предназначенная для облегчения и автоматизации процесса получения рабочих программ пользователей. Развитые САП включают алгоритмические языки (проблемно-ориентированные, машинно-ориентированные), трансляторы (программы перевода рабочих программ с алгоритмических языков на машинный язык), интерпретирующие и компилирующие системы, библиотеки стандартных подпрограмм, программы сервиса и т. д. Объем и разнообразие средств, имеющихся в САП, в конечном счете определяют «квалификацию» ЭВМ, удобство использования ее программистами и абонентами.

Пакеты прикладных программ (ППП) составляют значительную часть СПО ЕС ЭВМ и представляют собой комплексы программ, которые вместе с документацией, необходимой для их установки и эксплуатации, предназначены для решения определенных задач или класса задач. Следовательно, ППП — это обособленный элемент системы прикладного программного обеспечения. Обычно пакет ориентируется на решение задач большего объема и большой сложности. Пакеты прикладных программ представляют собой базу для построения интегрированных систем обработки данных. Они должны удовлетворять требованиям, предъявляемым к про-

граммам пользователей теми операционными системами, под управлением которых осуществляется их функционирование. В рамках СПО ЕС ЭВМ пакеты разрабатываются с учетом возможности их работы под управлением ОС различных типов и модификаций и на различных конфигурациях технических средств.

Вообще говоря, пакеты прикладных программ могут быть классифицированы по ряду признаков следующим образом [45].

1. По ориентации на реализацию определенного метода решения задач или на применение этого метода к реализации отдельной прикладной задачи различают методо-ориентированные и проблемно-ориентированные пакеты.

Методо-ориентированные пакеты создаются в тех случаях, когда в данной области накопились и появляются все новые прикладные задачи, требующие для своего решения применения одних и тех же математических методов. Алгоритмы решения этих задач могут быть различны, но метод решения один. Применение методо-ориентированных ППП не освобождает пользователей от необходимости программирования для решения конкретной прикладной задачи, а лишь упрощает процесс программирования, предоставляя удобные средства для описания алгоритма решения.

Проблемно-ориентированный пакет представляет собой программную реализацию решения определенной прикладной задачи, причем такой, в решении которой возникает регулярная необходимость у большого числа пользователей. Проблемно-ориентированные ППП непосредственно реализуют алгоритмы решения задач пользователей. Они, как правило, исключают для пользователей необходимость программирования. Проблемно-ориентированные пакеты могут быть созданы на базе методо-ориентированных ППП.

2. По структуре (или иначе – внутренней организации, форме программирования) прикладные программы могут быть разделены на три группы:

библиотеки прикладных программ (БПП);

системы программирования с использованием специализированных языков;

программные системы.

Эти группы достаточно полно охватывают всю систему прикладного программного обеспечения, хотя среди используемых в настоящее время программ имеются и такие, которые выходят за рамки данной классификации или занимают промежуточное положение. Следует заметить, что прикладные программы, относящиеся к различным классификационным группам по своей структуре, существенно различаются по степени автоматизации процесса вычислений, обеспечиваемой этими программами, иначе говоря, по степени участия пользователя или программиста в органаизации процессов подготовки и выполнения рабочих программ. Это естественно, так как понятие «структура программы» или «структура пакета» включает общие принципы построения программ пакета, его функционирования и эксплуатации.

Библиотека прикладных программ представляет собой наиболее простую (по сравнению с двумя другими структурами прикладных программ) форму программирования. В сущности, БПП – это универсальный набор

программ, применимый для решения различных, не связанных между собой задач. Каждая программа БПП имеет самостоятельное значение и используется для решения обычно достаточно простой задачи. Для ЭВМ первого и второго поколений библиотеки программ составляли основу систем обработки данных, однако они широко используются и при работе с ЕС ЭВМ. В зависимости от состава БПП и характера решаемых задач пользователь либо непосредственно применяет отдельные программы библиотеки для решения своих задач, либо использует их для реализации отдельных частей задачи. Расширение БПП осуществляется путем простого добавления новых программ, и в этом заключается преимущество библиотеки программ как формы программирования. Однако для широкого пользователя применение библиотек программ ограничено, так как при работе с ними необходимы достаточно глубокие знания основ программирования и принципов функционирования программного обеспечения.

Система программирования с использованием специализированных языков как форма программирования применима при разработке главным образом методо-ориентированных прикладных программ. Это объясняется тем, что практически с помощью специализированных языков (например, СИМСКРИПТ и GPSS, предназначенных для решения задач статистического моделирования) можно описывать алгоритм решения задачи на уровне укрупненной блок-схемы, а не детально, как это делается с помощью алгоритмических языков высокого уровня. Такая система программирования требует от программиста хорошего знания не только используемого специализированного языка, но и метода решения прикладных задач.

Наиболее высокий уровень внутренней организации характерен для ППП, которые представляют собой программные системы. Каждая программная система — это комплекс программных средств, имеющих свою внутреннюю системную организацию и ориентированных на решение определенного класса задач. Современные ППП по своей структуре разрабатываются преимущественно как программные системы. Каждый такой пакет включает следующие составные части:

набор обрабатывающих программных модулей, составляющих тело пакета и предназначенных для непосредственной обработки данных, т. е. для решения задач пользователей. Программные модули могут быть написаны на различных алгоритмических или машинно-ориентированных языках, если это способствует повышению эффективности использования пакета;

управляющую программу пакета (диспетчер), предназначенную для управления процессом обработки данных, осуществляемым обрабатывающими модулями. Для работы на ЭВМ третьего поколения управляющая программа реализуется обычно в форме описания решаемой задачи на языке управления заданиями, т. е. средствами операционной системы. По запросу на решение задачи управляющая программа выбирает необходимые обрабатывающие модули и формирует из них так называемую рабочую обрабатывающую программу;

комплекс обслуживающих программ, которые выполняют вспомога-

тельные функции: редактирование библиотеки пользователя, диагностику ошибок и т. д.;

средства для обеспечения генерации пакета, т. е. автоматического формирования пакета, ориентированного на решение конкретных задач пользователя на конкретной конфигурации технических средств.

Пакеты прикладных программ, построенные в виде программных систем, в настоящее время получили самое широкое распространение, поскольку они, представляя наиболее развитую форму программирования, обладают высокой степенью автоматизации процесса решения задач путем автоматического создания рабочей программы по запросам. От пользователя требуется лишь умение описать исходные данные и требуемый запрос на работу программы на языке программной системы. Эта форма программирования характеризуется также большой гибкостью, т. е. возможностью внесения изменений различными способами достаточно оперативно и без существенных затрат средств.

Следует заметить, что проблемно-ориентированные ППП обычно реализуются в форме программных систем. Это объясняется тем, что такие пакеты рассчитаны на применение в тех областях, где основную роль играет широкий пользователь и где, следовательно, высокая степень автоматизации вычислительного процесса является главным условием эффективной эксплуатации программ.

3. По сферам применения и классам решаемых задач ППП можно разделить на следующие группы:

расширяющие возможности операционных систем;

общего назначения;

для решения основных инженерных и научно-технических задач;

для решения задач автоматизированных систем управления (АСУ);

специального назначения.

К пакетам, расширяющим возможности ОС, относят те, которые обеспечивают функционирование различных конфигураций ЕС ЭВМ, но не включаются в состав основного комплекта ОС. Иначе говоря, функции, выполняемые этими пакетами, не обеспечиваются средствами генерации основных операционных систем ЕС ЭВМ. В то же время расширение возможностей ОС проводится на базе основных ОС для специальных комплексов технических средств и способов их применения (для многопроцессорных и многомашинных систем, для систем телеобработки).

К числу ППП, расширяющих возможности ОС, относят пакеты «Супервизор реального времени», «Ока», «Кама» и др.

Пакет прикладных программ «Супервизор реального времени» применяется для параллельного использования ресурсов вычислительной системы большим количеством пользователей, располагающих терминальными (стандартными и нестандартными) устройствами ввода-вывода. Он обеспечивает эффективное использование ЕС ЭВМ в системах управления объектами в реальном масштабе времени. Этот пакет дает возможность использования методов и техники программирования, соответствующих условиям применения систем управления процессами или объектами при жестких ограничениях на время ответа. По своей ориентации и структуре он представляет собой проблемно-ориентированный пакет, построенный в виде программной системы.

Пакет прикладных программ «Ока» является системой управления базами данных. Она ускоряет создание и повышает эффективность широкого класса систем обработки информации, в том числе информационно-поисковых и информационно-справочных систем, автоматизированных систем управления, систем автоматизированного проектирования (САПР). Система «Ока» реализуется в виде обрабатывающей программы ОС ЕС и по своим функциям расширяет возможности операционной системы. Однако она удовлетворяет требованиям различных областей применения, поэтому ППП «Ока» можно отнести к категории пакетов общего назначения.

Система телеуправления данными «Кама» по своей структуре представляет собой программную систему, а по ориентации — проблемно-ориентированный пакет. Она обеспечивает средства, необходимые для стандартных применений терминалов: коммутации сообщений, выдачи справок, сбора данных в диалоговом и пакетном режимах. Система реализована в виде многоцелевого интерфейса связи между ОС ЕС и прикладными программами пользователей.

Пакеты прикладных программ общего назначения являются программной реализацией прикладных задач и математических методов их решения, которые относятся к категории общих для многих сфер применения ЭВМ. Это задачи и методы теории массового обслуживания, статистические методы обработки данных, методы математического программирования, используемые в самых различных областях. В ряде случаев ППП общего назначения представляют базу для решения сложных прикладных задач, и тогда их строят по типу методо-ориентированных пакетов.

Характерной особенностью инженерных и научно-технических задач является большое их разнообразие и постоянное появление новых задач. В связи с этим программная реализация таких задач в виде сложных программных систем ограничена по причине экономической нецелесообразности. Здесь более широкое применение находят библиотеки программ, специализированные языки программирования, а также методо-ориентированные ППП.

Основным требованиям, предъявляемым к прикладному программному обеспечению АСУ, разработка которого составляет значительную долю общих затрат на создание автоматизированных систем управления, является требование типизации. Типовой прикладной программой считается такая программа, которая допускает внесение изменений без существенных затрат средств и времени. Следовательно, типовая программа должна быть легко адаптируемой к особенностям конкретного применения. Принципиальная возможность типизации прикладных программ АСУ обеспечивается определенным единством форм и методов управления, реализуемых в автоматизированных системах управления. Необходимой предпосылкой типизации является программная совместимость ЭВМ, используемых в АСУ. Важнейшее средство для обеспечения совместимости — применение проблемно-ориентированных языков программирования.

Имеются пакеты прикладных программ, специально предназначенные для решения вопросов совместимости различных ЭВМ. Например, ППП

«Совместимость в ДОС ЕС» используется для переноса программ и данных, разработанных применительно к ЭВМ «Минск-32», на ЕС ЭВМ. Он позволяет применять эти программы и данные при решении задач, записанных на алгоритмических языках ФОРТРАН и КОБОЛ. В состав этого пакета входят следующие компоненты: программа переноса данных на магнитной ленте, программа переноса данных на перфокартах, макрокоманды переноса данных, конвертор данных языка КОБОЛ, конвертор программ языка ФОРТАН, конвертор программ языка КОБОЛ. Конвертор программ КОБОЛа в сочетании с конвертором данных КОБОЛа обеспечивает программную совместимость ЭВМ «Минск-32» с ЕС ЭВМ на уровне языка КОБОЛ. Конвертор ФОРТРАНа ЭВМ «Минск-32» предназначен для автоматического преобразования программ, закодированных на ФОРТРАНе ЭВМ «Минск-32», в программы на ФОРТРАНе IV ДОС ЕС. Средства переноса данных позволяют воспринимать данные с носителей ЭВМ «Минск-32», перекодировать данные из кодов ЭВМ «Минск-32» в коды ЕС ЭВМ, производить перекомпоновку данных, обработку и замену контрольной информации, преобразовывать данные ЭВМ «Минск-32» в формат совместимости эмуляции.

В системе программного обеспечения ЕС ЭВМ выделяются две группы прикладных программ (см. рис. 1.2):

1. Прикладные программы, обеспечивающие расширение возможностей операционной системы для специальных комплексов технических средств и способов применения.

2. Прикладные программы, 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.

Телеобработка включает широкий круг применений. В системе телеобработки осуществляется обмен информацией между АП и центральным вычислительным органом по телефонным и телеграфным каналам связи.

Средства телеобработки обеспечивают обработку данных и заданий на расстоянии, решение задач в оперативном режиме (в режиме разделения времени).

Предусматриваются три вида обработки данных на расстоянии: коммутация сообщений, обработка запросов и сбор данных.

Коммутация сообщений – это такой вид телеобработки, при котором сообщения, полученные центральным вычислительным органом от некоторого АП, передаются одному или нескольким другим АП.

Обработка запросов представляет собой вид телеобработки, при котором в соответствии с запросами от АП центральная ЭВМ 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 АП. 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 ЭВМ происходит накопление и последующая обработка данных, поступающих от АП. Примером системы сбора данных может служить система учета готовой продукции некоторого предприятия: данные о количестве выработанной продукции каждого наименования поступают с рабочих мест через АП в течение смены и накапливаются в центральной ЭВМ. 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 ЭВМ с одного или нескольких АП и выполняются системой в режиме пакетной обработки. Результаты обработки либо отправляются на соответствующие АП, либо засылаются в библиотеку на хранение. Абонент, находящийся на АП, имеет доступ ко всем средствам центральной ЭВМ.

Решение задач в оперативном режиме, называемом также режимом разделения времени, представляет собой такой вид телеобработки, при котором ряд отдельных абонентов имеет возможность непосредственно, одновременно и оперативно использовать центральную вычислительную систему для решения своих задач. Работа в режиме разделения времени часто осуществляется в форме диалога между абонентом и программой, которая выполняется центральной ЭВМ 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 (дисплеи). Эти средства упрощают поиск, обработку и воспроизведение информации в графической форме. Они включают расширенный набор макрокоманд, управляющие программы, обрабатывающие программы, средства манипулирования данными, программы анализа прерываний от ввода-вывода, диагностические 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 использовании 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 их создания, анализа и преобразования 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 понимают системы, предназначенные для обработки банков данных, т. е. больших массивов взаимосвязанных данных, содержащих минимальную избыточную информацию. 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 программное обеспечение графопостроителей и устройств ввода-вывода с перфолент.

Программное обеспечение графопостроителей представляет собой набор программ, обеспечивающих формирование и вывод изображений на графопостроители под управлением операционной системы ОС ЕС. Основные компоненты этого программного 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 их работе;

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 – набор стандартных программ, обеспечивающих вывод на графопостроитель ряда часто используемых изображений;

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

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

Программное обеспечение устройств ввода-вывода с перфолент – совокупность программ, входящих в состав управляющей программы ОС ЕС 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 перфоленты производится обратное преобразование кодов. Предусматривается ввод информации с перфоленты и вывод на перфоленту без программного преобразования кодов.

На базе моделей ЕС ЭВМ могут строиться многомашинные вычислительные комплексы с различными уровнями комплексирования. Например, могут строиться многомашинные комплексы с вспомогательным процессором или комплексы, составленные из двух однотипных машин. Программное 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 быть использованы в качестве машин-спутников, связанных с машинами Единой системы более высокой производительности через каналы связи. Например, на базе машин 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.

Преемственность 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 программного обеспечения включает пакеты прикладных программ, предназначенных для решения широкого круга задач линейного и нелинейного программирования, сетевого анализа, теории запросов, 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 используют 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 назначение – это обработка массивов экономической и научно-технической информации и организация больших информационно-поисковых систем.

Комплекс программ технического обслуживания ЕС ЭВМ. Комплекс 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 нее.

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, 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, который выполняется 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 реализует оборудование, встроенное в процессор. С помощью диагностических средств реализуются такие методы диагностики, как метод диагностических таблиц, метод диагностических характеристик, метод списков подозреваемых элементов.

5. 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, включение новых тестов, исключение и исправление тестов.

Система документации на СПО ЕС ЭВМ. Она представляет собой самостоятельную часть программного обеспечения и предназначена для обеспечения эффективного применения, изучения и обслуживания программных средств ЕС ЭВМ.

Глава 2

КОДИРОВАНИЕ ДАННЫХ И КОМАНД

2.1. Системы счисления, применяемые в ЕС ЭВМ

В цифровых машинах Единой системы *все виды кодирования и представления данных и команд основаны на использовании двоичных цифр:* 1 и 0.

Определенными наборами этих цифр в устройствах ЭВМ представляются команды, различная служебная информация, такая как признаки данных и результатов, коды, отображающие состояние устройств машины и т. п. Преобразуемые данные также выражаются определенными совокупностями двоичных цифр. При этом в ЭВМ Единой системы, как и во многих других типах ЭВМ, данные могут представляться либо в виде двоичных чисел, либо как десятичные числа или любой произвольный текст, соответственно цифры или символы которых закодированы числами двоичной системы счисления. Такие широкие возможности представления данных позволяют использовать машины для решения самых разнообразных задач.

Кодирование десятичных чисел основано на *двоично-десятичном изображении* их цифр.

Двоично-десятичное изображение цифр. Для вычислений непосредственно в десятичной системе счисления каждая цифра десятичного числа кодируется четырехразрядным двоичным числом (тетрадой), например:

Десятичная цифра	0	1	2	3	4	5	6	7	8	9
Двоичный код	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Тогда десятичное число, например $A_{10} = 1978$, в двоично-десятичной записи будет иметь вид

$$A_{2-10} = 0001\ 1001\ 0111\ 1000.$$

Если при выполнении программы действия должны производиться над двоичными числами, то исходные десятичные числа, 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,

$$A_N = \pm (a_{m-1}N^{m-1} + a_{m-2}N^{m-2} + \ldots + a_1N^1 + a_0N^0 + a_{-1}N^{-1} + \ldots + a_{-l}N^{-l}) \quad (2.1)$$

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

$$A_N^{\text{ц}} = \pm ((\ldots((a_{m-1}N + a_{m-2})N + a_{m-3})N + \ldots)N + a_0) \qquad (2.2)$$

и

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

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.

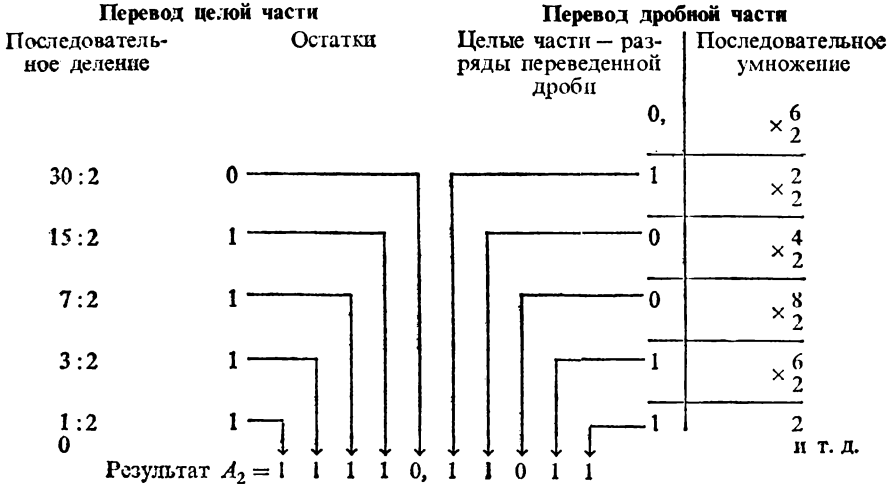
При переводе целой части числа в ходе процесса последовательного деления получаются остатки, выраженные цифрами исходной системы счисления.

Они представляют собой цифры a_0, a_1, ... *целой части числа, переведенного в новую систему счисления.* Последний остаток является старшей цифрой переведенного числа.

При переводе дробной части числа при каждом умножении получаются целые части, которые исключаются из последующих умножений. *Эти целые части,* изображенные цифрами исходной системы счисления, *представляют собой цифры дробной части, переведенной в новую систему счисления.* Значение первой целой части является первой цифрой после запятой переведенного числа.

Рассмотрим примеры, поясняющие общие процессы перевода чисел из одной системы счисления в другую.

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

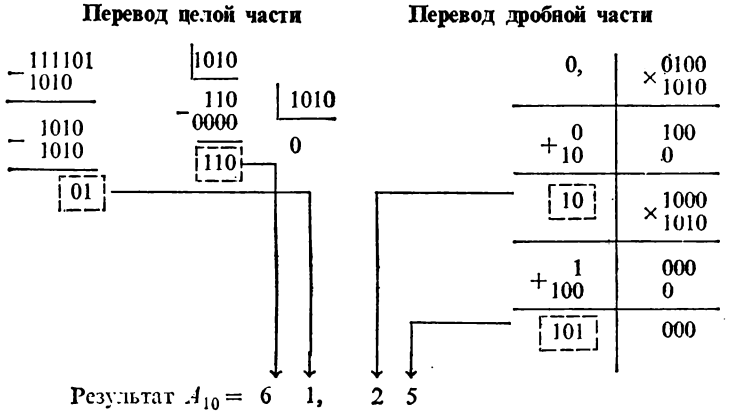


При переводе дробной части в любом случае, в том числе и когда получается периодическая дробь, как в рассматриваемом примере, определяются все разряды после запятой в пределах разрядной сетки, в которой представляются преобразуемые данные. В данном случае все число A_2 определено в пределах 10 двоичных разрядов.

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

Пример 2.2. Двоичное число $A_2 = 111101{,}01$ перевести в десятичную систему счисления.

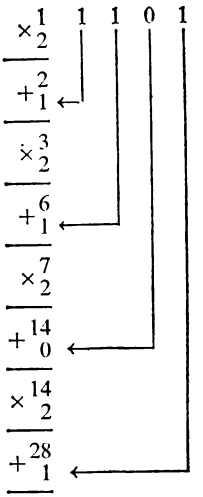
Основание десятичной системы счисления в двоичной записи имеет вид 1010. Поэтому при переводе целую часть $A_2^{ц}$ будем последовательно делить, а дробную часть $A_2^{др}$ последовательно умножать на число 1010.



Десятичные эквиваленты разрядов искомого числа находим, пользуясь примером на с. 41.

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 бо́льшим 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.

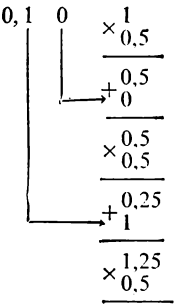
Пример 2.3. Перевести двоичное число $A_2^{ц} = 11101$ в десятичную систем счисления.



Результат $A_{10} = 29$

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 $A_2^{\text{др}} = 0{,}101$ в десятичную систему счисления.



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.*

Шестнадцатиричная система счисления. Эта система 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 крайние правая и левая тетрады будут неполными, то двоичное число необходимо соответственно справа и слева дополнить нулями, как это сделано в следующем примере.

Пример 2.5. Требуется преобразовать двоичное число

$$A_2 = 11011011011000011,\ 10101000111.$$

0001 1011 0110 1100 0011, 1010 1000 1110

1 B 6 C 3 A 8 E

Таким образом, $A_{16} = 1B6C3,A8E$.

Достоинство шестнадцатиричной системы счисления состоит в том, что она позволяет делать достаточно компактные записи двоичных чисел, что очень важно при больших диапазонах разрядности чисел, команд и специальных двоичных слов, которыми оперируют ЭВМ Единой системы. Например, адреса основной оперативной памяти могут иметь до 24 двоичных разрядов, специальные слова, отражающие состояние вычислительного процесса, имеют 64 двоичных разряда и т. д. Поэтому возможность делать достаточно короткие записи больших двоичных чисел явилась одной из первых причин применения шестнадцатиричной системы счисления.

Кроме того, в *ЕС ЭВМ в качестве основной единицы информации используется восьмиразрядный двоичный код – байт.* Переменные разрядности чисел и команд установлены кратными байту. Двоичный код байта при записи удобно представляется двухразрядным шестнадцатиричным числом.

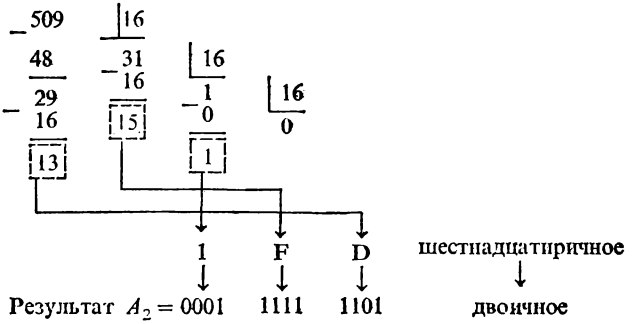
В шестнадцатиричной записи наглядно изображаются и десятичные числа.

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

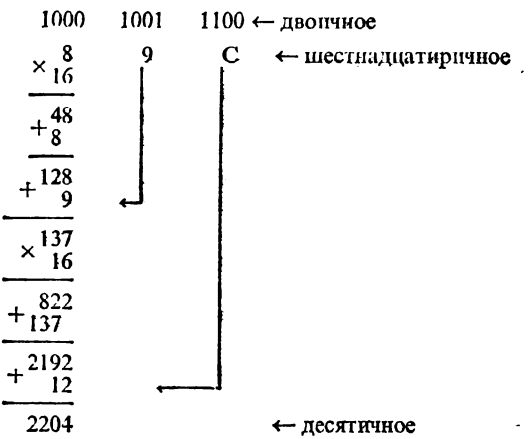
Как будет показано в п. 2.4, применение шестнадцатиричной системы счисления для выражения чисел в полулогарифмической форме с плавающей запятой, кроме ускорения выполнения операций, расширяет допустимый диапазон представления нормализованных чисел.

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 12-разрядное число $A_2 = 100010011100$ в десятичное



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, ..., Z, a, b, c, ..., z*), арабские цифры (0, 1, 2, ..., 9), ограничители (знаки арифметических и логических операций, разделители, скобки – см. табл. 3 приложения);

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 (см. табл. 4 приложения).

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 (см. табл. 5 приложения), букв русского алфавита и нескольких графических знаков.

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 применяются четыре системы, одна из которых основана на представлении символов 7-разрядными кодами, две – 8-разрядными и еще одна (12-позиционная) предназначена для представления символов на 12-позиционных перфокартах. Последняя система кодирования рассматривается в гл. 12.

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 перфоленты. Графические и управляющие символы полного алфавита распределены по трем кодовым таблицам (ГОСТ 13052–74): КОИ-7$Н_0$, КОИ-7$Н_1$ и КОИ-7$С_1$. Первые две таблицы различаются столбцами 4–7. В табл. КОИ-7$Н_0$ в этих столбцах находятся буквы русского алфавита, а в табл. КОИ-7$Н_1$ – латинского. Управляю-

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

	КОИ-7Н$_0$							
$б_7$	0	0	0	0	1	1	1	1
$б_6$	0	0	1	1	0	0	1	1
$б_5$	0	1	0	1	0	1	0	1
$б_4$ $б_3$ $б_2$ $б_1$								
0 0 0 0	0	1	2	3	4	5	6	7
0 0 0 1	0	НЗ	СУ1	|	1	а	я	А	Я
0 0 1 0	1	НТ	СУ2	„	2	б	р	Б	Р
0 0 1 1	2	КТ	СУ3	#	3	ц	с	Ц	С
0 1 0 0	3	КП	СТП	¤	4	д	т	Д	Т
0 1 0 1	4	КТМ	НЕТ	%	5	е	у	Е	У
0 1 1 0	5	ДА	СИН	&	6	ф	ж	Ф	Ж
0 1 1 1	6	ЗВ	КБ	'	7	г	в	Г	В
1 0 0 0	7	ВШ	АН	(	8	х	ь	Х	Ь
1 0 0 1	8	ГТ	КН	)	9	и	ы	И	Ы
1 0 1 0	9	ПС	ЗМ	*	:	й	з	Й	З
1 0 1 1	10	ВТ	АР2	+	;	к	щ	К	Ш
1 1 0 0	11	ПФ	РФ	,	<	л	э	Л	Э
1 1 0 1	12	ВК	РГ	–	=	м	щ	М	Щ
1 1 1 0	13	Вых	РЗ	.	>	н	ч	Н	Ч
1 1 1 1	14	Вх	РЭ	/	?	о	_	О	ЗБ

	КОИ-7Н$_1$							
$б_7$	0	0	0	0	1	1	1	1
$б_6$	0	0	1	1	0	0	1	1
$б_5$	0	1	0	1	0	1	0	1
$б_4$ $б_3$ $б_2$ $б_1$								
0 0 0 0	0	1	2	3	4	5	6	7
0 0 0 1	ПУС	АР1	Пробел	0	@	Р	`	р
0 0 1 0	НЗ	СУ1	!	1	А	Q	а	q
0 0 1 1	НТ	СУ2	„	2	В	R	b	r
0 1 0 0	КТ	СУ3	#	3	С	S	с	s
0 1 0 1	КП	СТП	¤	4	D	Т	d	t
0 1 1 0	КТМ	НЕТ	%	5	Е	U	е	u
0 1 1 1	ДА	СИН	&	6	F	V	f	v
1 0 0 0	ЗВ	КБ	'	7	G	W	g	w
1 0 0 1	ВШ	АН	(	8	Н	Х	h	х
1 0 1 0	ГТ	КН	)	9	I	Y	i	y
1 0 1 1	ПС	ЗМ	*	:	J	Z	j	z
1 1 0 0	ВТ	АР2	+	;	К	[	k	{
1 1 0 1	ПФ	РФ	,	<	L	\	l	|
1 1 1 0	ВК	РГ	–	=	М	]	m	}
1 1 1 1	Вых	РЗ	.	>	N	¬	n	~
	Вх	РЭ	/	?	0	_	o	ЗБ

	КОИ-7С$_1$	
$б_7$	1	1
$б_6$	0	0
$б_5$	0	1
$б_4$ $б_3$ $б_2$ $б_1$		
0 0 0 0	4	5
0 0 0 1	ВЦФ	
0 0 1 0	НЗН	
0 0 1 1	РП	УУК
0 1 0 0		
0 1 0 1	БК	ВКП
0 1 1 0	НС	ОСУ
0 1 1 1	НП	ВП
1 0 0 0	ОЖД	
1 0 0 1		
1 0 1 0		
1 0 1 1	УР	
1 1 0 0	СП2	СП3
1 1 0 1		ВЫП
1 1 1 0		ВСТ
1 1 1 1	НРВ	
	СП1	

щие символы и специальные знаки в обеих таблицах одинаковы. Кодовая таблица КОИ-7С$_1$ предназначена для кодирования дополнительных управляющих символов (см. табл. 5 приложения).

Согласно таблицам каждому символу соответствует 7-разрядный двоичный код $б_7б_6б_5б_4б_3б_2б_1$. Позиции в таблицах обычно обозначают дробью, числитель которой указывает номер столбца, а знаменатель — номер строки. Например, букве *а* в кодовой таблице КОИ-7Н$_0$ (позиция 4/1) соответствует двоичный код 1000001. В таблице КОИ-7Н$_1$ в аналогичной позиции находится буква *А*, а в таблице КОИ-7С$_1$ — управляющий символ НЗН. Для однозначности кодирования графических символов из столбцов 2—7, если используется табл. КОИ-7Н$_0$, то перед одиночным символом или группой символов посылается код символа ВХ, если табл. КОИ-7Н$_1$, то — код символа ВЫХ. При отсутствии этих дополнительных символов в начале передачи кодирование автоматически идет по таблице КОИ-7Н$_0$. Перед передачей каждого символа таблицы КОИ-7С$_1$ посылается символ АР2.

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 переключающие символы 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.

Таблица 2.2

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

		Зональная тетрада														
a_8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
a_7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
a_6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
a_5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$a_8a_7a_6a_5a_4a_3a_2a_1$		00	01	02	03	04	05	06	07	09	09	10	11	12	13	14	15
0 0 0 0	0	ПУС	АР1	Пров. вет	0	@	P	`	p	ВЦФ				ю	п	Ю	П
0 0 0 1	1	НЗ	СУ1	!	1	A	Q	a	q	НЗН					я		Я
0 0 1 0	2	НТ	СУ2	"	2	B	R	b	r	РП	УУК			б		Б	
0 0 1 1	3	КТ	СУ3	#	3	C	S	c	s					ц		Ц	
0 1 0 0	4	КП	СТП	¤	4	D	T	d	t	БК	ВКП			д		Д	
0 1 0 1	5	КТМ	НЕТ	%	5	E	U	e	u	НС	ОСУ						У
0 1 1 0	6	ДА	СИН	&	6	F	V	f	v	НП	ВП			ф	ж	Ф	Ж
0 1 1 1	7	ЗВ	КБ	'	7	G	W	g	w	ОЖД				г	в	Г	
1 0 0 0	8	ВШ	АН	(	8	H	X	h	x						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	[	k	{	СП2	СП3			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	Вых	РЗ	.	>	N	¬	n	¯	НРВ				н	ч		Ч
1 1 1 1	15	Вх	РЗ	/	?	O	_	o	ЗБ	СП1					ъ		ЗБ
Цифровая тетрада																	

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: $a_8a_7a_6a_5a_4a_3a_2a_1$. При этом значения цифр в позициях a_7, a_6, a_5, a_4, a_3, a_2, a_1 соответственно равны значениям $б_7$, $б_6$, $б_5$, $б_4$, $б_3$, $б_2$, $б_1$ 7-разрядного кода. Значение же цифры в позиции 8 зависит от того, каким таблицам КОИ-7 соответствует столбец таблицы КОИ-8: если КОИ-7$Н_1$, то в позиции a_8 будет ноль, если КОИ-7$Н_0$ и КОИ-7$С_1$, то – единица.

К преимуществам КОИ-8 по сравнению с КОИ-7 следует отнести наличие единой кодовой таблицы 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 0–3 таблиц КОИ-7$Н_0$ и КОИ-7$Н_1$.

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 показано в табл. 2.3.

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	0	1	1	1	1	1	1	1	1
1	0	0	0	0	1	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

Цифровая тетрада	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 0	ПУС	АР1	ВЦФ		Инв. вся	&	–		ц	й	я	ъ	{	}	\	0
0 0 0 1	НЗ	СУ1	КЗН				/		a	j	~	ы	A	J		1
0 0 1 0	НТ	СУ2	РП	СИН					b	k	s	э	B	K	S	2
0 0 1 1	КТ	СУ3							c	l	t	ш	C	L	T	3
0 1 0 0	ВЫП	ВСТ	БК	ВКП					d	m	u	э	D	M	U	4
0 1 0 1	ГТ	НС	ПС	ОСУ					e	n	v	щ	E	N	V	5
0 1 1 0	НП	ВШ	КБ	ВП				ю	f	o	w	ч	F	O	W	6
0 1 1 1	ЗБ	ОЖД	АР2	КП					g	p	x	ъ	G	P	X	7
1 0 0 0	8	АН						б	h	q	y	Ю	H	Q	Y	8
1 0 0 1	9	КН						`	i	r	z		I	R	Z	9
1 0 1 0	А	НРВ	УУК	УР		[	]	!	:	д	к		Б			З
1 0 1 1	В	ВТ	СП1	СП2	СП3	.	¤	,	#		л		Ц	И		У	Ш
1 1 0 0	С	ПФ	РФ		СТП	<	*	%	@	ф	м	т	Д	Й	П	Ж	Э
1 1 0 1	D	ВК	РГ	КТМ	НЕТ	(	)	_	'	г	н				Я		Щ
1 1 1 0	Е	Вых	РЗ	ДА		+	;	>	=			ж	ф	Л		ь	Ч
1 1 1 1	F	Вх	РЭ	ЗВ	ЗМ	!	¬	?	"	и	п	в	Г			ы	ЗБ

Кодовая таблица содержит полный алфавит, включающий кроме стандартных управляющих символов ряд дополнительных. Наименования графических знаков, стандартных и дополнительных управляющих символов даны в приложении (табл. 3–5).

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).

---

* ДКОИ соответствует универсальному международному двоично-кодированному десятичному коду для обмена информацией EBCDIC – Expanded Binary Coded Decimal Interchange Code.

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.

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 ¯, 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 соответствует в цифровой части байта двоично-десятичные коды от 0000 до 1001, то при размещении их в столбце 11 зоны 11 им будут соответствовать 8-разрядные двоичные коды: 11110000; 11110001; 11110010; ...; 11111001 (в шестнадцатиричной системе F0 – F9).

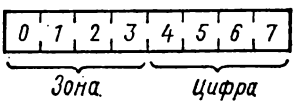


Рис. 2. 1. Деление кодирующего байта на зональную и цифровую части

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 ноль, а коду КОИ-8 – единица.*

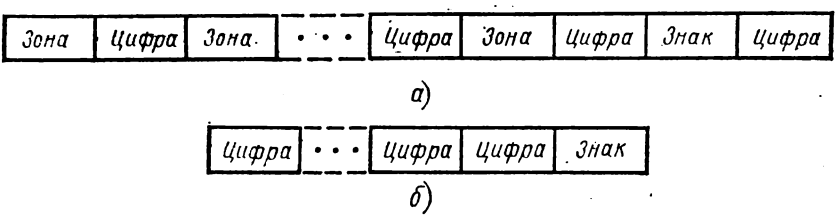


Рис. 2.2. Представление многоразрядных десятичных чисел: *а* – в распакованном; *б* – в упакованном форматах

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 (C и D), в КОИ-8 те же знаки изображаются кодами 1010 и 1011 (A и B). 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 *символьно-ориентированным* 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 части числа в зависимости от его значения 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 (рис. 2.3) фиксированные составные единицы — *полуслово, слово, двойное слово* и информационную единицу переменной длины — *поле*.

Полуслово состоит из двух байтов, слово — из четырех и двойное слово — из восьми (соответственно — 16, 32 и 64 двоичных разрядов); поле может быть

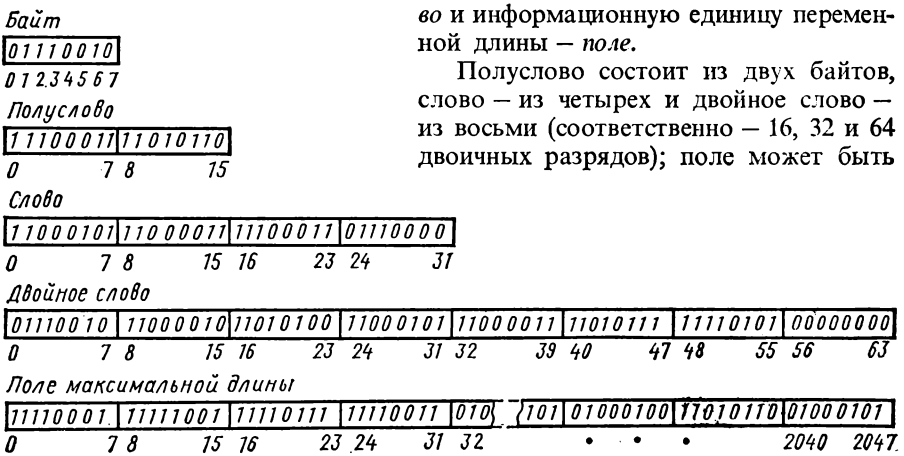


Рис. 2.3. Форматы данных

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 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 двум, слова – четырем и двойного слова – восьми. Иными словами, адрес (граница) составной единицы фиксированной длины всегда должен быть кратен количеству содержащихся в ней байтов. Такие границы получили название *целочисленных*.

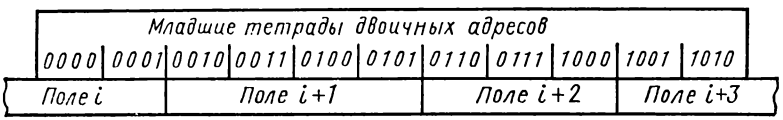


Рис. 2.4. Пример расположения полей данных в оперативной памяти

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. 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 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. В каждом блоке машины в зависимости от особенностей построения

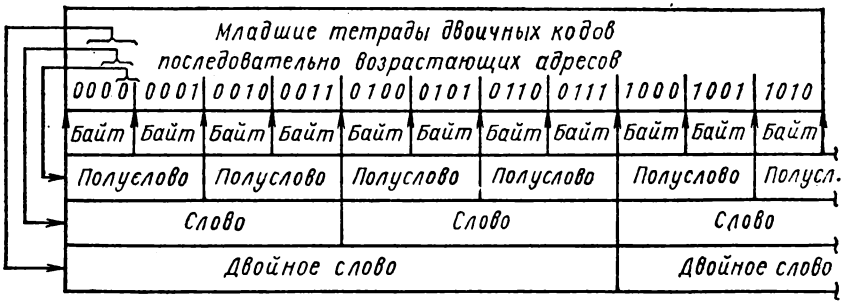


Рис. 2.5. Границы полуслов, слов и двойных слов при расположении их в оперативной памяти

конкретных электронных схем он вырабатывается из условия проверки 9-разрядных кодовых комбинаций на четность или нечетность (количество единиц в разрядах байта, включая контрольный, должно быть соответственно четным или нечетным). Количество контрольных разрядов, сопровождающих любой многобайтовый операнд или команду, равно количеству содержащихся в них байтов. Обнаружение ошибки при проверке информации на четность или 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. Количество разрядов может ограничиваться рядом фиксированных форматов (полуслово, слово и двойное слово), или быть переменным (выраженным определенным количеством байтов) 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 предшествующих выпусков, в ЭВМ Единой системы полулогарифмическая форма представления основывается на изображении чисел в виде

$$A = \pm p, \pm m, \tag{2.4}$$

соответствующем записи

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

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 заключается в том, что основание системы счисления 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 четыре разряда или на число разрядов, кратное этой величине, например, на восемь разрядов. Каждый сдвиг мантиссы на четыре разряда вправо или влево равносилен 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 0,00000001.

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

$$\frac{1}{N} \leqslant |m| < 1, \tag{2.6}$$

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. Код 0,11111...1 – машинная единица.

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

$$16^{-1} \cdot 16^{-(2^q-1)} \leqslant |A_{16}| \leqslant (1 - 16^{-\frac{l}{4}})\, 16^{2^q-1}, \tag{2.7}$$

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

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

что в десятичной системе эквивалентно

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

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

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

Полагая $1 - 2^{-l} \cong 1$, получим

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

или в десятичной системе

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

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 возникать 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 (см. табл. 2 приложения).

Нормализация числа, у которого в мантиссе одна старшая шестнадцатиричная цифра или более равна 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 четыре 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 полулогарифмической форме в ЕС ЭВМ заключается в том, что порядки чисел представляются увеличенными на 2^q. При шести разрядах, отводимых на представление порядка, т. е. при $q = 6$ (без знакового разряда), это составит 64. В результате отрицательные и положительные порядки p, увеличившись на 64 и превратившись в $p^* = p + 64$, оказываются смещенными на числовой оси в положительном направлении, как показано на рис. 2.6. Порядки диапазона

$$-2^q \leqslant p \leqslant 2^q - 1, \qquad (2.9)$$

т. е.

$$-64 \leqslant p \leqslant 63;$$

теперь ограничиваются пределами

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

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

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 сравнение. Для этого вычитание из одной характеристики другой можно производить без предварительного анализа их знаков.

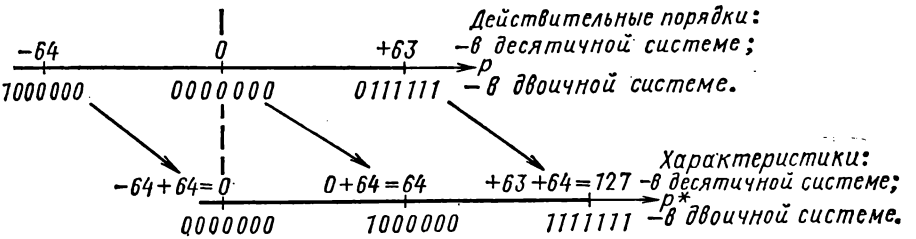


Рис. 2.6. Соотношения между величинами характеристик и действительными порядками

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, когда 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 др.

Для обеспечения возможности эффективной обработки информации указанного выше типа в *ЕС ЭВМ предусмотрены аппаратные и программные средства для выполнения действий над числами, представлен-*

ными 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 выравненными 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, предусмотренном командой для его размещения в оперативной памяти.

Диапазон представления целых двоичных чисел $A_2^{\text{ц}}$ в n-разрядных форматах данных ограничивается пределами

$$0 \leqslant |A_2^{\text{ц}}| \leqslant 2^n - 1. \qquad (2.11)$$

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

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,

$$0 \leqslant |A_{2-10}^{\text{ц}}| \leqslant 10^{\frac{n}{4}-1} - 1. \qquad (2.12)$$

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 чисел 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 доходить 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 фиксированной 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

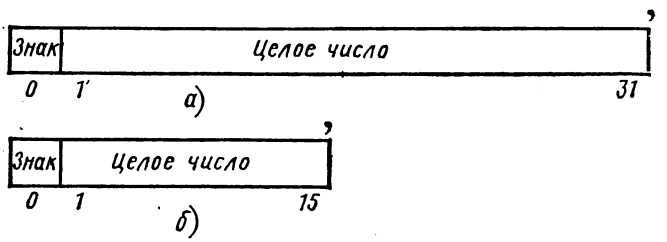


Рис. 2.7. Форматы двоичных чисел с фиксированной запятой: *а — слово; б — полуслово*

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

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 семь — для характеристики. Поле мантиссы располагается правее характеристики числа.

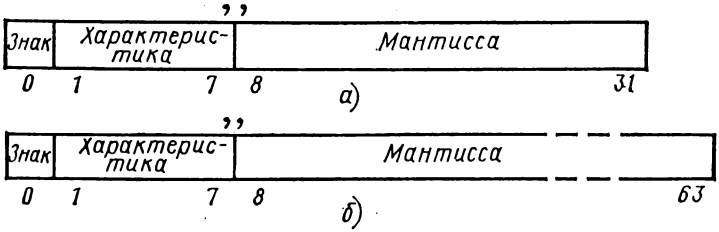


Рис. 2.8. Форматы двоичных чисел с плавающей запятой:
а — слово; б — двойное слово

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 семи 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' (см. рис. 2.6). Для получения характеристики истинный порядок как число со знаком складывается с числом 64. В результате коды характеристик, соответствующих отрицательным порядкам, во второй разрядной позиции формата слова или двойного слова имеют ноль, а для положительных порядков — единицу. 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 0100' '0111 1111 1111 0100 0000 0000.

Для шестнадцатиричного основания изображенная мантисса является нормализованной, характеристика $p^* = 4 + 2^6 = 68$. Весь код в шестнадцатиричной системе можно записать как C47FF400.

Число с нулевой характеристикой, нулевой мантиссой и положительным знаком называется истинным нулем. Он может появиться в результате арифметической операции при определенных величинах операндов. Результат операции может быть также принудительно получен 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 в других случаях 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 двоичные числа с фиксированной запятой хранятся в оперативной памяти и участвуют в операциях в дополнительном коде. При этом так как дополнительный код положительного числа совпадает с его прямым кодом, то положительные числа по сути дела оказываются представленными в прямом коде, а в виде дополнений – только отрицательные числа.

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

$$[A^{ц}]_{пр} = [n(0)] + A^{ц}, \tag{2.13}$$

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

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

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

$$[A^{ц}]_{доп} = 10^{n} + A^{ц}, \tag{2.14}$$

где n – также разрядность формата и 10 – два для двоичной системы счисления. Например, при $n = 8$ для $A^{ц} = -1011$

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

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 $A^{ц} = -1011$ в 8-разрядном формате

$$[A^{ц}]_{пр} = 1.000\,1011',$$

то, действуя согласно правилу, получим

$$[A^{ч}]_{доп} = \begin{array}{r} 1.111\,0100 \\ +\,1 \\ \hline 1.111\,0101 \end{array}.$$

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 $[0]_{пр} = 1.000...0$ в дополнительный код $[0]_{доп}$ 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 $[A^{ч}]_{пр} - [B^{ч}]_{пр}$ сложением с дополнением $[A^{ч}]_{пр} + [-B^{ч}]_{доп}$ 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: $A^{ц}_{10} = 69$ и $B^{ц}_{10} = -54$. Двоичные числа представляются в формате полуслова.

$$[A^{ц}_2]_{пр} = 0.000\ 0000\ 0100\ 0101$$
$$[B^{ц}_2]_{пр} = 1.000\ 0000\ 0011\ 0110$$
$$+\begin{matrix}[69]_{пр} \\ [-54]_{доп}\end{matrix}\begin{matrix}[A^{ц}_2]_{пр} = 0.000\ 0000\ 0100\ 0101 \\ [B^{ц}_2]_{доп} = 1.111\ 1111\ 1100\ 1010\end{matrix}\Big\}\oplus$$
$$\boxed{1}\boxed{1}111\ 1111\ 1000\ 000\leftarrow\ \text{Межразрядные переносы}$$
$$[15]_{пр}\quad [A^{ц}_2 + B^{ц}_2]_{пр} = 0.000\ 0000\ 0000\ 1111\quad \text{Сумма}$$

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: $A^{ц}_{10} = 54$ и $B^{ц}_{10} = -69$. Двоичные числа представляются в формате полуслова.

$$[A^{ц}_2]_{пр} = 0.000\ 0000\ 0011\ 0110$$
$$[P^{ц}_2]_{пр} = 1.000\ 0000\ 0100\ 0101$$
$$+\begin{matrix}[54]_{пр} \\ [-69]_{доп}\end{matrix}\begin{matrix}[A^{ц}_2]_{пр} = 0.000\ 0000\ 0011\ 0110 \\ [B^{ц}_2]_{доп} = 1.111\ 1111\ 1011\ 1011\end{matrix}\Big\}\oplus$$
$$\boxed{0}\boxed{0}000\ 0000\ 0111\ 110\leftarrow\ \text{Межразрядные переносы}$$
$$[-15]_{доп}\quad [A^{ц}_2 + B^{ц}_2]_{доп} = 1.111\ 1111\ 1111\ 0001\quad \text{Сумма}$$
$$[-15]_{пр}\quad [A^{ц}_2 + B^{ц}_2]_{пр} = 1.000\ 0000\ 0000\ 1111\quad \text{Результат в прямом коде}$$

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

В машинах отрицательные числа с фиксированной запятой в оперативной памяти хранятся в виде дополнений, поэтому перед выполнением операции алгебраического сложения их преобразования, 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 автома-

тически 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.

$$
\begin{array}{lll}
+\ [69]_{\text{пр}} & [A_2^{\text{ц}}]_{\text{пр}} = 0.100\ 0101 & \\
\underline{\ [69]_{\text{пр}}} & [A_2^{\text{ц}}]_{\text{пр}} = 0.100\ 0101 & \oplus \\
 & \boxed{0}\boxed{1}000\ 101\leftarrow & \text{Межразрядные переносы} \\
\hline
[138]_{\text{пр}} & [A_2^{\text{ц}} + B_2^{\text{ц}}]_{\text{пр}} = 1.000\ 1010 & \text{Сумма}
\end{array}
$$

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.

$$
\begin{array}{lll}
+\ [-69]_{\text{доп}} & [A_2^{\text{ц}}]_{\text{доп}} = 1.011\ 1011 & \\
\underline{\ [-69]_{\text{доп}}} & [A_2^{\text{ц}}]_{\text{доп}} = 1.011\ 1011 & \oplus \\
 & \boxed{1}\boxed{0}111\ 011\leftarrow & \text{Межразрядные переносы} \\
\hline
[-138]_{\text{доп}} & [A_2^{\text{ц}} + A_2^{\text{ц}}]_{\text{доп}} = 0.111\ 0110 & \text{Сумма}
\end{array}
$$

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.

Положительные и отрицательные *числа с плавающей запятой* (их характеристики и мантиссы) 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 порядков вычитаемая характеристика представляется в виде дополнения до 2^7, т. е.

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

В числах с плавающей запятой характеристики всегда положительны. При этом 7-разрядные характеристики отрицательных порядков p вычисляются по формуле

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

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

$$p_2^* = [1000\,0000 + (-000\,0010)] + 100\,0000 = 111\,1110 + 100\,0000 = 011\,1110.$$

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, если рассматривается формат числа с плавающей запятой).

Алгебраическое 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 всегда мантисса второго операнда (в ЕС-1022, ЕС-1033), либо мантисса операнда, имеющего больший исходный порядок [в ЕС-1050(52)].

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

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

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

Например, для $A = -0{,}10110111$ согласно (2.17)

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

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 операнд, мантисса которого поступила на сложение в дополнительном коде. Последнее имеет место, например, в процессоре ЕС-2050, где всегда инвертируется мантисса числа с бо́льшим порядком. Знак суммы, имеющей нулевую мантиссу, — всегда положительный.

Десятичные числа. Операции десятичной арифметики производятся над десятичными числами в упакованном формате (см. 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 арифметических операций они преобразуются в другой код, упрощающий сложение десятичных операндов в сумматорах. Например, в ЭВМ ЕС-1022, ЕС-1033 и ЕС-1050 (52) 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

$$\begin{array}{r} 0110\ \ 1101 \\ +\ 0001\ \ 1000 \\ \hline 1000\ \ 0101. \end{array}$$

$\uparrow$ — п — $\downarrow$

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

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

$\text{п} \leftarrow$ —

т. е. десятичное число 25.

При вычитании вычитаемое представляется в виде дополнения до 10^n, где n — число 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 и в рассмотренном выше примере, вычитание реализуется путем прибавления по $\bmod 2^4$ дополнения 1010.

Представление данных в форматах, разрядность которых кратна 8-разрядным байтам, и применение дополнительного кода при вычитании оказалось удобным для построения ряда машин различной производительности. Высокопроизводительные модели ЭВМ Единой системы [ЕС-1050(52) и ЕС-1060] для выполнения арифметических и логических операций имеют устройства в основном параллельного действия. Например, в ЕС-1050 (52) 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-1022, могут осуществлять только последовательно-параллельную переработку информации. В модели ЕС-1010 операции выполняются побайтно, а в ЕС-1022 данные могут обрабатываться как побайтно, так и по два байта. С целью экономии аппаратуры, например, при побайтовой обработке сложение n-разрядных слагаемых A и B выполняется в виде серии $\frac{n}{8}=k$ последовательных сложений байт A_i' и B_i' $(i=1, 2, \ldots, k)$, представленных параллельными кодами. Сложение начинается с младших (крайних правых) байтов. При сложении каждой пары байт вырабатывается и запоминается меж-

байтовый перенос, который учитывается при сложении следующей пары и т. д. В этих моделях поэтапно выполняются также и все остальные операции.

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 слагается из выполнения последовательности арифметических, логических и других операций. Тип каждой исполняемой операции, количество и местоположение операндов, указание, куда поместить результат и другая 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-1 имеет обширный набор, обеспечивающий выполнение более 140 видов операций. Она согласована с получившей широкое распространение системой команд машин серии 360 американской фирмы IBM. В табл. 2 приложения 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. табл. 2 приложения). 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. табл. 2 приложения). Адресная же часть может иметь различную длину: от 1 до 5 байт. Это объясняется особенностью иерархической структуры оперативной памяти машин.

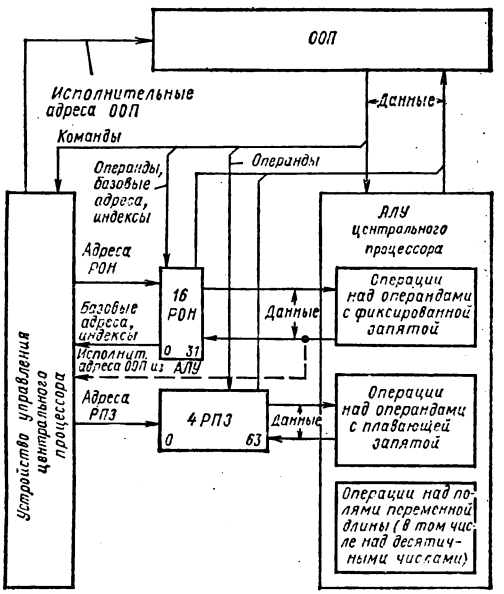


Рис. 2.9. Обобщенная структура информационных связей устройств центрального процессора с основной и регистровой оперативными памятями

Кроме основной оперативной памяти (ООП) большой емкости, во всех моделях ЭВМ Единой системы имеется еще доступная программистам 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). Регистровая память в новых моделях ЭВМ обычно реализуется на полупроводниковых интегральных запоминающих устройствах.

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 при малом времени обращения к полупроводниковой регистровой памяти получается выигрыш в быстродействии машин.

Регистры общего назначения – 32-разрядные. Они предназначаются 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.

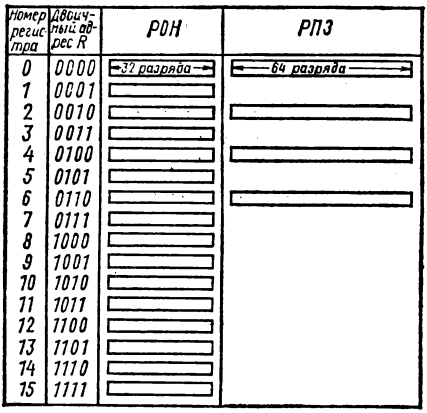


Рис. 2.10. Адресация регистров общего назначения и регистров для операндов с плавающей запятой

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 адресами *R*. Однако конкретный тип используемого 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 байтов, код адреса должен иметь $\log_2 16777216 = 24$ двоичных разряда. Помещать адреса такой разрядности в команды нерационально, так как это потребовало бы применения команд, имеющих чрезмерно большое количество разрядов. Программы, составляемые из таких многоразрядных команд, вызвали бы нерациональное расходование емкости оперативной памяти.

Для сокращения в командах 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 $A_{\text{исп}}$, 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 (см. рис. 2.9), что имеет место, например, в модели ЕС-1050(52), 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-1022 и ЕС-1033. На рис. 2.9 цепь получения исполнительных адресов из АЛУ показана пунктиром.

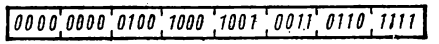


Рис. 2.11. Размещение базового адреса в регистре общего назначения

По количеству команд и объему данных программа может размещаться либо в пределах одной секции, либо занимать несколько. 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 4756335_{10}, которому соответствует шестнадцатиричное число $48936F_{16}$, 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 $A_{\text{исп}}$ автоматически вычисляется по формуле

$$A_{\text{исп}} = (B) + D, \qquad 2.18)$$

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

$$4756335_{10} = 48936F_{16} = 100\,1000\,1001\,0011\,0110\,1111_2,$$

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

$$\begin{array}{r} 100\ 1000\ 1001\ 0011\ 0110\ 1111 \\ + \qquad\qquad\qquad\quad 11\ 1111\ 1101 \\ \hline 100\ 1000\ 1001\ 0111\ 0110\ 1100 \end{array}$$

что соответствует $48976C_{16}$, или 4757356_{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 осуществляются автоматически управляющей программой операционной системы.

При выполнении операций над операндами фиксированной длины количество байтов, 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 байтов, 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 длины операнда в адресной части команды.

Использование рассмотренных способов адресации, связанное с 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 адреса регистров 1-го и 2-го операндов. 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

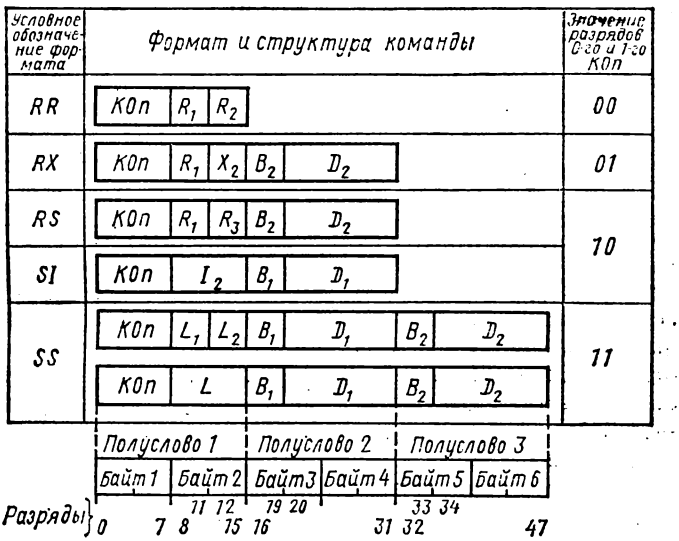
смещения D_2 с содержимым регистров базы и индекса, имеющими 4-разрядные адреса соответственно B_2 и X_2, т. е.

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

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.

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



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 кода 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 три 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 16 байт каждый.

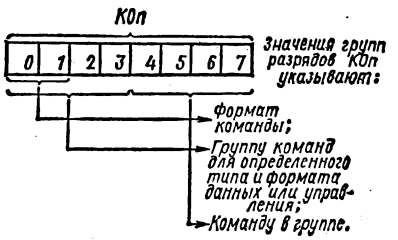


Рис. 2.13. Назначение составляющих частей кода операции

В командах логических операций над полями переменной длины оба поля берутся равными. Поэтому в таких командах используется один указатель длины L. Число байтов в поле равно $L+1$. При восьми разрядах, отводимых на код длины, $L_{\max} \leqslant 255$. Так как L указывает количество байтов в дополнение к каждому из адресуемых $A_{1\,\text{исп}} = (B_1) + D_1$ и $A_{2\,\text{исп}} = (B_2) + D_2$, то максимально возможная длина каждого поля составляет 256 байт.

В рассмотренных выше типах форматов коды X_t, B_t и D_t 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 (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. табл. 2 приложения). Значения следующих двух разрядов (2-го и 3-го) совместно с двумя первыми (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 проблем 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.

Кратность форматов команд полуслову и указание их длины в кодах операций дают возможность любой модели воспринимать команды независимо от того, сколько байтов за одно обращение позволяет выбирать ее оперативная память. Указатель формата, например, в ЕС-1020 позволяет выбирать команды из ООП и обрабатывать их по частям при ширине выборки 2 байт, в моделях ЕС-1022 и ЕС-1033 – при ширине выборки 4 байт, а в ЕС-1050 (52) – при ширине выборки 8 байт.

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.

Глава 3

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 в 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 структур кодов команд, адресов и данных. Организация такой проверки позволяет выявлять ошибки программ.

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 байтов, одновременно выбираемых из оперативной памяти 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-1022 и Е4-1033 за одно обращение к оперативной памяти выбирается или записывается 4 байта, в ЕС-1050 (52) — 8 байтов, а в ЕС-1045 — 16 байтов, является причиной ряда особенностей выполнения операций в каждой из перечисленных моделей. Например, в ЕС-1022 6-байтные коды команд приходится выбирать соответственно за два

обращения к памяти. Аналогично и операнды, длина которых превышает слово, представляется возможным выбирать в процессор по частям. В противоположность этому в ЕС-1050 (52) 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 очередной команды.

АЛУ в модели ЕС-1022 — 16-разрядное, в модели ЕС-1033 — 32-разрядное, а в ЕС-1050 (52) — 64-разрядное. В первых двух типах машин при выполнении арифметических и логических операций над двоичными данными фиксированной длины и над десятичными числами действия одновременно производятся соответственно над парами или четверками байтов. В логических же операциях над полями переменной длины данные обрабатываются побайтно. В АЛУ модели ЕС-1050 (52) арифметические и логические операции над данными фиксированной длины могут выполняться параллельно над любыми операндами, вплоть до 64-разрядных. Кроме того, имеется специальный блок для арифметических действий над десятичными числами и для выполнения логических операций над полями переменной длины. Он может осуществлять только побайтную 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 рассмотрены лишь способы выполнения арифметических и логических действий, положенные в основу машинных алгоритмов операций различных моделей. При этом основное внимание уделено вопросам упрощения побайтного преобразования информации, достижения высокой скорости ее переработки, а также принципиальным особенностям реализуемых в моделях машинных алгоритмов.

3.2. Арифметические операции над двоичными числами с фиксированной запятой

В ЕС ЭВМ арифметические 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 виде дополнений (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 команды как формата RR, так и RX.

Команды формата RX имеются в двух вариантах. Один вариант предназначен для операций, при выполнении которых выбираемый из ос-

новной памяти операнд является полным словом, а другой – 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 действия 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 вызываются из регистровой или основной оперативной памяти во входные регистры АЛУ процессора. Операции выполняются над числами, выровненными по младшим разрядам, правее которых считается фиксированной запятая.

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.

В зависимости от конструкции АЛУ команда ВЫЧЕСТЬ может выполняться либо путем непосредственного вычитания из кода уменьшаемого кода вычитаемого, либо путем сведения операции вычитания к суммированию кодов. Первое имеет место при использовании универсальных арифметико-логических блоков, которые могут работать как сумматоры и как вычитатели. Второй способ – сведение операции вычитания к алгебраическому сложению – применяется тогда, когда основу АЛУ составляет сумматор. В этом случае вычитаемое в сумматор подается с обратным знаком, т. е. в виде дополнения. Для этого оно поразрядно инвертируется и к младшему разряду прибавляется единица.

Таким образом, как при сложении, так и при вычитании не требуется предварительного анализа знаков исходных операндов, поступающих в АЛУ из запоминающих устройств, где они хранятся в дополнительном коде.

Для выполнения, например, сложения кодов арифметико-логическое устройство коммутируется на режим суммирования. По окончании процесса суммирования производится анализ результата и выработка его признака. В анализ входит определение: равен, больше или меньше нуля результат и не возникло ли переполнения. 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.

1) $+[+5]_{пр}$ 0.000 0101
$[-5]_{доп}$ 1.111 1011 $\oplus$
1.111 1110 — поразрядная сумма слагаемых по mod 2
$\oplus$ 111 111 — переносы
$[0]_{пр}$ 0.000 0000 — результат, получившийся равным нулю; ПР-00

2) $+[+5]_{пр}$ 0.000 0101
$[-125]_{доп}$ 1.000 0011 $\oplus$
Примечание: $+125_{10}=0111\ 1101_2$
1.000 0110 — поразрядная сумма слагаемых по mod 2
$\oplus$ 00 000 111 — переносы
$[-120]_{доп}$ 1.000 1000 — отрицательный результат в дополнительном коде; ПР-01

3) $+[+125]_{пр}$ 0.111 1101
$[-5]_{доп}$ 1.111 1011 $\oplus$
1.000 0110 — поразрядная сумма слагаемых по mod 2
$\oplus$ 11 111 111 — переносы
$[+120]_{пр}$ 0.111 1000 — положительный результат в прямом коде; ПР-10

4) $+[+5]_{пр}$ 0.000 0101
$[+125]_{пр}$ 0.111 1101 $\oplus$
0.111 1000 - поразрядная сумма слагаемых по mod 2
$\oplus$ 01 111 101 — переносы
$[+136]_{пр}$ 1.000 0010 результат; ПР-11. Возникло переполнение, так как перенос в знаковый разряд есть, а из знакового — нет!

5) $+\begin{matrix}[-5]_{доп}\\ [-125]_{доп}\end{matrix}$ $\begin{matrix}1.111\ 1011\\ 1.000\ 0011\end{matrix}\Big\}\oplus$

$\oplus\Big\{$ 0.111 1000 — поразрядная сумма слагаемых по mod 2

1 0 000 011 — переносы

$[-130]_{доп}$ 0.111 1110 — 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 ВЫЧЕСТЬ

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

0.111 1101 − 0.000 0101 = $\oplus\Big\{$ 0.111 1101 + 1.111 1011

1.111 1011 ←

$\oplus\Big\{$ 1.000 0110 — поразрядная сумма по mod 2

1 1 111 111 — переносы

0.111 1000 — результат; ПР-10

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

0.111 1101 − 1.111 1011 = $\oplus\Big\{$ 0.111 1101 + 0.000 0101

0.000 0101 ←

$\oplus\Big\{$ 0.111 1000 — поразрядная сумма по mod 2

0 1 111 101 — переносы

1.000 0010 результат; ПР-11. Переполнение!

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

В различных моделях ЭВМ Единой системы коды операндов поразному подаются на сложение. В ЕС-1022 схема суммирования 16-разрядная, поэтому 32-разрядные операнды складываются в два этапа, последовательно-параллельно, начиная с младших (правых) пар байтов. Межбайтный перенос, возникший при сложении двух младших пар байтов, запоминается и учитывается при сложении следующих (старших) пар байтов. В ЕС-1033 и ЕС-1050 (52), имеющих соответственно 32- и 64-разрядные суммирующие схемы, 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 раз-

ряды 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 переполнение 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-1033 и ЕС-1050(52) реализуется способ умножения без преобразования отрицательных сомножителей в прямой код. В модели ЕС-1022 множимое всегда участвует в умножении как положительное число в прямом коде. При этом во всех моделях применяются различные приемы, способствующие ускорению процессов умножения. В моделях ЕС-1033 и ЕС-1050(52) знак произведения вырабатывается автоматически в ходе перемножения сомножителей, и произведение получается в коде, соответствующем его знаку. В модели ЕС-1022 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 обусловливаются количествами одновременно анализируемых разрядов множителя, способами накопления сумм частных произведений и в связи с этими 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

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

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 очеред-

ной 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 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: 0111	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 $[A]_{пр}$, а отрицательный множитель — в дополнительном коде $[B]_{доп} = 2^n - |B|$, где n — разрядность формата множителя. Для большей наглядности в выражении дополнительного кода использовано абсолютное значение B.

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

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

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

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

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

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

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

		Множимое: 0.111	Множитель: 1.011
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	
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 $[-A]_{доп}2^4 = (2^4 - A)2^4 = 1001\ 0000_2$. Полученный результат есть дополнительный код $[AB]_{доп} = 1.101\ 1101$, равный десятичному числу -35.

3. При непосредственном умножении отрицательного множимого $A < 0$, представленного дополнительным кодом $[A]_{доп}$, на положительный множитель $B > 0$, представленный прямым кодом $[B]_{пр}$, если не корректировать результат, также получается псевдопроизведение

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

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

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

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 $[-B]_{доп}$, 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 переполнения. 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 $[-B]_{доп}$. 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 $[A]_{доп} = 1.001$ и $[B]_{пр} = 0.101$. Требуется найти их произведение.

	Множимое: 1.001	Множитель: 0.101
Исходная сумма частных произведений	0 000 0000	↓
1-е частное произведение	+ 1001	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	+ 1001	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 $[AB]_{доп} = 1.101\,1101'$, равное десятичному числу -35. В примере звездочками отмечены случаи коррекции путем модифицированных сдвигов, соответствующие дополнительному коду поправочного члена в (3.5): $[-B]_{доп} \times 2^4 = (2^4 - B)2^4 = 1011\,0000_2$.

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

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

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

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

Подставляя их в правую часть (3.6), получим

$$[C]_{пр} = 2^{2n} + |A||B|. \tag{3.7}$$

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

$$[C]_{пр} = [A]_{доп}[B]_{доп} + |A|2^n + |B|2^n. \tag{3.8}$$

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 $[A]_{доп} = 1.001$ и $[B]_{доп} = 1.011$. Требуется найти их произведение.

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

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-1033 применен способ одновременного умножения на два разряда множителя, а в ЕС-1050(52) – на четыре. В ЕС-1022 также одновременно анализируются четыре разряда множителя.

Суть умножения с одновременной расшифровкой двух разрядов множителя заключается в следующем. В каждом цикле умножения анализируется пара разрядов множителя, начиная с младшей. В результате анализа определяется одно частное произведение и способ его подсуммирования к уже накопленной сумме. Полученная очередная сумма затем сдвигается на два разряда вправо.

Очередная пара разрядов может 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 множимого, как при умножении на очередную единицу множителя в рассмотренных выше примерах, но с последующим двойным сдвигом вправо. Умножение на 10 требует удвоения множимого перед суммированием. Чтобы не делать цепей передачи множимого в сумматор со сдвигом на один разряд влево, что равносильно его удвоению, при расшифровывании комбинации 10 поступают иначе. Вначале на один разряд вправо сдвигают накопленную ранее сумму частных произведений, затем в сумматор подают 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. Это выполняется путем вычитания множимого из накопленной суммы частных произведений с последующим сдвигом уменьшенной таким образом суммы на два разряда вправо. 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 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 варианта находится в нулевом состоянии, или 10 либо 11 и триггер варианта – 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 варианта 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

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

а в примере 3.6

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

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 вычитанием 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.

Такая коррекция получается автоматически, если при умножении с расшифровкой пар разрядов множителя очередное частное произведение и характер действия над ним определять с учетом значения старшего разряда пары, на которую производилось умножение в предыдущем цикле. Этот способ умножения реализован в ЭВМ ЕС-1033. В данной модели при анализе каждой очередной пары разрядов, начиная с младшей, учитывается значение старшего разряда предыдущей пары. Для самой младшей пары таким значением всегда будет 0. В зависимости от значений трех очередных анализируемых разрядов будет выполняться одно из действий, приведенных в табл. 3.1.

Таблица 3.1

Действия при умножении на очередную пару разрядов с учетом значения старшего разряда предыдущей пары

i-я пара	Старший разряд $i-1$-й пары	Действие над предыдущей ($i-1$-й) суммой частных произведений
00	0	Сдвиг на два разряда вправо
00	1	Прибавление множимого и сдвиг на два разряда вправо
01	0	Прибавление множимого и сдвиг на два разряда вправо
01	1	Сдвиг на один разряд вправо, прибавление множимого и еще сдвиг на разряд вправо
10	0	Сдвиг на один разряд вправо, вычитание множимого и снова сдвиг на один разряд вправо
10	1	Вычитание множимого и сдвиг разности на два разряда вправо
11	0	Вычитание множимого и сдвиг разности на два разряда вправо
11	1	Сдвиг на два разряда вправо

Как следует из анализа табл. 3.1, в циклах умножения, в которых анализируются комбинации 100, 101 и 110, возникают недостатки в очередных суммах частных произведений, а при анализе комбинаций 001, 011 и 111 — избытки. При действиях с многоразрядными числами эти недостатки и избытки взаимокомпенсируются. Например, в случае умножения некоторого числа A на множитель 0010 (+2) процесс умножения согласно табл. 3.1 можно представить как $-2A + 4A = 2A$.

Действие при расшифровке старшей пары разрядов, содержащей знак множителя, когда анализируется комбинация 100, 101, 110 или 111, приводит к автоматической коррекции результата. Например, при комбинации 100 алгебраическое вычитание множимого из предварительно сдвинутой на один разряд вправо суммы частных произведений с последующим сдвигом разности на разряд вправо полностью соответствует рассмотренному выше требуемому действию коррекции при умножении на 10 в старших разрядах множителя.

Учет корректирующих членов $[-B]_{доп}2^n$ и $|B|2^n$ согласно выражениям (3.5) и (3.8) производится в ходе умножения путем модифицированных сдвигов сумм частных произведений.

Пример 3.7. Рассмотрим особенности действий при различных комбинациях цифр в анализируемых парах разрядов с учетом значения старшего разряда предыдущей пары. Производимые действия определяются по табл. 3.1.

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

Множимое:		1001	Множитель: 01110
Исходная сумма частных произведений	+	0000 0000	
Вычитание		0111	110
		0111 0000	
Сдвиг на два разряда вправо		0001 1100	
Сдвиг на один разряд вправо		0000 1110	011
Прибавление	+	1001	
		1001 1110	
Сдвиг отрицательной суммы на один разряд вправо		1100 1111 (−49)	

Результат получился в дополнительном коде.

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

Множимое:		1001	Множитель: 01010
Исходная сумма частных произведений	+	0000 0000	
Прибавление		1001	010
		1001 0000	
Сдвиг отрицательной суммы на два разряда вправо		1110 0100	
Прибавление	+	1001	010
		1←0111 0100	
Сдвиг отрицательной суммы на два разряда вправо		1101 1101 (−35)	

Результат – в дополнительном коде.

в) $A < 0$; $B > 0$. $[A]_{доп} = 1.001$ (-7) и $[B]_{доп} = 0.101$ $(+5)$

Множимое: 1001 Множитель: 01110

Исходная сумма частных произведений	$+\begin{matrix}0000\,0000\\0111\end{matrix}$	110
Вычитание	0111 0000	
Сдвиг на два разряда вправо	0001 1100	
Сдвиг на один разряд вправо	$+\begin{matrix}0000\,1110\\1001\end{matrix}$	011
Прибавление	1001 1110	
Сдвиг отрицательной суммы на один разряд вправо	1100 1111 (−49)	

Результат – в дополнительном коде.

г) $A < 0$; $B < 0$. $[A]_{доп} = 1.001$ (-7) и $[B]_{доп} = 1.011$ (-5).

Множимое: 1001 Множитель: 10110

Исходная сумма частных произведений	$+\begin{matrix}0000\,0000\\0111\end{matrix}$	110
Вычитание	0111 0000	
Сдвиг положительной суммы на два разряда вправо	0001 1100	
Вычитание	$+\begin{matrix}0111\end{matrix}$	101
	1000 1100	
Сдвиг положительной суммы на два разряда вправо	0010 0011 (+35)	

Результат – в прямом коде.

В модели ЕС-1050(52) 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 произведения 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 расшифровке 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. 3.2.

Таблица 3.2

Частные произведения при умножении одновременно на четыре разряда

Комбинация цифр в $i+1$-й тетраде множителя	Пара частных произведений, прибавляемая к i-й сумме частных произведений	Комбинация цифр в $i+1$-й тетраде множителя	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	$0A - 1A$

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

Множимое, сдвинутое на четыре разряда вправо: 0000 0111 Множитель: 0111 1001

Исходная сумма частных произведений: 0000 0000 0000 0000

Первая пара частных произведений: + 0000 0111 / 0011 1000

0011 1111 0000 0000

Сдвинутая 1-я сумма частных произведений: 0000 0011 1111 0000

Вторая пара частных произведений: + 1111 1001 / 0011 1000

Результат в прямом коде: 0011 0100 1111 0000 (+13552)

01

10

11 = 100 − 1

01 + 01 = 10

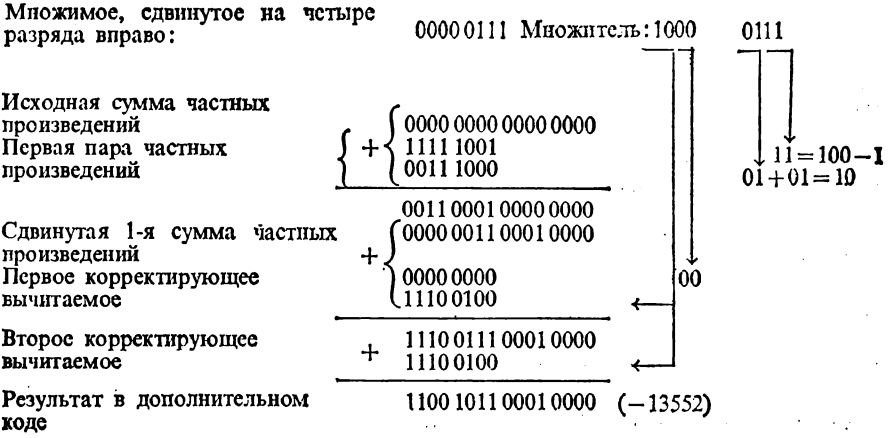
После предварительного сдвига 32-разрядного множимого все его разряды продолжают участвовать в накоплении суммы частных произведений с помощью 64-разрядного сумматора. При умножении операндов, представленных в дополнительных кодах, приходится выполнять необходимые коррекции. В случаях отрицательного множимого имеют место модифицированные сдвиги. Коррекция на величину множимого согласно (3.3) и (3.8) при отрицательном множителе вводится только при наличии в старшей тетраде последнего 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 $-A2^{31}$ указывает на то, что умножение на знаковый разряд множителя не произведено. Умножение не выполняется, а к псевдопроизведению добавляется величина $2^{64} - A2^{31}$. В итоге получается точный результат

$$A2^{32} - A|B| - A2^{31} + 2^{64} - A2^{31} = 2^{64} - A|B|. \qquad (3.11)$$

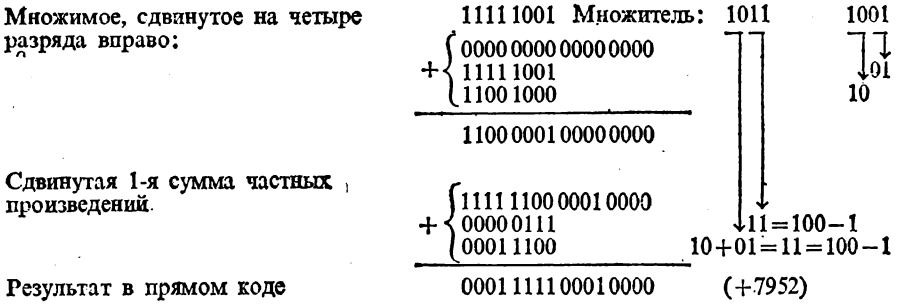
Корректирующее слагаемое вводится путем сдвига на два разряда влево кода множимого, сдвинутого в начале умножения. В результате предварительного сдвига в умножении участвует множимое $(A2^{32})2^{-4} = A2^{28}$. Сдвигая его при коррекции на два разряда влево, получим $(A2^{28})2^2 = A2^{30}$. Последующее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



При комбинации в старшей тетраде множителя 1100, 1101, 1110 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.



В модели ЕС-1022 каждой тетраде множителя ставится в соответствие подсуммирование к текущей сумме частных произведений не более двух частных произведений, как показано в табл. 3.3.

Таблица 3.3

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	$-(6A + 2A)$	$-(6A + 1A)$
0001	$1A$	$2A$	1001	$-(6A + 1A)$	$-6A$
0010	$2A$	$3A$	1010	$-6A$	$-(6A - 1A)$
0011	$3A$	$6A - 2A$	1011	$-(6A - 1A)$	$-(6A - 2A)$
0100	$6A - 2A$	$6A - 1A$	1100	$-(6A - 2A)$	$-3A$
0101	$6A - 2A$	$6A$	1101	$-3A$	$-2A$
0110	$6A$	$6A + 1A$	1110	$-2A$	$-1A$
0111	$6A + 1A$	$6A + 2A$	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 локальной памяти (ЛП). Если множимое отрицательно, то предварительно для его кода берется дополнение и фактически по положительному числу определяются указанные кратные величины. Множитель участвует в операции также в прямом коде. Если он отрицателен, то для его кода также формируется дополне-

ние. Знак произведений определяется согласно алгебраическому правилу о знаках сомножителей и произведения.

Тетрады множителя анализируются последовательно, начиная со старшей. Для определения характера действий на очередном этапе умножения учитывается значение следующей (младшей) тетрады. Для самой младшей тетрады следующая считается равной нулю.

По результату анализа определяется целесообразный вариант суммирования кратных множимого. Перечисленные в табл. 3.3 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 влево.

При умножении на самую старшую тетраду множителя знаковый разряд считается разрядом данных. Так как множитель участвует в умножении положительным, то в знаковом разряде всегда — нуль.

Если сомножители оба положительные или отрицательные, то автоматически получается положительное произведение. Если сомножители разных знаков, то для полученного произведения формируется дополнение, т. е. определяется дополнительный код.

Пример 3.11. Перемножим по рассмотренному алгоритму следующие числа: множимое $[A]_{пр} = 0.000\,0111$ (+7) и множитель $[B]_{доп} = 1.111\,0101$ (−11).

Кратные множимого

$$\begin{cases} 0000\,0111 & -1A \\ 0000\,1110 & -2A \\ 0001\,0101 & -3A \\ 0010\,1010 & -6A \end{cases}$$

Дополнение множителя

0000 1011

Умножение на старшую тетраду

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

1A

Сдвиг на четыре разряда влево
Умножение на младшую тетраду

$$\begin{array}{r} 0000\,0111\,0000 \\ +\ 1111\,1101\,0110 \\ \hline 0000\,0100\,0110 \\ +\ 0000\,0000\,0111 \\ \hline 0000\,0100\,1101 \end{array}$$

$-6A$

$+1A$

Абсолютная величина результата

Так как знаки $0 \oplus 1 = 1$, то $2n$-разрядное отрицательное произведение в дополнительном коде: 1111 1111 1011 0011 (−77).

При умножении на младшую тетраду промежуточная сумма частных произведений участвует в сложении со сдвигом на четыре разряда

влево («с перекосом»). Так как множитель отрицателен, то для конечной суммы частных произведений определяется дополнение.

Если множимое или множитель — минимально возможное отрицательное число, дополнение от которого есть такое же число, то умножение производится по другому алгоритму. Этот более простой алгоритм, основанный на сдвиге соответственно множимого или множителя влево на 31 разряд, дает возможность искусственным приемом скомпоновать 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 по команде как 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 знаком имеют 32-разрядные форматы. В модели ЕС-1050 отрицательные исходные операнды участвуют в делении в дополнительном коде. Поэтому знак частного в них определяется автоматически в процессе деления кода делимого на код делителя. Отрицательные остаток и частное также получаются в дополнительном коде. В моделях ЕС-1033 и ЕС-1022 участвуют 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 $\left|(2^{31}-1)2^{31}\right| = 2^{62}-2^{31}$. 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

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

в случае одинаковых знаков делимого D и делителя d или

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

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 по условиям либо (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.

Процесс собственно деления слагается 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 один 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.

$$2R_k + |d| = 2(R_k + |d|) - |d|. \tag{3.14}$$

В моделях ЕС-1022 и ЕС-1033 делятся абсолютные значения делимого и делителя. Для вычитания из абсолютного значения делимого и положительных текущих остатков используется двоичное дополнение кода делителя. При таком способе деления цифры частного определяются по следующему правилу: *если очередная разность положительна, то в соответствующий разряд частного заносится единица, если отрицательна, то — нуль.* 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

* В модели ЕС-1050 удобнее оказалось реализовать анализ по разностям $2^{31}d - D$ и $2^{31}d + d - D$. В моделях ЕС-1033 и ЕС-1022, так как участвуют абсолютные значения делимого и делителя, проверка корректности всегда осуществляется с помощью разности $D - 2^{31}d$.

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 $[D]_{пр} = 0.001.\ 0001'$ $(+17)$ и делитель $[d]_{пр} = 0.100'$ $(+4)$. Требуется выполнить деление рассмотренным выше методом.

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

$$\begin{array}{r} 0.001\,0001 \\ +\ 1.1100 \\ \hline 1.111\,0001 \end{array}$$

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 — незначащая цифра, поэтому сдвигается делимое на разряд влево и начинается деление[1].

Сдвинутое на разряд влево делимое:	0010	0 0 1 0	Делитель: 0.100.
Вычитание делителя	+ 1100		
Отрицательный остаток. Деление возможно!	1110	0 0 1 [0]	
Сдвиг первого остатка	1100	0 1 [0] 0	
Добавление делителя	+ 0100		
	0000	0 1 [0 1]	
Сдвиг второго остатка	0000	1 [0 1] 0	
Вычитание делителя	+ 1100		
	1100	1 [0 1 0]	
Сдвиг третьего остатка	1001	[0 1 0] 0	
Добавление делителя	+ 0100		
Коррекция 4-го остатка путем восстановления	{ + 1101 0100	[0 1 0 0]	— остаток должен быть положительный!
Результат	0001	[0 1 0 0]	
	Остаток	Частное	

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

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

При положительных операндах в коррекции частного нет необходимости.

В некоторых моделях деление осуществляется таким образом, что операнды участвуют в делении вместе со знаками, и все отрицательные числа в АЛУ представляются в дополнительном коде. В связи с этим, поскольку исходный делитель может быть 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 правилу: *если знак очередного остатка совпадает со знаком делителя, то соответствующий шагу деления разряд частного равен единице, в противном случае — нулю.* 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 и 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 обусловливается 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, к младшему разряду частного прибавляется единица.

Деление положительных чисел согласно последнему алгоритму не отличается от деления 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:	$+\begin{matrix}0010\\1100\end{matrix}$	0 0 1 0	Делитель: 1.100
	1110	0 0 1 [1]	
Сдвиг 1-го остатка	$+\begin{matrix}1100\\0100\end{matrix}$	0 1 [1] 0	
	0000	0 1 [1 0]	
Сдвиг 2-го остатка	$+\begin{matrix}0000\\1100\end{matrix}$	1 [1 0] 0	
	1100	1 [1 0 1]	
3-го остатка	$+\begin{matrix}1001\\0100\end{matrix}$	[1 0 1] 0	
Коррекция остатка и частного	$+\begin{matrix}1101\\0100\end{matrix}$	[1 0 1 1] +1	
Результат	0001	[1 1 0 0]	
	Остаток	Частное	

В данном случае частное $[p]_{доп} = 1.100\ (-4)$ и остаток $[r]_{пр} = 0.001\ (+1)$. 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 $[D]_{доп} = 1.110\ 1111\ (-17)$ на $[d]_{пр} = 0.100\ (+4)$.

Сдвинутое на разряд влево делимое:	$+\frac{1101}{0100}$	1 1 1 0	Делитель: 0.100
	0001	1 1 1 **1**	
Сдвиг 1-го остатка	$+\frac{0011}{1100}$	1 1 **1** 0	
	1111	1 1 **1 0**	
Сдвиг 2-го остатка	$+\frac{1111}{0100}$	1 **1 0** 0	
	0011	1 **1 0 1**	
Сдвиг 3-го остатка	$+\frac{0111}{1100}$	**1 0 1**	
Коррекция остатка и частного	$+\frac{0011}{1100}$	**1 0 1 1** +1	
Результат	1111	**1 1 0 0**	
	Остаток	Частное	

Частное и остаток получились в дополнительном коде $[p]_{доп} = 1.100\ (-4)$ и $[r]_{доп} = 1.111\ (-1)$. 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 $[D]_{доп} = 1.110\ 1111\ (-17)$ на $[d]_{доп} = 1.100\ (-4)$.

Сдвинутое на разряд влево делимое:	$\frac{1101}{0100}$	1 1 1 0	Делитель: 0.100
	0001	1 1 1 **0**	
Сдвиг 1-го остатка	$+\frac{0011}{1100}$	1 1 **0** 0	
	1111	1 1 **0 1**	
Сдвиг 2-го остатка	$+\frac{1111}{0100}$	1 **0 1** 0	
	0011	1 **0 1 0**	

Сдвиг 3-го остатка $+\begin{matrix}0111\\1100\end{matrix}$ $\boxed{0\ 1\ 0}\ 0$

Коррекция окончательного остатка $+\begin{matrix}0011\\1100\end{matrix}$ $\boxed{0\ 1\ 0\ 0}$

1111 $\boxed{0\ 1\ 0\ 0}$ $-1;\ +4$

Остаток Частное

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.

Преобразование десятичных чисел в двоичные. По команде ПРЕОБРАЗОВАТЬ В ДВОИЧНУЮ, имеющей формат RX, 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 должны удовлетворять требованиям соответствующей границы, т. е. 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 *R* 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-1022 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 3.16. Перевести десятичное число 95 (1001 0101) в двоичное.

	$\times \begin{matrix}1001 & 0101\\ 1010 & \smile\!\!\!\!\!\!\smile\end{matrix}$
1-е слагаемое	001 0010
2-е слагаемое	100 1000
	$+\begin{matrix}101\,1010\\ 0101\end{matrix}\leftarrow$
Следующая цифра	
Результат	101 1111

В случае отрицательного десятичного числа результат – двоичное число – в конце операции переводится в дополнительный код.

Если преобразование осуществляется специальными блоками, выполняемыми на интегральных схемах – преобразователях двоично-десятичных чисел в двоичные, то реализуется алгоритм перевода целых чисел, иллюстрируемый примером 2.1. При этом деление на два заменяется сдвигом двоично-десятичного кода на одну двоичную позицию вправо. Выдвигаемые из младшей тетрады двоичные цифры образуют код получаемого двоичного числа. Первая выдвинутая двоичная цифра – младший разряд двоичного результата.

Для того чтобы путем сдвига вправо получить правильные результаты деления на два, необходимо после каждого сдвига анализировать коды вновь образующихся тетрад. Из тетрад, по величине превышающих семь, следует вычитать избытки, равные трем. Сдвиги осуществляются до тех пор, пока в младшей тетраде не будет содержаться число, меньшее восьми, а все остальные тетрады не станут равными нулю. Вычитание избытков 3 (0011_2) осуществляется путем прибавления их дополнений до 2^4, т. е. 1101, без учета межтетрадных переносов. Такой алгоритм реализуется, например, в модели ЕС-1033.

Пример 3.17. Перевести десятичное число 95 (1001 0101) в двоичное.

Исходное число	1001	0101	0000	0000
Сдвиг 1-й	0100	$+$ 1010	1000	0000
Коррекция		1101		
	0100	0111	1000	0000
Сдвиг 2-й	0010	0011	1100	0000
Сдвиг 3-й	0001	0001	1110	0000
Сдвиг 4-й	0000	$+$ 1000	1111	0000
Коррекция		1101		
	0000	0101	1111	0000
Результат		0101	1111	

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. 22). 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 либо путем последовательного деления его на 10, как показано, например, в примере 2.2, либо путем последовательного процесса умножений и сложений согласно формуле (2.2), как показано в примере 2.3. В данном случае $N = 2$. Второй способ применяется в ЭВМ ЕС-1022. В этой модели умножение на два выполняется как сложение числа с самим собой. В процессе перевода все двоично-десятичные числа складываются как десятичные, с представлением в коде с избытком 6 первого операнда. Как положительные, так и отрицательные числа переводятся в десятичные в прямом коде.

Поэтому для отрицательных операндов предварительно определяется дополнение.

В модели ЕС-1033 блок перевода в десятичную систему счисления строится на интегральных схемах — преобразователях двоичных чисел в двоично-десятичные. В этом блоке перевод также осуществляется по алгоритму, рассмотренному в примере 2.3. Однако в отличие от модели ЕС-1022, здесь в процессе перевода умножение на два выполняется путем сдвига операндов влево. Такой процесс при сдвиге переводимого числа на каждом этапе перевода на разряд влево реализуется в общем следующим образом.

Выталкиваемые из одного регистра двоичные цифры последовательно передаются в другой регистр, где формируются и накапливаются двоично-десятичные тетрады переведенного числа. После каждого сдвига образующиеся тетрады анализируются. Если в каких-либо из них суммы двоичных цифр оказываются большими четырех, то к содержимому каждой из таких тетрад прибавляется двоичное число 0011 (вносятся избытки, равные трем). Затем следует очередной сдвиг на разряд влево, и так до тех пор, пока младший разряд переводимого двоичного числа не переместится в младший разряд регистра результата перевода.

Пример 3.18. Перевести двоичное число 0101 1111 в десятичную систему счисления.

Исходное число			0101	1111
Сдвиг 1-й	0000	0000	1011	1110
Сдвиг 2-й	0000	0001	0111	1100
Сдвиг 3-й	0000	0010	1111	1000
Сдвиг 4-й	0000	0101	1111	0000
Коррекция		+ 0011		
	0000	1000	1111	0000
Сдвиг 5-й	0001	0001	1110	0000
Сдвиг 6-й	0010	0011	1100	0000
Сдвиг 7-й	0100	0111	1000	0000
		+ 0011		
	0100	1010	1000	0000
Сдвиг 8-й	1001	0101	0000	0000

В итоге получили десятичное число 95.

3.3. Арифметические операции над двоичными числами с плавающей запятой

В системе команд ЕС ЭВМ имеется обширный 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, имеется два варианта полных наборов команд для операций с плавающей запятой (см. приложение, табл. 2). В каждом из этих вариантов для большинства операций предусмотрены команды как формата RR, так и RX. В данном случае адресами R указываются номера специальных 64-разрядных регистров для операндов с плавающей запятой. Этих регистров четыре (см. 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, характеризующий результат операции.

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. При этом в моделях ЕС-1022, ЕС-1033 и ЕС-1050(52) из характеристики первого операнда всегда вычитается характеристика второго. Если порядки слагаемых или уменьшаемого и вычитаемого неодинаковы, 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 $2^4 = 16$, то при выравнивании порядков сдвиги производятся на количества порядков, кратные четырем. Иными словами, производится сдвиг вправо на количество шестнадцатиричных цифр, равное величине разности характеристик. При разности, большей 8, в ЕС-1050 (52) осуществляются ускоренные сдвиги сразу на восемь разрядов. Одновременно со сдвигом мантиссы вправо на четыре или на восемь 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

$$S = m_S 16^{p_S^*} = m_A 16^{p_A^*} + m_B 16^{p_B^*}, \tag{3.15}$$

где p_A^*, p_B^* и p_S^* – характеристики слагаемых и суммы.

Мантиссы представляются меньшими единицами. 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.

$$B\frac{16^{p_A^*-p_B^*}}{16^{p_A^*-p_B^*}} = \frac{m_B}{16^{p_A^*-p_B^*}} 16^{p_B^*} \cdot 16^{p_A^*-p_B^*} = m'_B 16^{p_A^*}, \tag{3.16}$$

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

$$S = m_S 16^{p_S^*} = m_A 16^{p_A^*} + m'_B 16^{p_A^*} = (m_A + m'_B) 16^{p_A^*}. \qquad (3.17)$$

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 при алгебраическом сложении отрицательная мантисса обычно преобразуется в дополнительный код. Знак результата обычно определяется автоматически путем суммирования значений знаковых разрядов мантисс с учетом переноса из старшего разряда суммы мантисс.

Если в итоге сложения кодов мантисс результат получается в дополнительном коде, то в конце выполнения операции он переводится в прямой код. При сложении мантисс, по абсолютной величине меньших единицы, возможны случаи переполнения разрядной сетки мантиссы результата. Обычно это называют нарушением нормализации влево. Оно устраняется сдвигом мантиссы вправо на четыре разряда и увеличением характеристики результата на единицу. При этом может возникнуть переполнение порядка. Тогда вырабатывается требование прерывания.

Получение результата, по абсолютной величине меньшего 0'.00010...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(52), 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 [8]. Знак суммы при нулевой мантиссе результата всегда положителен.

Пример 3.19. Пусть требуется выполнить операцию СЛОЖИТЬ и НОРМАЛИЗОВАТЬ над числами:

$$[A]_{\text{пр}} = \underbrace{0.1000101'}_{\text{характеристика}}\ \underbrace{'00010010}_{\text{мантисса}}$$

знак числа

$$[B]_{\text{пр}} = \underbrace{1.10000100'}_{\text{характеристика}}\ \underbrace{'11110000}_{\text{мантисса}},$$

знак числа

что соответствует

$$A = m_A 16^{p_A^*} = +\,0{,}00010010 \cdot 10000_2^{1000101};$$

$$B = m_B 16^{p_B^*} = -\,0{,}11110000 \cdot 10000_2^{1\,000\,100}.$$

Для упрощения рассматриваем мантиссы 8-разрядными.

Характеристики p_A^* и p_B^* представляют собой

$$p_A^* = [p_A]_{\text{пр}} + 2^6_{10} = 0.000101' + 1000000_2 = 1000101';$$

$$p_B^* = [p_B]_{\text{пр}} + 2^6_{10} = 0.000100' + 1000000_2 = 1000100',$$

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 2^7, 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:

$$[m_A]_{\text{пр}} = 0.0001\,0010$$

$$+$$

$$[m'_B]_{\text{доп}} = 1.1111\,0001$$

$$[m_A]_{\text{пр}} + [m'_B]_{\text{доп}} = 0.0000\,0011 = [m_A + m'_B]_{\text{пр}}.$$

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 бо́льшая характеристика p_A^*, то

$$p_S^* = 100\,0100.$$

Окончательный результат

$$[S]_{пр} = 0.100\,0100''0011\,0000.$$

Умножение. По команде УМНОЖИТЬ множимое умножается на множитель и нормализованный результат помещается на место первого операнда. Имеется четыре варианта команд УМНОЖИТЬ (см. приложение, табл. 2).

Произведение двух чисел $A = m_A 16^{p_A}$ и $B = m_B 16^{p_B}$ в общем виде

$$C = m_C 16^{p_C} = m_A m_B 16^{p_A + p_B - 64}, \qquad (3.18)$$

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-1033 предварительный порядок произведения вычисляется до перемножения мантисс, в ЕС-1050 (52) – одновременно с их умножением, а в ЕС-1022 – после умножения. Окончательный порядок определяется в конце операции после нормализации результата.

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.

Во всех моделях ЭВМ Единой системы собственно перемножение мантисс чисел, представленных в форме с плавающей запятой, выполняется в основном по тем же алгоритмам, по которым производится умножение чисел с фиксированной запятой. 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, от разрядности мантисс, а в некоторых моделях также и от структуры множителя. Множителем либо всегда является мантисса второго операнда, а множимым – мантисса первого, как например в моделях ЕС-1022 и ЕС-1050 (52), либо в качестве множителя используется операнд, для умножения на который требуется меньше циклов умножения. Этот способ сокращения времени на умножение применен, например, в ЕС-1033. В моделях ЕС-1022 и ЕС-1050 (52) в каждом цикле умножения множимое умножается на четыре разряда множителя, т. е. на одну шестнадцатиричную цифру. В этих моделях в зависимости от разрядности мантисс (24-разрядные или 56-разрядные) количество циклов умножения – соответст-

венно 6 или 14. В модели ЕС-1033 операнды анализируются перед умножением, и в качестве множителя выбирается либо ненормализованная мантисса, либо мантисса первого операнда, если оба сомножителя нормализованы или ненормализованы. Совместно с предварительным поиском нулевого операнда и, тем самым, с обеспечением возможности записи нулевого результата без перемножения сомножителей этим выбором множителя с более короткой значащей частью достигается уменьшение среднего времени, затрачиваемого при решении задач на операции умножения.

Мантиссы результатов как длинных, так и коротких форматов нормализуются. Мантиссы исходных операндов также нормализуются.

После нормализации мантиссы результата, если в этом была необходимость, и окончательного формирования его характеристики произведение анализируется на переполнение характеристик и на исчезновение порядка. При переполнении разрядной сетки характеристики (если она превысит 127) исполнение программы прерывается. Если характеристика результата окажется меньше 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. 2) 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 они вначале нормализуются обычно с точностью до шестнадцатиричной цифры. Затем вычисляется предварительная характеристика частного и осуществляется деление мантисс делимого и делителя.

При нормализации операндов их характеристики уменьшаются на количество единиц, соответствующее количеству старших нулевых шестнадцатиричных цифр в исходных мантиссах. По этим скорректированным характеристикам и вычисляется предварительная характеристика частного. Для этого из характеристики нормализованного делимого вычитается характеристика нормализованного делителя и к полученной разности прибавляется число $64_{10}(1000\,000_2)$, т. е. находится $p^*_{дл} - p^*_{дл} + 64$. Этот результат затем может корректироваться после вычисления мантиссы частного.

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.

Процессы деления мантисс в моделях ЕС-1050 (52) и ЕС-1022 в основном строятся по тем же алгоритмам, по которым в данных моделях ЭВМ делятся целые числа в операциях с фиксированной запятой. Различие 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 восстановле-

ния остатка. Знак частного определяется путем анализа знаков делимого и делителя. Результирующий знак заносится в нулевой разряд кода частного в конце операции.

В модели ЕС-1022, 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: если остаток положительный, то очередной разряд частного равен единице, а если отрицательный, то – нулю. После выработки четырех двоичных цифр 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.

В конце деления определяется окончательное значение характеристики частного. В модели ЕС-1022 это осуществляется следующим образом. Если старшая шестнадцатиричная цифра мантиссы частного получилась равной нулю, то его характеристика равна исходной, полученной в начале деления. Нулевая старшая шестнадцатиричная цифра частного при этом не учитывается. Если же старшая шестнадцатиричная цифра частного отлична от нуля, то начальная характеристика, полученная в начале деления, увеличивается на единицу. При этом производится анализ характеристики на переполнение. В случае ее переполнения происходит прерывание исполняемой программы.

Получение старшей шестнадцатиричной цифры, отличной от нуля, при делении нормализованных мантисс по сути дела есть нарушение нормализации частного влево. Оно возникает, когда мантисса нормализованного делимого оказывается больше мантиссы нормализованного делителя. Учет всей мантиссы частного и коррекция характеристики равносильны сдвигу мантиссы результата на одну шестнадцатиричную цифру вправо.

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.

В модели ЕС-1033 мантиссы делятся не так, как целые двоичные числа. Для деления мантисс чисел с плавающей запятой применен способ деления, по которому в каждом цикле деления формируются два двоичных разряда частного.

Во всех моделях ЭВМ ЕС 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. 2) 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 оперативной 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 [8].

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-1033 оно осуществляется путем сдвига второго операнда вправо или влево таким образом, чтобы его правый крайний байт оказался выравненным по правой границе первого операнда. В модели ЕС-1050(52) в выбранных в АУ двойных словах местоположения правых границ операндов фиксируются специальными счетчиками. От положений границ ведется счет количеств байт, составляющих десятичные числа, над которыми производится операция. В модели ЕС-1022 при каждом обращении к ООП выбирается слово, содержащее 4 байта. В двух словах, в которых располагаются начала операндов, младшие цифры слагаемых также могут располагаться различно в любых байтах слов — в четных, либо в нечетных байтах, либо в байтах различной четности. Это обстоятельство учитывается при организации выполнения операции. Организуются специальные счетчики, в которые заносятся пары младших разрядов кодов адресов оперативной памяти, где расположены операнды.

По ним в микропрограмме операция действия автоматически организуются так, 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 поразному. Например, в моделях ЕС-1033 и ЕС-1050 (52) всегда из первого операнда вычитается второй, а в модели ЕС-1022 всегда вычитается более короткий операнд.

В модели ЕС-1033 реализуется непосредственное вычитание с помощью универсального арифметико-логического блока. В модели ЕС-1050 (52) вычитание реализуется путем сложения прямого кода первого операнда с дополнительным кодом второго (дополнением до 10^n; см. п. 2.5). Если уменьшаемое оказалось меньше вычитаемого, то результат получается в дополнительном коде. Чтобы представить его в прямом коде, он повторно вычитается из нуля. 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.

Непосредственное сложение кодов операндов в модели ЕС-1050 (52) выполняется побайтно, в модели ЕС-1022 — по два байта, а в ЕС-1033 — 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-1022 и ЕС-1050 (52) представляется первый операнд, а в ЕС-1033 — второй; при разных знаках избыток содержится в дополнении.

Пример 3.20. 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 10^2 равно 25; тогда

$+\overset{82}{25}$ или	$+\overset{1000\ 0010}{1000\ 1011}$	A Дополнение B (с избытком 6 в каждой тетраде)
$\pi \leftarrow \overline{07}$	$\pi \leftarrow 0000\ \overset{1101}{+1010}$	Коррекция результата
	0000 0111	Истинный результат $A + B$.

В выполненных действиях дополнение второго числа определено путем инвертирования значений двоичных разрядов в тетрадах и прибавлением единицы к младшему разряду правой тетрады. При инвертировании обратные коды тетрад

получаются с 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. Следует заметить, что надо учитывать и переносы, которые могут возникнуть при прибавлении единицы в процессе образования дополнения.

Наличие переноса п 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. Вычесть из $A = 75$ $B = 82$, т. е. найти разность $75 - 82$. Вычитание заменяем сложением $75 + (-82)$ или $75 + [-82]_{доп}$. Дополнение 82 до 10^2 равно 18; тогда

$+\frac{75 \text{ или}}{18}$	$+\frac{0111\ 0101}{0111\ 1110}$	A	
		Дополнение B (с избытком 6 в каждой тетраде)	
	1111 0011	Нескорректированный результат в дополнительном коде	
	$+$ ↑п↓		
	1010	Коррекция	
93	1001 0011	Десятичное дополнение числа −7	

Отсутствие переноса из старшей тетрады при сложении A с дополнением $-B$ свидетельствует о том, что результат получился в виде дополнения до 10^2. 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.

$+\frac{00}{[93]_{доп}}$	$+\frac{00000000}{01101101}$	Двоичное дополнение десятичного дополнения результата
	$+\frac{01101101}{10101010}$	Коррекция
07	00000111	Прямой код результата (без знака)

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 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 моделях ЕС-1022 и ЕС-1033 — это 4 байт, а в ЕС-1050 (52) — 8 байт. В ЕС-1050 (52) операнды полностью выбираются из оперативной памяти. В ЕС-1033 вначале выбирается из ООП второй операнд и выравнивается по правой границе первого операнда. Операция сложения или вычитания выпол-

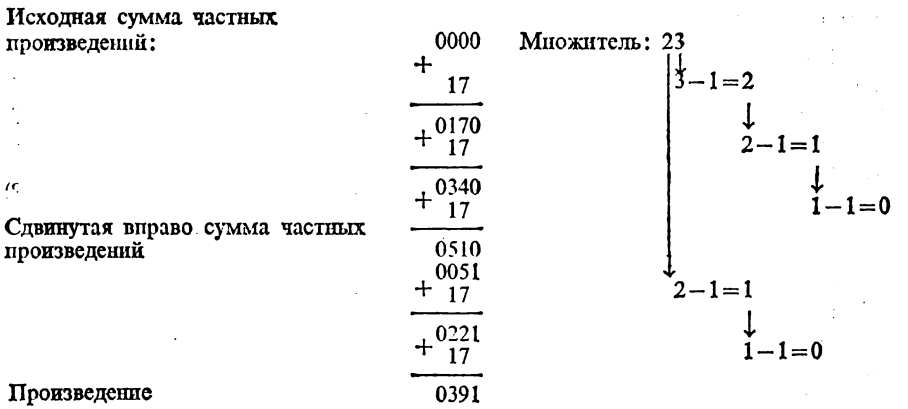
няется на фоне выборки первого операнда. В ЕС-1022 оба операнда выбираются для выполнения над ними действий по частям (по четыре байта). В ЕС-1033 второй операнд после выравнивания записывается в специальную местную память процессора – в операционные регистры. В ЕС-1050 (52) выбранные операнды также 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 операнд, имеющий меньшую длину, дополняется слева необходимым количеством нулевых байтов.

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-1033 и ЕС-1050 (52) 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-1022 результат по мере выполнения операции над парами байт накапливается в локальной памяти процессора, откуда он может быть выбран при взятии итогового дополнения. После окончательного формирования результата в прямом коде он отсылается в ООП.

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 3.21. Найти произведение 17×23.



Число 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 правой тетраде его младшего байта в конце выполнения операции.

В модели ЕС-1022 в процессе накопления сумм частных произведений для ускорения умножения, кроме самого множимого, используются еще его удвоенные и учетверенные значения, а в модели ЕС-1033 – еще и ушестеренные и увосьмеренные. Они получаются путем специальных действий, например сдвигом множимого в начале операции при вызове из ООП первого операнда. Для удвоения множимое сдвигается на один двоичный разряд влево и, если необходимо, корректируется. Далее удвоенное значение аналогично сдвигается еще на разряд влево для получения учетверенной величины.

Например, удвоенные и учетверенные двоично-десятичные значения множимого 17 ($0001\,0111_{2-10}$) будут:

Множимое	0001	0111	
Сдвиг на один разряд влево	+ 0010	1110	
Коррекция		0110	
Удвоенное множимое	0011	0100	(17·2 = 34)
		↑п ←	
Учетверенное множимое	0110	1000	(34·2 = 68)

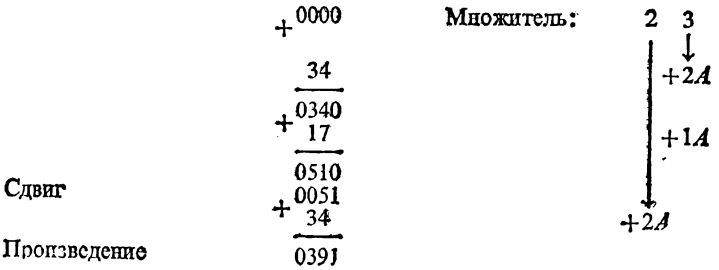
Если до сдвига в каких-либо тетрадах находились цифры, большие четырех, то после сдвига производится коррекция. К полученным значениям цифр в этих тетрадах прибавляются цифры 6 с учетом переносов в соседние слева старшие тетрады.

Как множимое A, так и его кратные значения на время операции хранятся в специально отведенных для них (рабочих) ячейках местной оперативной памяти. В процессоре модели ЕС-1033 для этого используются операционные регистры. В этой модели при умножении на очередную цифру множителя в зависимости от ее величины выполняются действия, указанные в табл. 3.4.

Таблица 3.4

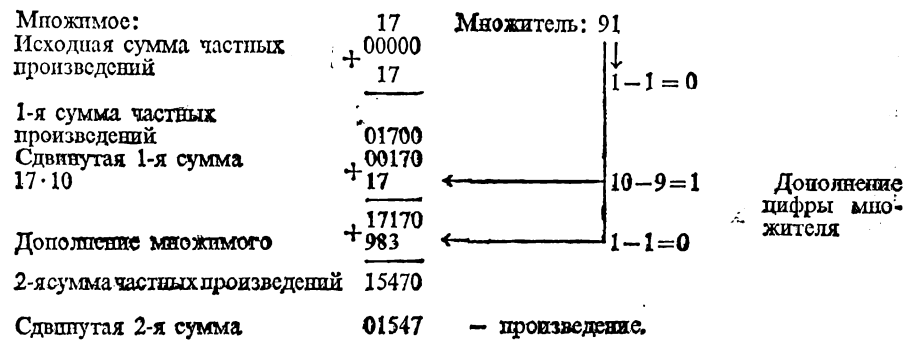
Очередная цифра множителя	Вариант суммирования при накоплении суммы частных произведений в модели ЕС-1033	Действия при накоплении суммы частных произведений в модели ЕС-1022	Очередная цифра множителя	Вариант суммирования при накоплении суммы частных произведений в модели ЕС-1033	Действие при накоплении суммы частных произведений в модели ЕС-1022
0	0	0	5	$+4A+1A$	$+4A+1A$
1	$+1A$	$+1A$	6	$+6A$	$-4A$
2	$+2A$	$+2A$	7	$+6A+1A$	$-2A-1A$
3	$+2A+1A$	$+2A+1A$	8	$+8A$	$-2A$
4	$+4A$	$+4A$	9	$+8A+1A$	$-1A$

Следуя этим указаниям, процесс умножения, рассмотренный в примере 3.1, можно представить следующим образом:



В модели ЕС-1050 (52) 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 множителя 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 3.22. Способом комбинированных действий найти произведения 17×91.



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-1022. 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 слева цифра 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-1033 оно выполняется методом последовательного вычитания делителя из делимого с восстановлением остатка. После каждого вычитания, если остаток неотрицательный, формируемая цифра частного увеличивается на единицу. В общем случае процесс деления с восстановлением остатка можно проиллюстрировать следующим примером.

Пример 3.23. 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
	060 → 12 Формирование второй цифры частного
	+94
	000 → 13
	частное

В данном примере вычитание делителя 06 заменяется прибавлением дополнения 94. Второй остаток получился отрицательным. Для восстановления к нему пришлось прибавить делитель.

В модели ЕС-1033 для ускорения такого процесса деления наряду с самим делителем используется его учетверенное значение. Каждый новый этап определения очередной цифры частного начинается с вычитания этого учетверенного значения. Если результат вычитания положительный, то в формируемый разряд частного заносится цифра 4, и вычитание учетверенного значения делителя повторяется. При получении в результате вычитания учетверенного делителя отрицательного остатка осуществляется переход к вычитанию просто делителя. Делитель вычитается из делимого или очередного остатка, вычитание из которого учетверенного значения дает отрицательный результат. Это предыдущее значение сохраняется в операционном регистре процессора, что исключает необходимость шага восстановления остатка. После каждого вычитания делителя, если получается положительный остаток, то формируемая цифра частного увеличивается на единицу. Формирование очередной цифры частного заканчивается на шаге получения в результате вычитания делителя отрицательного или

нулевого остатка. Для продолжения процесса деления остаток, полученный на предыдущем шаге, сдвигается на тетраду влево, и новый этап деления начинается с вычитания учетверенного делителя.

В модели ЕС-1050 (52) 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 3.24. Выполним процесс деления 162 на 6 по методу без восстановления остатка.

Исходное делимое	162	
Дополнение делителя	+94	
	102	→ 1
	+94	
	042	→ 2
	+94	
	982	
Сдвиг делителя	+6	
	988	→ 29
	+6	
	994	→ 28
	+6	
	000	→ 27

В модели ЕС-1022 реализован комбинированный метод деления с частичным восстановлением остатка. Если при действии восстановления остаток полностью не восстановится, то следующая цифра частного определяется как по методу деления без восстановления остатка. В действиях вычитания и прибавления при определении цифр частного участвует удвоенный делитель.

Во всех моделях 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 в поле делимого. В поле делимого первая слева цифра должна быть нулем. 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. 2) 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, суммирования по модулю два (соответственно операции *И, ИЛИ* и *Исключающее ИЛИ*), проверки, перекодировки, редактирования, сдвига и специальных видов пересылки.

Только в операциях редактирования данные — десятичные числа рассматриваются как числа. В остальных логических операциях операнды трактуются как группы восьмиразрядных двоичных кодов (байтов), не имеющих количественного значения. В некоторых операциях действия производятся раздельно над левыми и правыми тетрадами байт. Байты операндов обрабатываются слева направо. Кроме сдвигов и пересылок, после каждой из остальных логических операций устанавливается код условия [8].

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 логического умножения $x_i y_i$, логического сложения $x_i \vee y_i$ или суммирования по mod 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 (см. 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 (см. табл. 2.4) кодируемая условным числом 0111 0100 (116_{10}). В словаре код этой буквы, согласно ДКОИ (см. табл. 2.5) равный 1010 0011 (163_{10}), должен располагаться в 116-м 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 числах 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 (см. рис. 2.12). Длина *L* операнда, задаваемая командой, относится к образцу. Один байт образца дает возможность отредактировать одну цифру редакти-

руемых 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 5C. 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. 5 и табл. 2.3):

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 – незначащий 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* (выбор цифры). Он замещается либо первой цифрой исходного поля, либо символом-заполнителем. Для того чтобы определить вид замещения, анализируется значение цифры исходного поля. Значащая цифра посылается 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.

Пример 3.25. Предположим, что имеем

исходное поле	0 1 2 7 3 0 5 C 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.
Результат редактирования:	*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 заполнении 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 при обнаружении в образце символов начала значимости и разделителя полей. Подробное описание применения этих символов можно найти, например, в работе [8].

Глава 4

ЭЛЕМЕНТНАЯ БАЗА ЭВМ

4.1. Типовые конструктивные модули

При конструировании электронной аппаратуры ЭВМ Единой системы было предусмотрено применение унифицированной конструктивно-технологической базы, основанной на широком использовании современных достижений микроэлектроники. Электронная аппаратура строится по блочному принципу. Основными ее элементами являются полупроводниковые интегральные схемы (ИС).

Применение интегральных схем позволило использовать типовые конструктивные решения. Теперь простейшей составной частью электронной аппаратуры машин и периферийных устройств является 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 до 40 ТЭЗ.

Блоки панелей крепятся в рамах *4*, представляющих собой прямоугольные каркасы сварного типа из нормализованных профилей с перемычками и направляющими для установки блоков панелей. В рамах также размещаются блоки питания. Для осуществления межрамных соединений на рамах крепятся разъемы.

Рамы помещаются в защитные кожухи. В зависимости от размеров и назначения устройства рама с кожухом оформляется 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.

Все рассмотренные составляющие части конструкций машин обладают конструктивной завершенностью и могут быть изготовлены 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.

Таблица 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

Унификация конструктивных модулей различных уровней в наибольшей мере способствует удовлетворению требований, предъявляемых к конструкциям современных серийно выпускаемых 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. Интегральные схемы

В моделях ЕС-1022—ЕС-1033 электронная аппаратура процессоров и каналов строится в основном на интегральных транзисторно-транзисторных логических схемах (ТТЛ-схемы). На ТТЛ-схемах также выполняются устройства управления аппаратуры ввода-вывода и накопителей всех моделей. В процессоре и каналах модели ЕС-1050(52) применяются более быстродействующие элементы — интегральные транзисторные логические схемы на переключателях тока с объединенными эмиттерами, получившие название эмиттерно-связанных логических схем (ЭСЛ-схемы). В наборах комплексов тех и других схем имеются интегральные схемы различной степени интеграции.

Транзисторно-транзисторные логические схемы. Интегральные схемы этого типа изготовляют на пластинах кремния по планарно-эпитаксиальной технологии. Кремниевые пластины с выполненными на них микро-

схемами заключают в герметические пластмассовые корпуса *1* прямоугольной формы с 14—16-штыревыми выводами (рис. 4.1).

Функционально полную основу комплекса таких интегральных схем составляют универсальные элементы типа *И—НЕ*. Они являются логи-

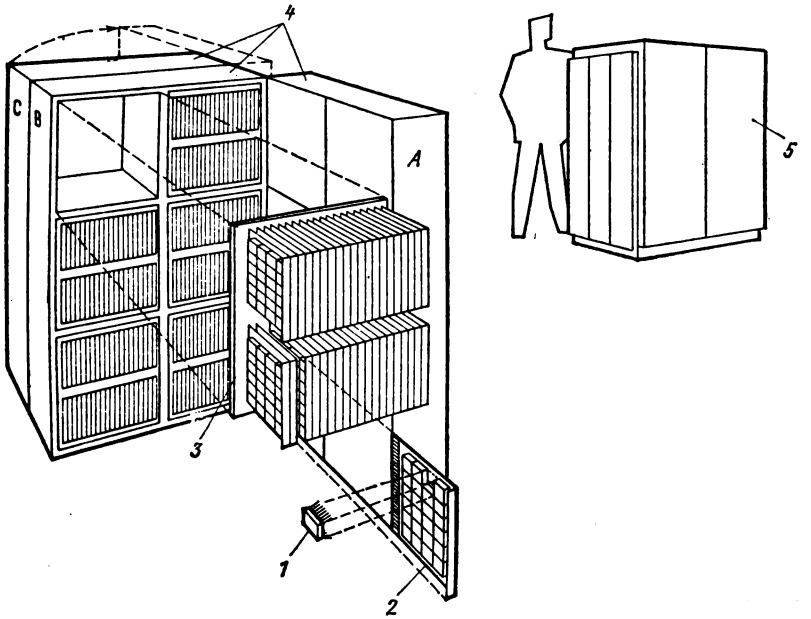


Рис. 4.1. Вариант типовой конструкции стойки ЭВМ:

1 — модуль интегральной схемы; *2* — типовой элемент замены (ТЭЗ); *3* — блок панели; *4* — рамы; *5* — стойка, оформленная в виде шкафа

ческими 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, *а* и *б*.

Принципиальная схема содержит двухэмиттерный транзистор *Т1*, на котором выполнена двухвходовая схема *И*, и сложный инвертор, собранный на транзисторах *Т2—Т4*, диоде *Д* и резисторах *R2—R4*. При напряжении питания $E_k = +5$ В высокий уровень выходного напряжения $U_1 \geqslant \geqslant +2{,}4$ В, а низкий — $U_0 \leqslant +0{,}4$ 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 *Т1* имеет бóльшее число эмиттеров, например 3, 4, 8. В зависимости от этого входная схема совпадения имеет

соответствующее количество входов. Она работает следующим образом. Если на все эмиттеры транзистора *T1* подаются высокие уровни потенциала U_1, то переходы база – эмиттер оказываются смещенными в обратном направлении, а переход база – коллектор – в прямом. Через ре-

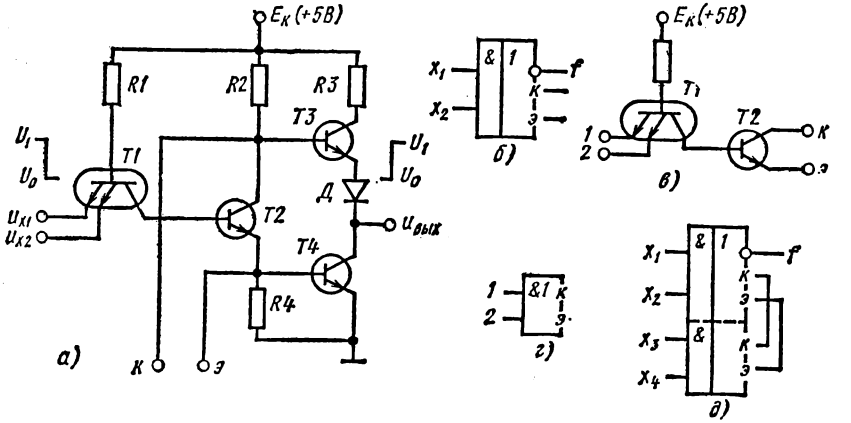
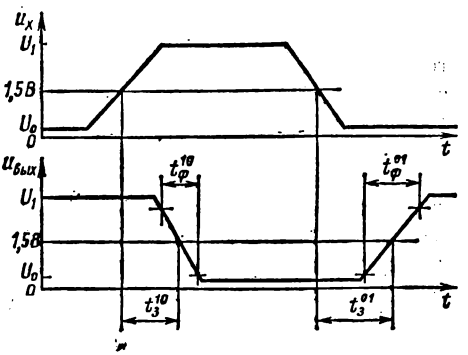


Рис. 4.2. Транзисторно-транзисторные логические схемы: *а, б* — принципиальная схема и условное изображение элемента *И-НЕ*; *в, г* — принципиальная схема и условное изображение расширителя схемы *И-ИЛИ-НЕ* по входу *ИЛИ*; *д* — пример схемы *И — ИЛИ — НЕ* с подключенным расширителем

зистор *R1* и смещенный в прямом направлении переход база – коллектор протекает прямой ток базы транзистора *T2*. Этим током последний открывается и вводится в режим насыщения. Одновременно полностью открывается и транзистор *T4*, на котором выполнен выходной инвертор. С коллектора транзистора *T4* снимается низкий уровень выходного

Рис. 4.3. Формы входного u_x и выходного $u_{вых}$ сигналов ТТЛ-элемента



напряжения U_0. Транзистор *T3* при этом находится в закрытом состоянии. Оно обеспечивается низким уровнем напряжения, установившимся на коллекторе транзистора *T2*. Кроме того, надежному запиранию *T3* способствует наличие диода *Д*. Это объясняется следующим образом.

При открытом транзисторе *T4* на его коллекторе устанавливается напряжение + (0,1 ÷ 0,4) В, а на коллекторе открытого транзистора *T2* – напряжение $U_{б.эT4} + U_{к.эT2} \approx 0{,}7$ В + (0,1 ÷ 0,4) В = (0,8 ÷ 1,1) В. Разность между напряжениями на коллекторах транзисторов *T2* и *T4* 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 транзисторов *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 неосновных носителей из базы транзистора *Т2* и его запирание. 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 $u_{вых}$ 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 $f = \overline{x_1 x_2}$, где x_1, x_2, f – соответственно u_{x1}, u_{x2}, $u_{вых}$. Сложный инвертор имеет достаточно высокую нагрузочную способность. Применяемые интегральные схемы рассматриваемого типа характеризуются следующими основными параметрами:

$$t_з^{10} = (15 \div 18) \text{ нс}; \; P_{ср} = (25 \div 35) \text{ мВт};$$
$$t_з^{01} = (25 \div 33) \text{ нс}; \; m \leqslant 8;$$
$$t_{ф0} = (14 \div 16) \text{ нс}; \; l \leqslant 8;$$
$$t_{ф1} = (18 \div 22) \text{ нс}; \; n \leqslant 10,$$

где $t_з^{10}$ и $t_з^{01}$ – задержки переключения выходного напряжения соответственно с уровня U_1 на уровень U_0 и с U_0 – на U_1; $t_{ф1}$ и $t_{ф0}$ – длительности фронта и среза выходного сигнала; $P_{ср}$ – 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 *к* и *э* (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 включения транзисторов *Т2* элементов ЛР и ЛП получаются логические схемы, подобные изображенной на рис. 4.2, *д*. Эта схема реализует логическую функцию $f = x_1 x_2 \vee x_3 x_4$. 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.

Транзисторные схемы со связанными эмиттерами (типа ЭСЛ). Схемы этого типа также изготовляют по планарно-эпитаксиальной технологии. Корпуса микросхем – прямоугольные пластмассовые с 14–16-штыревыми выводами, как и у ТТЛ-схем.

Функционально полную основу комплексов элементов на переключателях тока составляют универсальные элементы типа *И, И–НЕ*, имею-

щие парафазные выходы. Принципиальная схема такого базового элемента, применяемого в ИС малой степени интеграции, 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*,

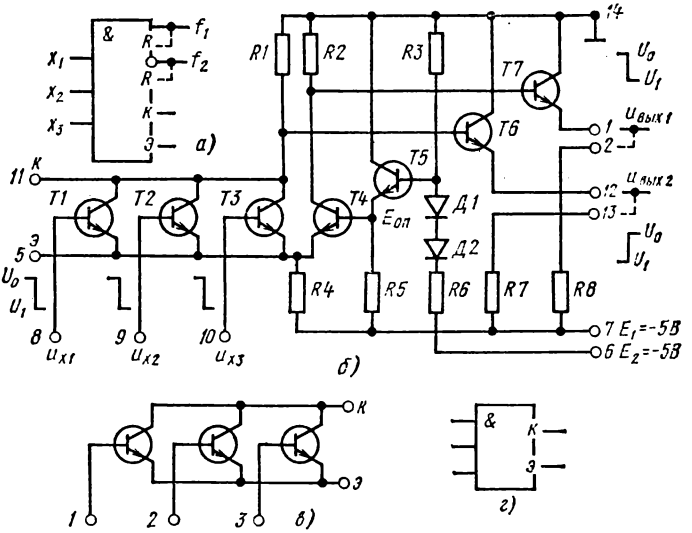


Рис. 4.4. Схема элемента на переключателе тока со связанными эмиттерами: *а,б* – условное изображение и принципиальная схема универсального элемента *И, И–НЕ*; *в, г* – принципиальная схема и условное изображение расширителя для схемы *И*

диодах *Д1*, *Д2* и резисторах *R3*, *R5* и *R6* собран источник опорного напряжения $E_{оп} = -1{,}2$ 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 $+(0{,}25 \div 0{,}4)$ В и $-(0{,}25 \div 0{,}7)$ В, симметричными относительно опорного напряжения. Формы входного и выходных сигналов элемента типа ЭСЛ приведены на рис. 4.5.

При низком уровне напряжения U_1 на всех трех входах *8–10* (см. 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, протекавший через транзистор *Т4*, переключается в открывшийся транзистор.

В кремниевых транзисторах падение напряжения на переходе база — эмиттер примерно равно (0,7 — 0,8) В. Примем это падение напряжения равным 0,7 В. Тогда при входном сигнале $U_0 = -0{,}95$ В, на эмиттерах транзисторов $T1-T4$ будем иметь напряжение $U_0 - U_{б.э} = -0{,}95 - 0{,}7 = -1{,}65$ В. Разность потенциалов между $E_{оп} = -1{,}2$ В и этим напряжением составит $E_{оп} - (U_0 - U_{б.э}) = -1{,}2 + 1{,}65 = 0{,}45$ В. Она недостаточна для отпирания транзистора $T4$, так как порог его срабатывания находится в диапазоне 0,7 — 0,8 В.

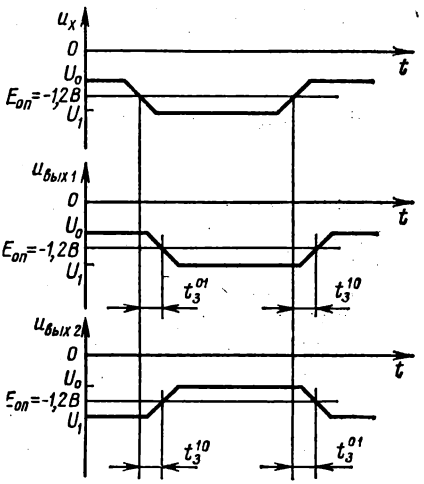


Рис. 4.5. Формы входного u_x и выходных $u_{вых1}$, $u_{вых2}$ сигналов элемента типа ЭСЛ

Если же, наоборот, открыт транзистор $T4$, то на объединенных эмиттерах будет $E_{оп} - U_{б.э} = -1{,}2 - 0{,}7 = -1{,}9$ В. Это имеет место, когда на все входы подаются, например, сигналы уровней $U_1 = -1{,}45$ В. В этом случае разности потенциалов $U_1 - (E_{оп} - U_{б.э}) = -1{,}45 + 1{,}9 = 0{,}45$ В не хватает для отпирания транзисторов $T1-T3$.

В схемах рассматриваемого типа транзисторы работают в ненасыщенном режиме. Это способствует высокой скорости переключения таких элементов. В интегральных схемах типа ЭСЛ малой степени интеграции применяются базовые элементы в основном двух типов. Одни из них характеризуются средним временем задержки распространения сигналов $t_{з.ср} = 5$ нс при средней мощности рассеяния $P_{ср} = 70$ мВт, другие — $t_{з.ср} = 8$ нс при $P_{ср} = 35$ мВт.

Выходные эмиттерные повторители, формирующие выходные сигналы, обеспечивают высокую нагрузочную способность элементов $n \leqslant 15$. Они служат развязкой между переключателями тока и нагрузкой. Кроме того, они обеспечивают симметричность выходных сигналов относительно уровня $-1{,}2$ В. На коллекторах транзисторов $T1-T3$ и $T4$ 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 $T6$ и $T7$.

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 $f_1 = x_1x_2x_3$, а второй (клеммы *12*, *13*) — соотношением $f_2 = \overline{x_1x_2x_3}$, где x_1 соответствует u_{x1}, $x_2 - u_{x2}$, ..., $f_2 - u_{вых.2}$. Очевидно, что $f_2 = \overline{f_1}$.

Для возможности увеличения количества входов у схем рассматриваемого типа в комплексах интегральных схем 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 к и э 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 быть до $M = 9$.

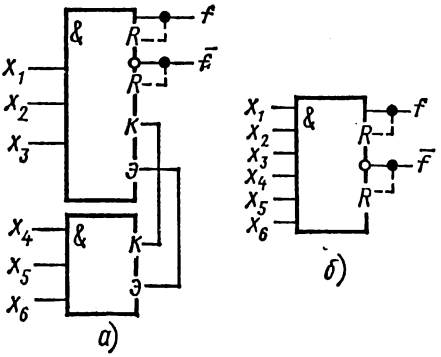


Рис. 4.6. Схема подключения расширителя: *а* — логическая схема; *б* — условное изображение элемента *И, И-НЕ* с увеличенным числом входов

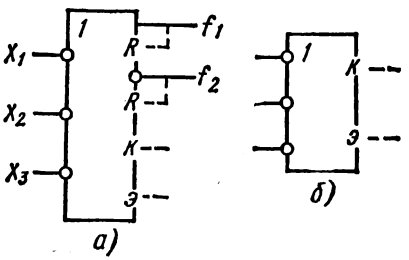


Рис. 4.7. Условные изображения: *а* — элемента ИЛИ, ИЛИ — НЕ; *б* — расширителя для схемы ИЛИ

Элементы на токовых переключателях можно рассматривать и как схемы типа *ИЛИ, ИЛИ—НЕ*, если коду 1 поставить в соответствие высокий уровень потенциала, а коду 0 — низкий. При таком рассмотрении элементов входы в них изображаются инверсными, как показано на рис. 4.7. В этом случае выходные переменные f_1 и f_2 на рис. 4.7, *а* связаны с входными $x_1 - x_3$ логическими соотношениями: $f_1 = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ и $f_2 = \overline{\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3}$. Здесь также $f_2 = \bar{f}_1$.

Для расширения входов схем *ИЛИ* применяются такие же расширители, как в схемах *И*. Поэтому допустимый обобщенный предел расширения входов токовых переключателей выше был обозначен буквой *M*, в отличие от *m* — допустимого коэффициента объединения только по схеме *И*.

В комплексах ИС малой степени интеграции имеется несколько модификаций элементов *И, И—НЕ*, различающихся количествами и типами входов и выходов, а также нагрузочной способностью. Кроме того, в комплексах есть различные модификации триггеров, а также такие схемы, как полусумматоры и др.

Принципиальная схема полусумматора приведена на рис. 4.8, *а*. Она состоит из двух двухвходовых токовых переключателей с общим опорным плечом, выполненным на двухэмиттерном транзисторе *Т3*. 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 *T3* будет заперт только тогда, когда будут открыты по одному из транзисторов в парах: *T1*, *T2* и *T4*, *T5*. Это соответствует случаю, когда переменные неравнозначны. При таком их сочетании на выходе *1* будет выдаваться высокий уровень потенциала, соответствующий значению $\overline{x\bar{y}} \vee \overline{\bar{x}y} = xy \vee \bar{x}\bar{y}$, как показано на условном изображении полусумматора на рис. 4.8, *б*. Здесь, как и ранее, кодам 1 на входах и выходах ставится 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 *S*. На выходах *P* (клеммы *2*, *3*) и *1* (клеммы *12*, *13*) формируются сигналы, соответствующие инверсии логического произведения $\bar{x}\bar{y}$ и логической сумме $x \vee y$.

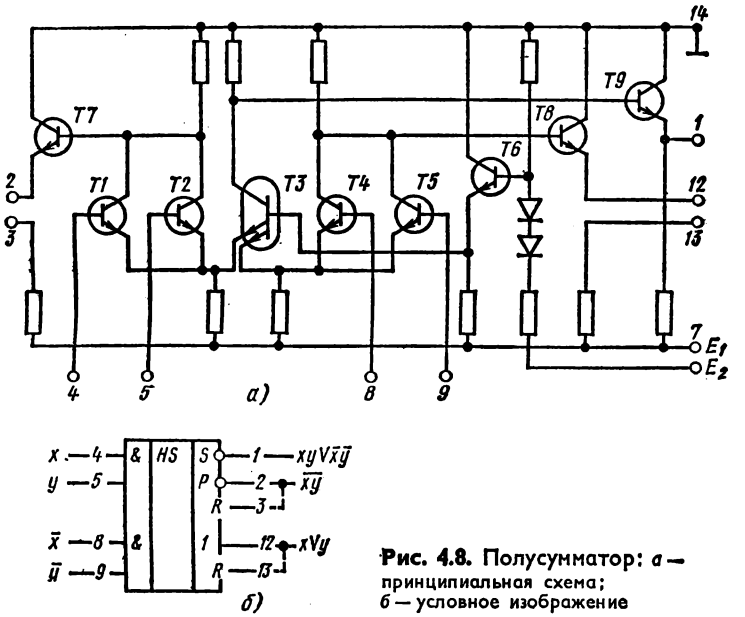


Рис. 4.8. Полусумматор: *а* — принципиальная схема; *б* — условное изображение

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 всех серий $L \leqslant 5$.

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 рассмотренных интегральных схемах 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 соответствие низкий уровень потенциала, то объединенная схема реализует логическую формулу $f_1 = x_1x_2x_3x_4x_5x_6$.

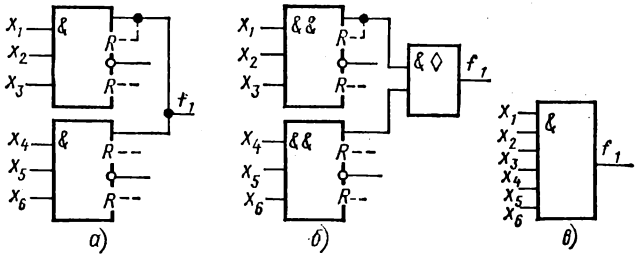
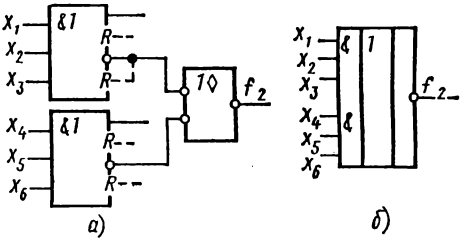


Рис. 4.9. Схема объединения прямых выходов элементов *И*, *И—НЕ*: *а* — непосредственное изображение объединения; *б* — изображение объединения с помощью псевдоэлемента; *в* — условное изображение расширенной схемы *И*

На функциональных схемах непосредственное соединение выходов нескольких элементов, благодаря которому реализуются новые логические связи, обозначают псевдоэлементом. При чтении схем это позволяет исключить неоднозначность в определении направления распространения информации и функции, реализуемой таким соединением (монтажной логикой). К символам функций соединяемых логических элементов добавляют знак & или 1, или сочетание этих знаков в соответствии с функцией псевдоэлемента (см. ГОСТ 2.743—72).

Рис. 4.10. Схема объединения инверсных выходов элементов: *а* — изображение с помощью псевдоэлемента; *б* — условное изображение полученной схемы *И—ИЛИ—НЕ*



Изображение такого соединения с использованием псевдоэлемента показано на рис. 4.9, *б*. В данном случае реализуется конъюнкция (монтажное *И*), поэтому в прямоугольнике псевдоэлемента реализуемая функция обозначена совокупностью символов & ◇. Элемент, соединяемый с псевдоэлементом, обозначается двойным символом &&. 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 $f_2 = \overline{x_1x_2x_3 \vee x_4x_5x_6}$. 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 $f_3 = \bar{x}_1 x_2 x_3 x_4 x_5 x_6$. Поэтому, если на схеме, показанной на рис. 4.8, *б*, объединить выходы *Р* 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. $\bar{x}y\,(x \vee y) = x\bar{y} \vee \bar{x}y$.

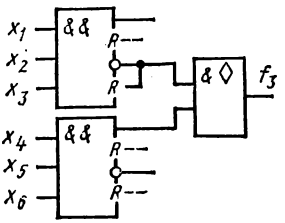


Рис. 4.11. Схема объединения прямого и инверсного выходов элементов

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 также используются как графические единицы для построения функциональных схем.

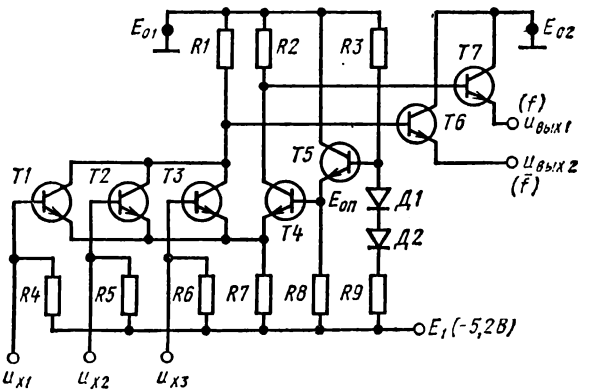


Рис. 4.12. Логический элемент *3И,ЗИ—НЕ* для сигналов низкого уровня

Транзисторные схемы типа ЭСЛ для интегральных схем повышенной степени интеграции. В таких ИС применяются как элементы с токовыми переключателями, подобные рассмотренным и называющиеся одноярусными, так и элементы с последовательно включенными токовыми ключами, получившие название многоярусных.

Пример простейшей типовой одноярусной схемы — трехвходового элемента *3И*, *3И—НЕ* для сигналов низкого уровня — приведен на рис. 4.12. В интегральных схемах повышенной степени интеграции используются подобные элементы с различным числом входов. В рассматриваемой схеме все входные клеммы через резисторы *R4—R6* с сопротивлением

50 кОм подключены к минусовой шине источника питания. Это исключает необходимость подключать незадействованные входы к специальному внешнему источнику напряжения, поддерживающего в закрытом состоянии неиспользуемые входные транзисторы. В выходных эмиттерных повторителях используются внешние нагрузочные резисторы. Величина их может быть от 51 Ом до 2 кОм в зависимости от требований к быстродействию и выходной мощности элемента. В составе комплекса интегральных схем предусмотрены специальные интегральные модули – резисторные сборки, содержащие резисторы разных номиналов. Эти сборки помещаются в такие же корпуса, как и остальные интегральные схемы. Последнее упрощает монтаж нагрузочных резисторов на многослойных печатных платах.

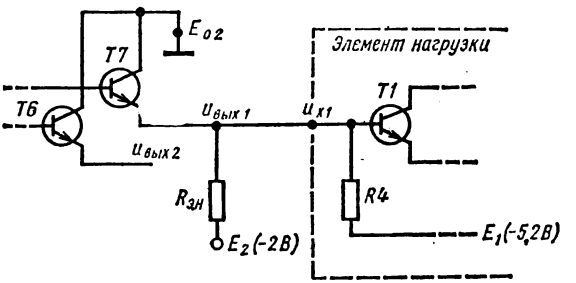


Рис. 4.13. Схема, поясняющая подключение элемента нагрузки и внешнего нагрузочного резистора $R_{э.н}$

Токовый переключатель с встроенным источником опорного напряжения работает при питающем напряжении $E_1 = -5{,}2$ В. Питание выходных эмиттерных повторителей обычно осуществляется от источника напряжения $E_2 = -2$ В. К эмиттерам выходных транзисторов нагрузочные резисторы $R_{э.н}$ подключаются, как показано на рис. 4.13. Чтобы можно было использовать источники различных номиналов, в корпусах интегральных схем предусмотрено по два заземляемых вывода, один из которых соединяется с шиной питания E_{01}, а другой – с шиной E_{02} (см. рис. 4.12).

Логика работы элемента на рис. 4.12 такая же, как у элемента, рассмотренного на рис. 4.4, *б*. Для низких уровней входных сигналов $f = x_1x_2x_3$ и $\bar{f} = \overline{x_1x_2x_3}$. Типовая средняя задержка распространения сигналов в элементах рассматриваемого типа $t_{з.ср} = 3$ нс при средней рассеиваемой мощности $R_{ср} = 25$ мВт на вентиль. В величине $P_{ср}$ не учитывается мощность, рассеиваемая на внешних нагрузочных резисторах $R_{э.н}$. При номинальном питающем напряжении, нагрузке $R_{э.н} = 51$ Ом и $T = 25\,°C$ уровни выходного напряжения $U_{0\,вых} = -0{,}88$ В и $U_{1\,вых} = -1{,}72$ В, при этом пороговыми для входных сигналов являются $U_{0\,пор} = -1{,}05$ В и $U_{1\,пор} = -1{,}475$ В. Средний уровень переключаемого напряжения $U_{ср} = -1{,}29$ В.

Максимальная нагрузочная способность для логических схем в пределах одного ТЭЗ $N \leqslant 20$. Коэффициент $L \leqslant 4$. Номинальное сопротивление нагрузочных резисторов для схем, соединяемых в пределах одного ТЭЗ (в том числе при использовании монтажной логики), равно 100 Ом.

На рис. 4.14 приведена схема *2И—2ИЛИ* для сигналов низкого уровня, т. е. реализующая функцию $f = x_1x_2 \vee x_3x_4$, где x_1 соответствует u_{x1}; $x_2 - u_{x2}$; ...; $f - u_{вых}$. В этой, также одноярусной схеме параллельно работают два токовых ключа: один собран на транзисторах *Т1, Т2, Т3*, а другой — на транзисторах *Т4, Т5, Т6*. Коллекторы опорных транзисторов *Т3* и *Т6* соединены, благодаря чему реализуется на выходе операция *ИЛИ* для сигналов низкого уровня. Транзистор *Т7* ускоряет уста-

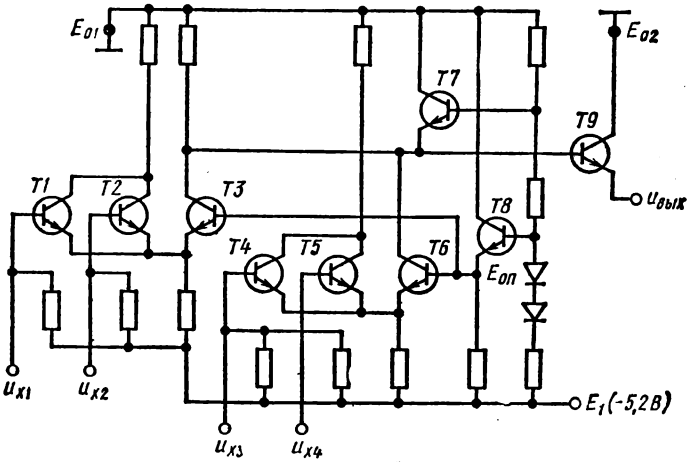


Рис. 4.14. Логический элемент *2И — 2ИЛИ* для сигналов низкого уровня

новление низкого уровня напряжения на выходе схемы. В этой схеме входные и выходные цепи обладают теми же особенностями, что и в предыдущей. Схемы рассмотренного типа получили название селекторов. С их помощью реализуются, например, схемы приема информации в регистры с различных направлений.

4.3. Полупроводниковые запоминающие элементы

В цепях передачи и преобразования информации процессоров, каналов и т. п. для запоминания и временного хранения информации используются триггеры. В ЭВМ ЕС первых выпусков они строились в основном из базовых и расширяемых логических элементов применяемых комплексов интегральных схем. В настоящее время в составах комплексов интегральных схем имеются специальные триггерные микросхемы, выполняемые в виде отдельных модулей. В этих модулях содержатся либо один или несколько несвязанных между собой триггеров, либо триггеры, связанные между собой определенным образом.

В моделях ЭВМ Единой системы применяются симметричные потенциальные триггеры с однофазными и парафазными входами, как одно-

ступенчатые, так и двухступенчатые. Пример принципиальной схемы одноступенчатого триггера с парафазными входами (RS-триггера), выполненного на базовых элементах типа ТТЛ, приведен на рис. 4.15, *а*. 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

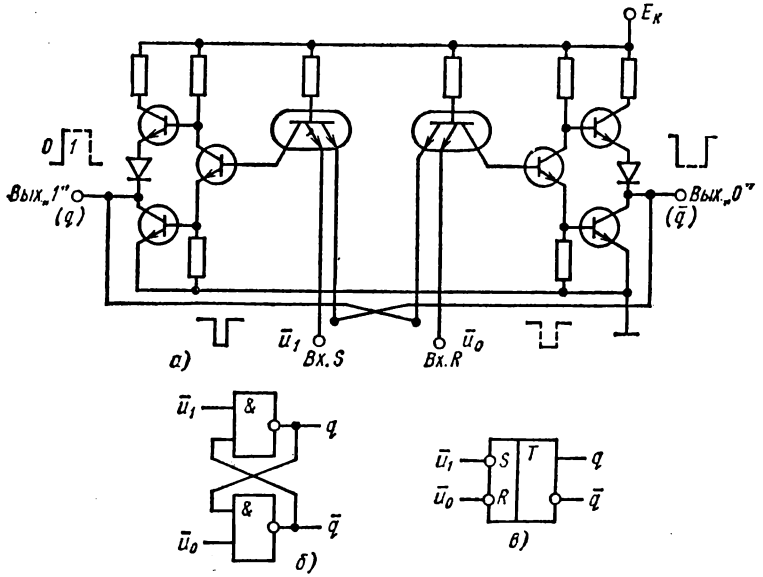


Рис. 4.15. Принципиальная схема RS-триггера на базовых ТТЛ-элементах: *а* — принципиальная схема; *б* — функциональная схема; *в* — условное изображение

установки в нулевое Вх. R и единичное Вх. S 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 производятся сигналами низкого уровня $\bar{u}_1$ и $\bar{u}_0$, подаваемыми на входы S и R. При этом одновременная подача сигналов низкого уровня на оба входа запрещена, т. е. должно соблюдаться условие $\bar{u}_1\bar{u}_0 = 0$. Во время переключения перекрестные обратные связи являются положительными, они способствуют развитию лавинообразного процесса. Тем не менее для надежного переключения длительность входного запускающего сигнала должна превышать время переходных процессов в триггере.

Запоминающие элементы, строящиеся на триггерах, различаются в основном типами входных цепей. Схемы как с двухфазными, так и с одно-

фазными входами обычно выполняются синхронизируемыми. Это необходимо для обеспечения возможности осуществления синхронного приема информации, например, в такие узлы, как регистры.

Функциональная схема синхронизируемого RS-триггера, полностью выполненного на элементах *И—НЕ*, представлена на рис. 4.16, *а*. Его условное изображение дано на рис. 4.16, *б*. Элементы *И—НЕ* могут быть, например, типа ТТЛ. Если используются такие элементы, то при отсутствии совпадения входных сигналов u_1 и u_c или u_0 и u_c на выходах входных схем *И—НЕ* имеют место высокие уровни потенциала. Наличие таких потенциалов необходимо для обеспечения устойчивости собственно триггера в статическом состоянии, как и в схеме на рис. 4.15, *а*.

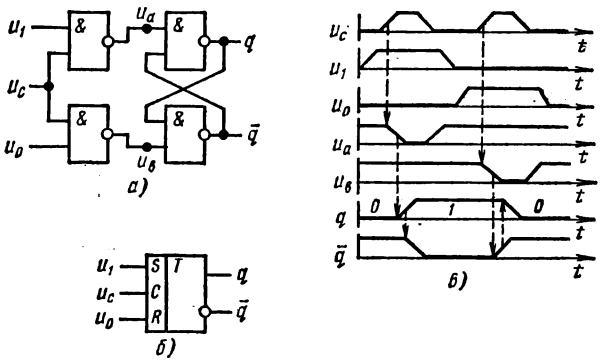


Рис. 4.16. Синхронизируемый RS-триггер на элементах *И-НЕ*: *а* — функциональная схема; *б* — условное изображение; *в* — пример временной диаграммы работы

Пример временной диаграммы работы триггера представлен на рис. 4.16, *в*. Кодовые сигналы u_1 и u_0 могут поступать на входы в произвольные моменты времени. Однако через входные схемы *И—НЕ* они проходят только тогда, когда на входе синхронизации присутствует сигнал u_c. Иными словами, переключение может происходить только при совпадении одного из кодовых сигналов с синхронизирующим. При этом одновременное поступление на входы обоих кодовых и синхронизирующего сигналов запрещено, т. е. налагается ограничение, что $u_0 u_1 u_c = 0$. Следовательно, синхронизируемый RS-триггер является частично определенным автоматом, и его каноническое уравнение переходов имеет вид

$$q_{t+1} = u_1 u_c \vee q_t \overline{u_0 u_c}. \qquad (4.1)$$

Для уменьшения числа межэлементных соединений в схемах ЦВМ широко используются триггеры с однофазными входами — D-триггеры. Эти триггеры также синхронизируемые, и, кроме того, синхронизация выполняется управляемой, т. е. применяются так называемые DV-триггеры. Следует заметить, что и другие типы триггеров часто имеют цепи синхронизации, управляемые специальными внешними сигналами. Это, например, RSV-триггеры.

Пример логической схемы синхронизируемого одноступенчатого DV-триггера приведен на рис. 4.17, *а*. Рассматриваемая схема является элементарным автоматом, функционирующим в соответствии с 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

$$q_{t+1} = u_1 u_{cv} \vee q_t \overline{u}_{cv}, \tag{4.2}$$

где q_t и q_{t+1} — состояния триггера в дискретные моменты времени t и $t+1$; u_1 — входной кодовый сигнал; $u_{cv} = \overline{\overline{u}_c \vee \overline{u}_v}$ (здесь u_c — сигнал синхронизирующей серии, а u_v — сигнал разрешения воздействия синхроимпульсов на триггер).

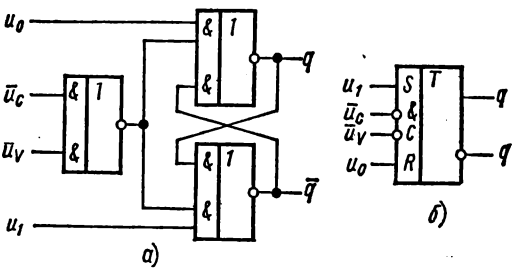


Рис. 4.17. Синхронизируемый триггер с однофазным входом (DV-триггер): *а* — логическая схема; *б* — условное изображение

Подставляя значение u_{cv} в (4.2), после преобразований получим

$$q_{t+1} = u_1 u_c u_v \vee q_t \overline{u_c u_v}. \tag{4.3}$$

С учетом входной схемы для формирования u_{cv} триггер с однофазным кодовым входом D может быть условно изображен, как показано на рис. 4.17, *б*. Такое изображение предназначается для функциональных схем.

Пример схемы синхронизируемого RSV-триггера, построенного на элементах *И—ИЛИ—НЕ* и имеющего раздельные (парафазные) кодовые входы, показан на рис. 4.18, *а*. Такого типа триггеры работают согласно табл. 4.3, в которой также $u_{cv} = \overline{\overline{u}_c \vee \overline{u}_v}$. Этой таблице также соответствует каноническое уравнение вида (4.1). Условно данный триггер с парафазными входами может быть изображен, как показано на рис. 4.18, *б*. Здесь входы установки в единичное и нулевое состояния обозначены соответственно через S и R.

Таблица 4.2

Таблица переходов DV-триггера

u_1	u_{cv}	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 RSV-триггера

u_1	u_0	u_{cv}	q_t	
			0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1

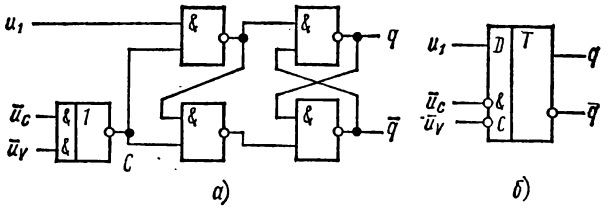
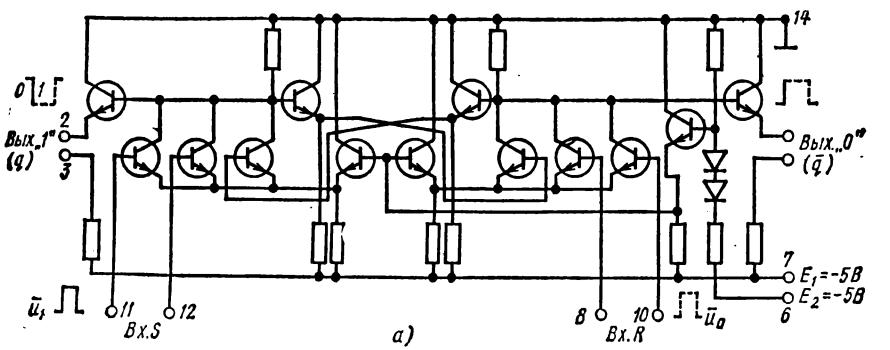


Рис. 4.18. Синхронизируемый триггер с парафазными входами (RSV-триггер), реализуемый на элементах ТТЛ:
а — логическая схема;
б — условное изображение

На токовых переключателях RS-триггер (с раздельными кодовыми входами) может быть построен по схеме, показанной на рис. 4.19. В этой схеме каждый из двух трехвходовых 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 на входах *8, 10* и *11, 12* должны быть низкие уровни потенциала (u_1 и u_0). Переключается триггер входными сигналами высокого уровня $\bar{u}_1$ и $\bar{u}_0$. Для однозначности работы схемы сигналы высокого уровня не должны подаваться на входы противоположных плеч триггера одновременно. В комплексе ИС ЭСЛ-элементов имеется интегральная схема такого триггера. Она служит основой для построения запоминающих элементов — синхронизируемых триггеров с парафазными и однофазными входами.



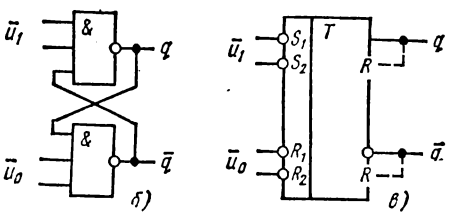


Рис. 4.19. RS-триггер на ЭСЛ-элементах:
а — принципиальная схема; *б* — функциональная схема; *в* — условное изображение

Схема синхронизируемого RSV-триггера приведена на рис. 4.20, *а*. Собственно триггер *Т* с кодовыми входами здесь соответствует схеме, показанной на рис. 4.19, *а*. 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 u_c обозначены синхронизирующие сигналы, а через u_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.

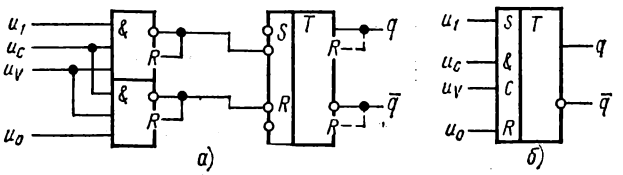


Рис. 4.20. Синхронизируемый RSV-триггер, реализуемый на элементах ЭСЛ: *а* — логическая схема; *б* — условное изображение

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 RSV-триггера, применяемое на функциональных схемах, приведено на рис. 4.20, *б*.

На рис. 4.21, *а* приведена схема DV-триггера. Здесь также сигналам кода 1 на входах u_1, u_c, u_v и выходе q ставятся в соответствие сигналы низкого уровня. Логика работы такой схемы соответствует табл. 4.2.

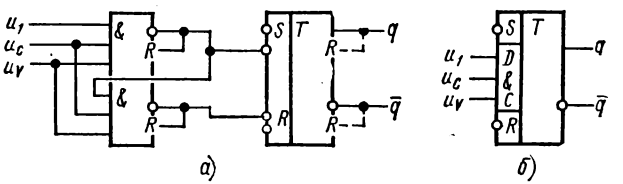


Рис. 4.21. Синхронизируемый DV-триггер, реализуемый на элементах ЭСЛ: *а* — логическая схема; *б* — условное изображение

На рис. 4.21, *б* буквой *D* обозначен кодовый вход, а буквой *С* — вход синхронизации, управляемый разрешающим сигналом u_v. В комплексе ИС на ЭСЛ-элементах имеются специальные интегральные схемы, содержащие полностью выполненные синхронные триггеры с однофазными входами, включая входные цепи.

На интегральных схемах рассмотренных выше типов триггеры со счетными входами (Т-триггеры) строятся двухступенчатыми. В каждой ступени используется синхронизируемый триггер с парафазными входами, например, как показано на рис. 4.22. Выходы q и $\bar{q}$ правого триггера (вторая ступень) соединены перекрестными обратными связями с входами входных элементов *И—НЕ* левого (первая ступень). На другие входы этих входных элементов подаются информационные считаемые сигналы $u_{сч}$ и синхронизирующие сигналы u_{C1}. Сигналы с выходов q' и $\bar{q}'$ триггера 1-й ступени

вместе с синхронизирующими сигналами u_{C2}, сдвинутыми на полпериода относительно u_{C1}, подаются на входы входных вентилей правого триггера.

С одновременным приходом сигналов $u_{сч}$ и u_{C1} (рис. 4.23) вначале в очередное состояние переключается триггер 1-й ступени. Состояние,

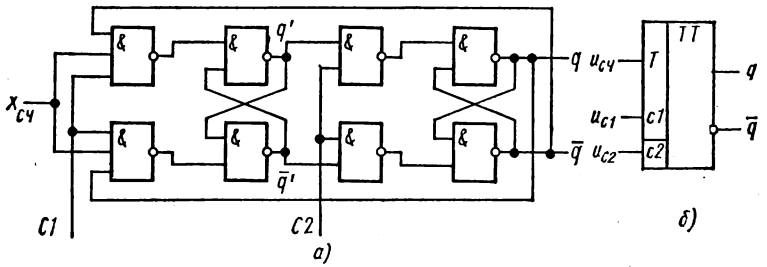


Рис. 4.22. Синхронизируемый двухступенчатый триггер со счетным входом: *а* — логическая схема; *б* — условное изображение

в которое он должен переключиться, указывается сигналами с выходов q и $\bar{q}$ триггера второй ступени. Последний выполняет роль элемента, запоминающего предыдущее состояние всей схемы на время перехода в новое состояние первой ступени. Предыдущее состояние сохраняется, пока отсутствует сигнал u_{C2}. Через перекрестные обратные связи удерживается в открытом состоянии только тот входной элемент *И—НЕ*, через который триггер первой ступени устанавливается в очередное состояние. Триггер второй ступени принимает это же состояние с приходом очередного сигнала u_{C2}. Наличие в рассмотренном триггере дополнительной запоминающей ступени обеспечивает необходимую задержку $t_з$ (рис. 4.23) переключения выходных сигналов, управляющих входами первой ступени, по отношению к входным сигналам. Благодаря этому создаются условия для надежного пропуска входных сигналов через входные схемы совпадения триггера первой ступени и его переключения. По длительности входные сигналы должны превышать время переходных процессов в этом триггере.

Рис. 4.23. Временна́я диаграмма работы триггера со счетным входом

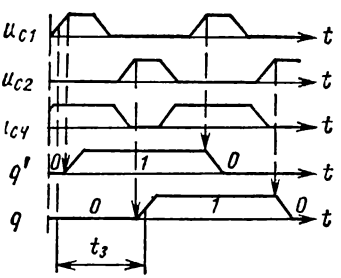


Таблица 4.4

Таблица переходов Т-триггера

$u_{сч}$	q_t	
	0	1
0	0	1
1	1	0

Триггер со счетным входом является вполне определенным элементарным автоматом, функционирующим в соответствии с табл. 4.4.

Согласно этой таблице каноническое уравнение переходов Т-триггера имеет вид

$$q_{t+1} = q_t \bar{u}_{сч} \vee \bar{q}_t u_{сч} \qquad (4.4)$$

где, как и ранее, q_t и q_{t+1} — состояния всей схемы в моменты времени t и $t+1$. Интервал времени между моментами t и $t+1$ равен или больше периода следования входных сигналов $u_{сч}$.

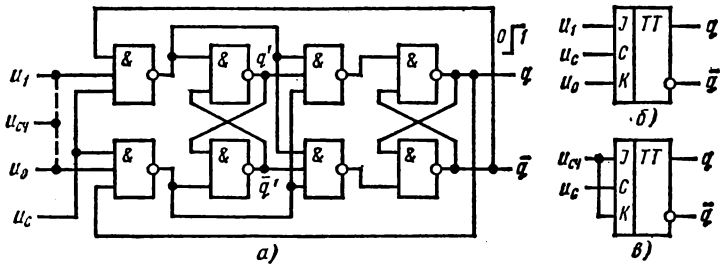


Рис. 4.24. JK-триггер: *а* — вариант функциональной схемы; *б* — условное изображение триггера при раздельном использовании входов; *в* — условное изображение триггера при организации счетного входа

В интегральной схемотехнике широкое применение находят так называемые JK-триггеры. Они имеют раздельные кодовые входы: J — вход установки в единичное состояние и K — в нулевое, особенность которых состоит в том, что при соединении их между собой образуется счетный вход. Пример функциональной схемы JK-триггера дан на рис. 4.24, *а*. Схема состоит из двух одинаковых синхронизируемых RS-триггеров. Выходы q и $\bar{q}$ второй ступени связаны перекрестными обратными связями с входами входных схем *И—НЕ* первой ступени. Инверсные сигналы для управления входами второй ступени снимаются с выходов входных схем *И—НЕ*. Этими сигналами вторая ступень отключается от первой на время переключения последней. Если кодовые входы соединить между собой, как показано на схеме пунктиром, то схема будет работать как Т-триггер. В данном случае каноническое уравнение переключения будет

$$q_{t+1} = \overline{u_0 u_1 q_t} \vee u_0 u_1 \bar{q}_t. \qquad (4.5)$$

Задержка переключения второй ступени превышает длительность синхронизирующего сигнала u_C.

Кроме JK-триггеров, широко применяют двухступенчатые D-триггеры. Они также являются универсальными. Их можно использовать как для временного хранения информации, так и для счета. Вариант функциональной схемы двухступенчатого D-триггера приведен на рис. 4.25, *а*. В первой ступени включен одноступенчатый D-триггер, аналогичный используемому в схеме на рис. 4.17, *а*; во второй ступени — синхронизируемый RS-триггер. В двухступенчатой схеме нет обратных связей с выходов второй ступени на входы логических схем первой. Поэтому в целом она работает как линия задержки кодовых сигналов u_1, воспринимаемых входными цепями в моменты действия синхронизирующих сигналов u_c. Время задержки определяется интервалами следования сигналов u_c и поэтому

равно периоду синхронизации. Рассматриваемый триггер является полностью определенным элементарным автоматом, имеющим полную систему переходов. Каноническое уравнение переходов триггера

$$q_{t+1} = u_1 u_c \vee \bar{u}_c q_t. \tag{4.6}$$

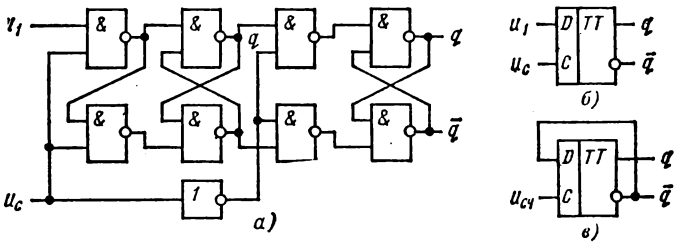


Рис. 4.25. Двухступенчатый D-триггер: *а* — вариант функциональной схемы; *б* — условное изображение D-триггера; *в* — условное изображение D-триггера, включенного по схеме со счетным входом

При подстановке в (4.6) переменной $\bar{q}_t$ вместо u_1 в правой части получается выражение логической операции отрицания равнозначности:

$$q_{t+1} = q_t u_c \vee q_t \bar{u}_c. \tag{4.7}$$

Это означает, что, соединив выход $\bar{q}$ с входом D, получим сумматор по mod 2, или одноразрядный счетчик по отношению к сигналам, подаваемым на вход C. Варианты условных изображений двухступенчатых D-триггеров для случаев использования их в качестве элемента запоминания с однофазным входом и триггера со счетным входом приведены на рис. 4.25, *б* и *в*.

Двухступенчатые триггеры, как и одноступенчатые, могут иметь управляемые входы синхронизации. Такими, например, являются двухступенчатые DV-триггеры и др. Как и в схемах на рис. 4.20, 4.21, для организации управляемых входов синхронизации на входах триггеров синхронизирующие сигналы u_c объединяют с управляющими u_y с помощью специальных входных схем совпадения.

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 про-

цессы 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-1022–ЕС-1050(52), реализуются в пределах отдельных модулей интегральных схем, то функциональные элементы могут не ограничиваться этими пределами. Для их построения может требоваться несколько десятков модулей ИС, как это имеет место, например, в модели ЕС-1050(52). В зависимости от требований к быстродействию схем могут налагаться условия на компоновку модулей, составляющих ФЭ. Например, в модели ЕС-1050(52) любой функциональный элемент должен целиком размещаться на плате одного ТЭЗ.

Функциональные элементы делятся на классы, каждый из которых охватывает группу схем, реализующих логические соотношения определенного типа. В каждой группе ФЭ различаются допустимыми количествами входов и выходов. Для функциональных элементов на простейших ИС типа ЭСЛ 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.

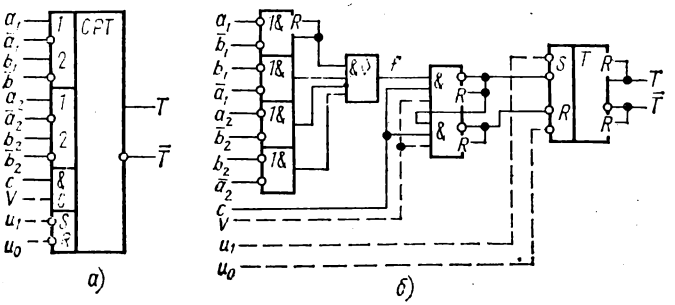


Рис. 4.26. Схема поразрядного сравнения с триггерным выходом: *а* — условное изображение ФЭ; *б* — пример построения ФЭ на базовых элементах

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, 4.10, 4.11. По типу схемы, приведенной на 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. 4.26. При представлении сигналов кода 1 низким уровнем потенциала на выходе f псевдоэлемента сигналы формируются согласно логическому соотношению

$$f = (a_1 b_1 \vee \bar{a}_1 \bar{b}_1)(a_2 b_2 \vee \bar{a}_2 \bar{b}_2). \qquad (4.8)$$

Схема запоминания этих сигналов такая же, как была рассмотрена на рис. 4.21. Как показано на рис. 4.26, *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.

Простейшие примеры двух вариантов сумматора по mod 2 приведены на рис. 4.27. Схема, соответствующая условному обозначению, показанному на рис. 4.27, *а*, 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 P и 1. В результате получается

$$f = \overline{a_1 a_2}(a_1 \vee a_2) = a_1 \bar{a}_2 \vee \bar{a}_1 a_2. \qquad (4.9)$$

Во втором случае (рис. 4.27, *в*) имеет место соотношение

$$f = \overline{\overline{a_1\bar{a}_2 \vee \bar{a}_1 a_2}} = a_1\bar{a}_2 \vee \bar{a}_1 a_2. \qquad (4.10)$$

В наборе функциональных элементов имеются схемы свертки, позволяющие суммировать по mod 2 также 4–5- и 8–9-разрядные двоичные коды.

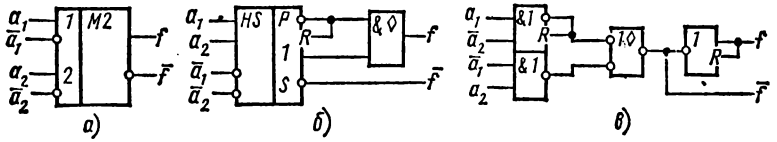


Рис. 4.27. Схема сложения по mod 2: *а* — условное изображение ФЭ; *б* — построение на основе модуля полусумматора; *в* — построение на универсальных логических элементах типа ЭСЛ

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 многовходовые триггерные схемы. Пример триггера с двумя независимыми однофазными синхронизируемыми входами и с двумя параллельно работающими RS-триггерами, аналогичными

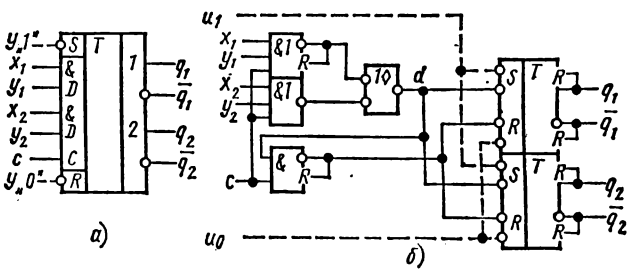


Рис. 4.28. Вариант многовходовой схемы триггера с однофазным запуском: *а* — условное изображение ФЭ; *б* — пример построения на базовых элементах

показанному на рис. 4.19, приведен на рис. 4.28. Здесь каждый из однофазных входов выполнен по схеме, показанной на рис. 4.21, *а*. На входах входных схем *И–НЕ* сигналам 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

$$d = \overline{(x_1y_1 \vee x_2y_2)\,C}. \qquad (4.11)$$

В схеме дополнительно могут быть использованы входы установки триггеров в единичное и нулевое состояния сигналами u_1 и u_0 (штриховые линии).

Среди ФЭ имеются триггеры, способные иметь до 15—25 входных схем совпадения, подобных изображенным на рис. 4.28, *б*.

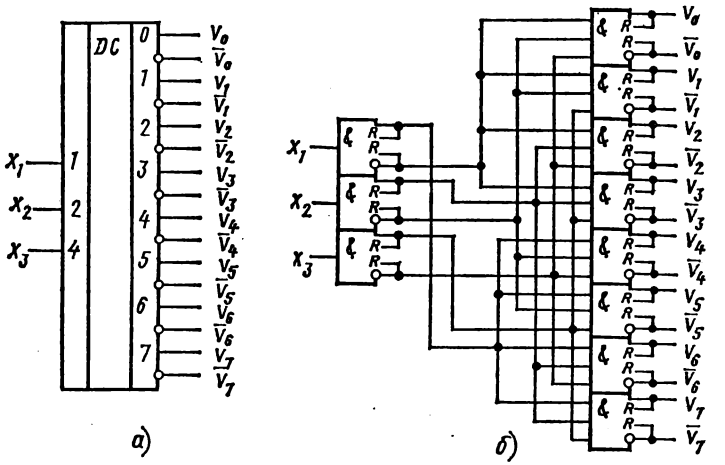


Рис. 4.29. Дешифратор: *а* — условное изображение ФЭ; *б* — пример построения на базовых элементах

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. 4.29. В этой схеме выходные сигналы связаны с входными следующими логическими соотношениями:

$$
\begin{aligned}
v_0 &= \bar{x}_1\bar{x}_2\bar{x}_3; \quad \bar{v}_0 = \overline{\bar{x}_1\bar{x}_2\bar{x}_3};\\
v_1 &= \bar{x}_1\bar{x}_2 x_3; \quad \bar{v}_1 = \overline{\bar{x}_1\bar{x}_2 x_3};\\
v_2 &= \bar{x}_1 x_2\bar{x}_3; \quad \bar{v}_2 = \overline{\bar{x}_1 x_2\bar{x}_3};\\
v_3 &= \bar{x}_1 x_2 x_3; \quad \bar{v}_3 = \overline{\bar{x}_1 x_2 x_3};\\
v_4 &= x_1\bar{x}_2\bar{x}_3; \quad \bar{v}_4 = \overline{x_1\bar{x}_2\bar{x}_3};\\
v_5 &= x_1\bar{x}_2 x_3; \quad \bar{v}_5 = \overline{x_1\bar{x}_2 x_3};\\
v_6 &= x_1 x_2\bar{x}_3; \quad \bar{v}_6 = \overline{x_1 x_2\bar{x}_3};\\
v_7 &= x_1 x_2 x_3; \quad \bar{v}_7 = \overline{x_1 x_2 x_3}.
\end{aligned}
\tag{4.12}
$$

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 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: посылки, запоминания, сдвига кодов, сложения 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 и т. д.

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 кодов и напряжений специальной формы, усилители слабых сигналов и т. п.

На структурных схемах узлы изображают в виде ограниченных отрезков, квадратов, прямоугольников, трапеций и других плоских фигур установленной формы. Внутри указанных фигур даются полные или сокращенные условные названия узлов.

Типовые узлы строятся из логических и запоминающих элементов применяемых комплексов ИС различной степени интеграции. Особенности использования интегральных схем зависят от типа реализуемой схемы узла, 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 простейших ИС 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 вентилей.

Например, как показано на рис. 4.30, *а*, 8-разрядный регистр может принимать 8-разрядные коды с двух направлений. Приемом кодов X

управляют сигналы *V1*, а кодов *Y* – сигналы *V2*. Работа триггеров регистра синхронизируется сигналом *C1* и управляющим сигналом *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. 4.30, *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

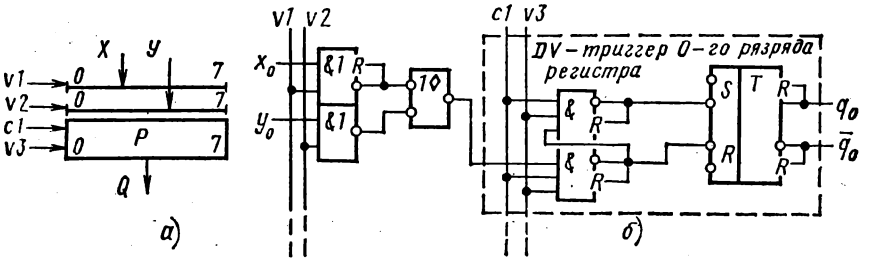


Рис. 4.30. Восьмиразрядный регистр для оперативного запоминания и временного хранения информации: *а* — условное структурное изображение; *б* — функциональная схема

и выходным сигналам кода 1 ставятся в соответствие низкие уровни потенциала. При этом условии работу схемы можно описать уравнением

$$q_{0,t+1} = q_{0,t}\overline{C1}\,\overline{V3} \vee (x_0 V1 \vee y_0 V2)\, C1 V3. \qquad (4.13)$$

Триггер нулевого разряда регистра выполнен по схеме, показанной на рис. 4.21, *а*. Такой регистр обеспечивает хранение информации минимум в течение периода следования синхроимпульсов *C1*. Продолжительность хранения зависит от интервала времени между управляющими сигналами *V3*.

Сдвигающие регистры. Посредством этих регистров, кроме запоминания кодов, осуществляются 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:

„прочесть" 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 запомнить на время, на которое отстоят друг от друга эти синхросигналы. Такое временное запоминание в потенциаль-

ных 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. 4.31, *а*.

Исходный код *X* принимается в регистр *P1* по сигналу ПМ (Прием). В триггеры регистра он заносится по синхросигналу *C1* при наличии разрешающего потенциала *V1*. Для осуществления сдвига принятый код по синхросигналу *C2* засылается для промежуточного хранения в регистр *P2*. Затем из этого регистра он возвращается с необходимым сдвигом вправо или влево в регистр *P1*.

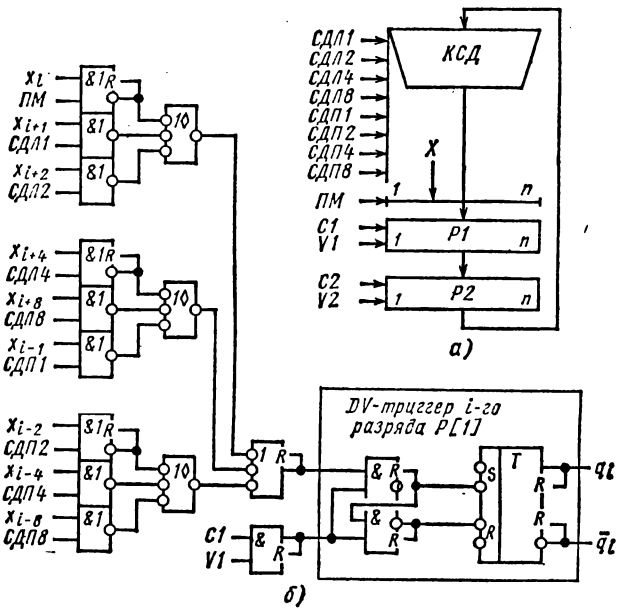


Рис. 4.31. Схема сдвига: *а* — условное структурное изображение; *б* — функциональная схема *i*-го разряда регистра *P2* с входным коммутатором

Собственно сдвиг производится комбинационной схемой сдвига КСД, включенной между регистрами *P2* и *P1*. В рассматриваемом сдвигающем регистре возможны однократные сдвиги влево и вправо на один, два, четыре и восемь разрядов (Л1, Л2, Л4, Л8, П1, П2, П4, П8). Способ построения комбинационной схемы, включенной на входе *P1*, и цепи подачи управляющих сигналов сдвига, опущенные на схеме, приведенной на рис. 4.31, *а*, поясняются функциональной схемой (рис. 4.31, *б*).

На входе триггера *i*-го разряда в общем случае включается схема *И—ИЛИ*, имеющая девять двухвходовых схем совпадения. Первая сверху предназначается для приема значения x_i исходного операнда *X*, остальные — для передачи содержимого триггеров регистра *P2* с соответствующим сдвигом. Сигналам кода 1 в схеме соответствует низкий уровень

потенциала. Построение схемы *И—ИЛИ* аналогично рассмотренному на рис. 4.30, *б*. Элемент памяти регистра также выполнен одновходовым.

Комбинационные схемы коммутации цепей обмена информацией. Эти схемы являются теми узлами, посредством которых осуществляется управление микрооперациями межрегистровых передач, выдача на магистрали и прием с магистралей машинных слов или их частей. В ряде случаев эти схемы оказываются достаточно сложными, требующими для своей реализации значительных аппаратурных затрат. Примером могут служить вентильные схемы межрегистровых передач в сдвигающем регистре, рассмотренном на рис. 4.31.

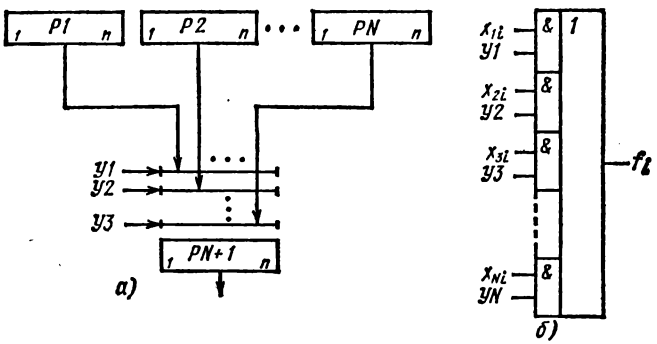


Рис. 4.32. Схема коммутации цепей приема информации с различных направлений: *а* — структурная схема связей между регистрами; *б* — пример функциональной схемы селектора

Цепи коммутации приема информации в регистры с различных направлений используются в различных устройствах машин. В общем случае структура параллельных цепей связи входов одного регистра с выходами N других регистров по обособленным nN шинам может иметь вид, показанный на рис. 4.32, *а*. Здесь коммутация цепей передачи осуществляется с помощью N групп вентилей, управляемых сигналами $У1 - УN$.

Для построения коммутационных вентильных схем используются комбинационные схемы типа *И—ИЛИ* (рис. 4.32, *б*). Их часто называют селекторами. Они реализуются либо на отдельных интегральных элементах *И—НЕ, ИЛИ—НЕ, И—ИЛИ—НЕ*, либо на специальных интегральных схемах, содержащих несколько параллельно работающих селектирующих цепей. В коммутационные магистральные схемы также могут быть включены запоминающие элементы — триггеры. Это обеспечивает возможность расширения по времени передаваемых сигналов, что может быть необходимым, если к цепям передачи подключаются комбинационные схемы преобразования информации, например, типа сумматоров.

Счетчики. При использовании потенциальных элементов такие узлы, предназначаемые 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 ее уменьшения (вычитания). В счетчиках на потенциальных триггерах обычно применяют цепи либо сквозного переноса, либо параллельного.

На рис. 4.33 приведен пример функциональной схемы двух младших разрядов n-разрядного счетчика, реализуемого на элементах типа ЭСЛ. Соответствующее ему условное изображение, применяемое на структурных схемах, дано на рис. 4.33, *б*. Двухступенчатые триггеры выполнены на одновходовых триггерах, однако в целом двухступенчатые схемы работают аналогично рассмотренному на рис. 4.22.

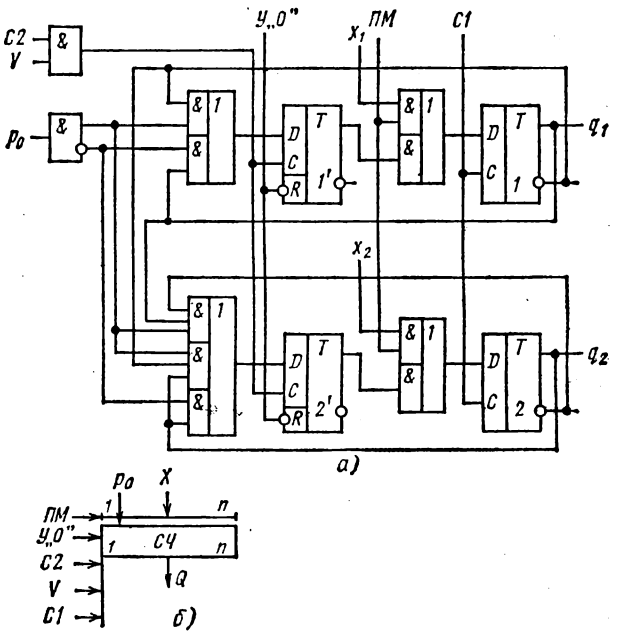


Рис. 4.33. Два разряда счетчика, выполненного на одноступенчатых D-триггерах: *а* — функциональная схема; *б* — условное изображение

Младший разряд счетчика образуют триггеры *1* и *1'*. За исключением сигнала установки в нулевое состояние «У«0», все входные кодовые, синхронизирующие и управляющие сигналы — низкого уровня. Считаемые сигналы, поступающие на вход младшего разряда, обозначены символом p_0. Длительность их должна быть не меньше времени полного срабатывания триггеров первой ступени (на схеме триггеры *1'* и *2'*). Работа ступеней синхронизируется сигналами *С1* и *С2*, сдвинутыми относительно друг друга на полпериода. Воздействие на схему синхронизирующих сигналов серии *С2* разрешается управляющими сигналами *V*.

Общая установка триггеров счетчика в нулевое состояние производится сигналами высокого уровня У«0», подаваемыми непосредственно на входы *R*. Занесение в счетчик исходного числа $X = \ldots x_2 x_1$, с которого начинается счет, осуществляется по сигналам приема ПМ.

Цепь переноса между разрядами выполнена по параллельной схеме. Если триггер *1'* первого разряда функционирует согласно уравнению

$$q_{1',t+1} = (q_{1',t}\bar{p}_0 \vee q_{1',t}p_0)\,C2V, \qquad (4.14)$$

то переключение триггера *2'* второго разряда происходит в соответствии с соотношением

$$q_{2',t+1} = (q_{2',t}\bar{p}_0 \vee q_{2',t}\bar{q}_{1',t}p_0 \vee \bar{q}_{2',t}q_{1',t}p_0)\,C2V. \qquad (4.15)$$

При последовательной подаче на вход сигналов $p_0 = 1$ состояния $q_{2'}q_{1'}$ последовательно будут принимать, например, значения: 00, 01, 10, 11, 00, ... Переход в очередное состояние происходит при совпадении на входе сигналов p_0, *C2* и *V*. После переключения триггеров *1'* и *2'* в очередное состояние по следующему за *C2* сигналу *C1* это же состояние принимают и триггеры *1* и *2*.

Кроме регистров и счетчиков большую роль в схемах машин играют также другие узлы, перечисленные в начале параграфа.

Дешифраторы, схемы сравнения и свертки, во множестве используемые в различных устройствах машин, строятся, например, на элементах типа ЭСЛ 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.

4.6. Интегральные схемы повышенной степени интеграции

Совершенствование ЭВМ ЕС, улучшение их технико-экономических показателей, осуществляются в первую очередь за счет применения более совершенной элементной базы. Достижения в области создания интегральных схем повышенной степени интеграции позволили шире использовать их как в модернизированных моделях ЭВМ, так и во вновь разрабатываемых. В настоящее время интегральные схемы бо́льшего функционального содержания имеются в составах комплексов и ТТЛ-элементов, и элементов типа ЭСЛ.

Все схемы изготовляются по полупроводниковой планарно-эпитаксиальной технологии в монокристаллах кремния с изоляцией компонентов обратно смещенными $p-n$-переходами. Подавляющее большинство интегральных схем повышенной степени интеграции изготовляется в 16-выводных корпусах. В ЭВМ Единой системы используются схемы в пластмассовых корпусах со штырьковыми выводами. Основу логических и запоминающих схем комплексов составляют универсальные логические элементы типа *И–НЕ* (см. рис. 4.2, *а*) для ТТЛ-схем и элементы типа *И/И–НЕ* (см. рис. 4.12) для схем типа ЭСЛ. Наборы схем комплексов

образуют удобные в применении технически полные системы элементов. Например, в комплекс интегральных схем на токовых переключателях (типа ЭСЛ) входят:

универсальные элементарные логические схемы типа *И—НЕ, И/И—НЕ*, различающиеся количествами входов и количествами элементов в корпусах;

триггерные схемы и регистры;

селекторные схемы (типа *И—ИЛИ* для однофазных и парафазных сигналов);

дешифраторы и шифраторы;

мультиплексоры;

схемы сложения по mod 2;

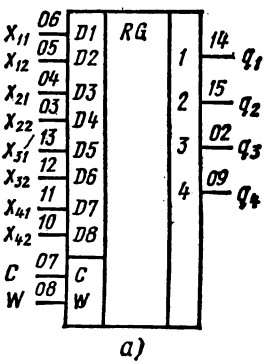
счетчики;

арифметико-логические схемы и схемы ускоренных переносов;

схемы оперативных и постоянных запоминающих устройств;

специальные схемы согласования цепей, усиления и преобразования сигналов, индикации и т. д.

В комплексах ТТЛ-элементов, кроме коммутационных, регистровых, счетчиковых, арифметико-логических и прочих схем, имеются преобразователи чисел из двоично-десятичной формы в двоичную систему счисления и обратно, преобразователи двоичных кодов в коды знаков русского или английского алфавитов, программируемое ПЗУ и т. д. Рассмотрим особенности построения и организации функционирования некоторых из перечисленных выше типов схем.



а)

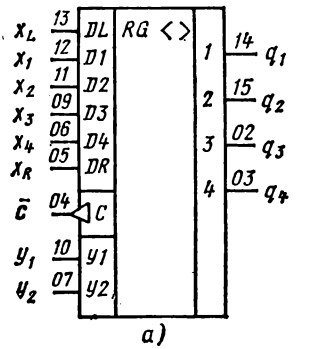
C	W	x_{i1}	x_{i2}	$q_{i,t+1}$
1	1	~	1	1
1	1	~	0	0
1	0	1	~	1
1	0	0	~	0
0	~	~	~	$q_{i,t}$

б)

Рис. 4.34. Четырехразрядный регистр с однофазными входами и синхронизируемым приемом информации: *а* — условное изображение интегральной схемы регистра; *б* — таблица состояний

В комплексе ЭСЛ-элементов для построения регистров имеется, например, интегральная схема 4-разрядного регистра с однофазными входами (рис. 4.34, *а*). Триггеры имеют общую входную цепь *C* для сигналов синхронизации приема информации. В каждый D-триггер информация может приниматься с направлений x_{i1} и x_{i2}. Включением нужного направления управляют сигналы, подаваемые на вход *W*. Работа регистра поясняется таблицей на рис. 4.34, *б*. Сигналы кода 1 в ней соответствуют низким уровням потенциала. Безразличные состояния в этой таблице, как и во всех последующих, обозначены волнистой чертой.

Для построения многоразрядных сдвигающих регистров имеется 4-разрядная схема, изображенная на рис. 4.35, *а*. В этой ИС четыре D-триггера связаны цепями сдвига вправо и влево. Кроме однофазных кодовых входов D_i для ввода в регистр сдвигаемого кода, имеются входы D_L и D_R. Они используются при построении многоразрядных сдвигающих регистров. Работа схемы поясняется таблицей на рис. 4.35, *б*. Сигналами *У1* и *У2* осуществляется переключение режимов работы регистра. Изменение выходных сигналов q_i, отражающих состояния выходных ступеней триггеров регистра, происходит по фронту сигнала $\bar{C}$.



Режим	y_1	y_2	$q_{1,t+1}$	$q_{2,t+1}$	$q_{3,t+1}$	$q_{4,t+1}$
Запись	1	1	x_1	x_2	x_3	x_4
Сдвиг вправо	1	0	$q_{2,t}$	$q_{3,t}$	$q_{4,t}$	x_R
Сдвиг влево	0	1	x_L	$q_{1,t}$	$q_{2,t}$	$q_{3,t}$
Хранение	0	0	$q_{1,t}$	$q_{2,t}$	$q_{3,t}$	$q_{4,t}$

б)

Рис. 4.35. Четырехразрядный сдвигающий регистр с однофазными входами: *а* — условное изображение интегральной схемы регистра; *б* — таблица состояний

Кроме схем регистров, в рассматриваемом комплексе ИС имеются триггерные схемы. В корпусах этих ИС содержатся, например, либо два одноступенчатых, либо два двухступенчатых RSDV-триггера с общей синхронизацией *D*-входов.

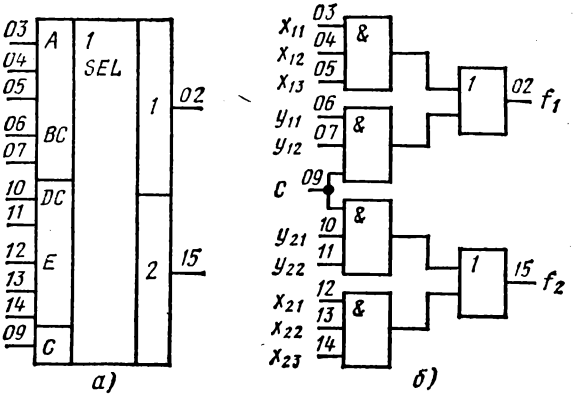
Для упрощения построения коммутационных цепей приема информации в регистры по многим направлениям (см. рис. 4.32, *а*) в комплексе ИС повышенной степени интеграции имеется несколько типов селекторных схем. Пример схемы одного из селекторов дан на рис. 4.36. На токовых переключателях такие схемы *И—ИЛИ* реализуются аналогично рассмотренной на рис. 4.14. Как следует из функциональной схемы (рис. 4.36, *б*), на выходы f_1 и f_2 сигналы передаются либо с направления x, либо — с y, т. е.

$$f_i = x_{i1}x_{i2}x_{i3} \vee y_{i1}y_{i2}C. \qquad (4.16)$$

Входы схем, через которые принимаются сигналы по направлению y, имеют общую синхронизацию.

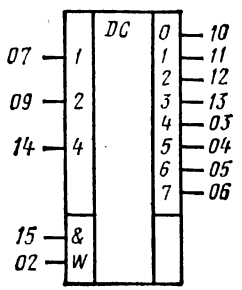
На рис. 4.37 изображены схемы и таблица, поясняющие принцип построения и работы одного из вариантов дешифраторов, помещаемых в 16-выводном корпусе ИС. Данная схема — трехвходовая. Входные однофазные сигналы преобразуются в парафазные, необходимые для реализации дешифратора, непосредственно внутри интегральной схемы (рис. 4.37, *б*). Вход синхронизации в схеме — управляемый.

Рис. 4.36. Селектор типа 2 × (3И — 2ИЛИ): *а* — условное изображение интегральной схемы селектора; *б* — логическая структура



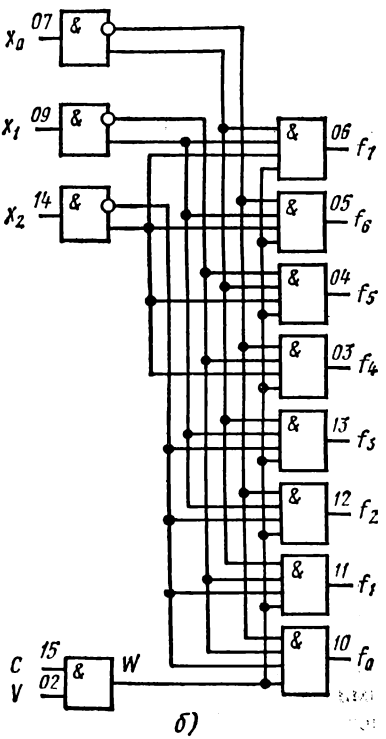
В случаях, когда необходима раздельная коммутация большого количества цепей на одно направление, применяются схемы типа мультиплексоров (рис. 4.38, *а*). Соединение входов ($x_0 - x_7$) с выходом осуществляется здесь встроенной схемой дешифратора, который дешифрирует управляющие коды $Y_0Y_1Y_2$. В каждый очередной момент времени, как следует

Рис. 4.37. Дешифратор с однофазными входами: *а* — условное изображение интегральной схемы дешифратора; *б* — логическая структура; *в* — таблица истинности



C	V	X_0	X_1	X_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
0	0	~	~	~	0	0	0	0	0	0	0	0
1	0	~	~	~	0	0	0	0	0	0	0	0
0	1	~	~	~	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	1	0	0	0	0	0	0
1	1	0	1	0	0	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	1	0	0	0
1	1	1	0	1	0	0	0	0	0	1	0	0
1	1	0	1	1	0	0	0	0	0	0	1	0
1	1	1	1	1	0	0	0	0	0	0	0	1

в)



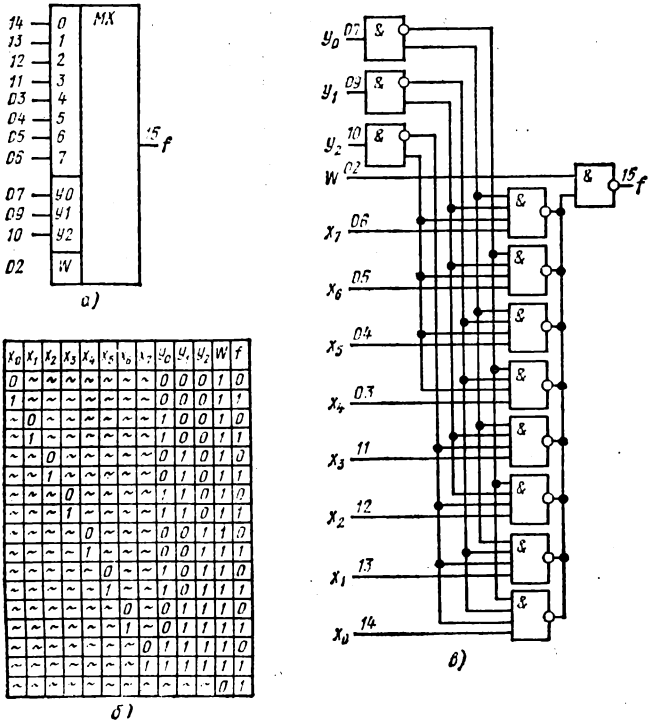


Рис. 4.38. Восьмивходовой мультиплексор: *а* — условное изображение интегральной схемы мультиплексора; *б* — логическая структура; *в* — таблица истинности

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_0	y_1	y_2	W	f
0	~	~	~	~	~	~	~	0	0	0	1	0
1	~	~	~	~	~	~	~	0	0	0	1	1
~	0	~	~	~	~	~	~	1	0	0	1	0
~	1	~	~	~	~	~	~	1	0	0	1	1
~	~	0	~	~	~	~	~	0	1	0	1	0
~	~	1	~	~	~	~	~	0	1	0	1	1
~	~	~	0	~	~	~	~	1	1	0	1	0
~	~	~	1	~	~	~	~	1	1	0	1	1
~	~	~	~	0	~	~	~	0	0	1	1	0
~	~	~	~	1	~	~	~	0	0	1	1	1
~	~	~	~	~	0	~	~	1	0	1	1	0
~	~	~	~	~	1	~	~	1	0	1	1	1
~	~	~	~	~	~	0	~	0	1	1	1	0
~	~	~	~	~	~	1	~	0	1	1	1	1
~	~	~	~	~	~	~	0	1	1	1	1	0
~	~	~	~	~	~	~	1	1	1	1	1	1
~	~	~	~	~	~	~	~	~	~	~	0	1

б)

из таблицы рис. 4.38, *в*, на выходе присутствует сигнал, определяемый соотношением

$$f = \overline{w} \vee w(x_0\overline{y}_0\overline{y}_1\overline{y}_2 \vee x_1 y_0\overline{y}_1\overline{y}_2 \vee \ldots \vee x_7 y_0 y_1 y_2). \qquad (4.17)$$

Для построения схем контроля в наборе комплекса ИС предусмотрены схемы сложения по mod 2. В одном из вариантов ИС три сумматора по mod 2 имеют самостоятельные входы и выходы и поэтому могут использоваться раздельно. В другой интегральной схеме сумматоры образуют пирамидальную схему свертки по mod 2 12-разрядных двоичных кодов (рис. 4.39) согласно соотношению

$$f = \overline{x_0 \oplus x_1 \oplus \ldots \oplus x_{11}}. \qquad (4.18)$$

Интегральные схемы частей таких операционных узлов, как регистры, счетчики, арифметико-логические и т. п., спроектированы таким образом, что они обладают определенной универсальностью. Например, сдвигающий регистр, рассмотренный на рис. 4.35, позволяет осуществлять прием кодов, хранение, сдвиги вправо и влево. На рис. 4.40, *а* приведен пример 4-разрядного двоичного реверсивного счетчика. Схема, смонтированная в одном корпусе ИС, может осуществлять счет в режиме суммирования и вычитания. Переключение режимов работы осуществляется управляющими сигналами $У_1$ и $У_2$. Возможные режимы работы счетчика поясняются таблицами на рис. 4.40, *б*. Как и в схеме на рис. 4.35, переклю-

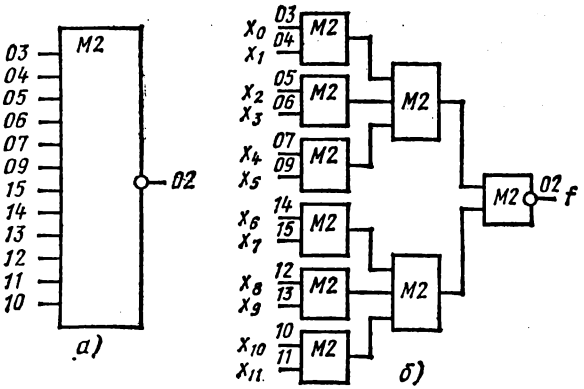
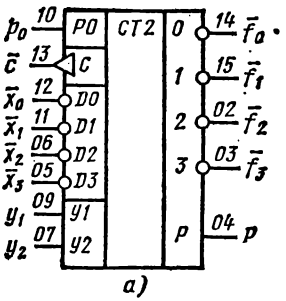


Рис. 4.39. Схема сложения по mod2:
а — условное изображение интегральной схемы;
б — логическая структура

чение выходных сигналов счетчика происходит по фронту синхронизирующего сигнала $\bar{C}$ при его изменении с низкого уровня на высокий.



В таблице $\bar{C}$=0 соответствует переходу 1→0

y_1	y_2	Режим
1	1	Запись
1	0	Прямой счет
0	1	Обратный счет
0	0	Хранение

y_1	y_2	$\bar{x}_0$	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	p_0	$\bar{C}$	$\bar{f}_0$	$\bar{f}_1$	$\bar{f}_2$	$\bar{f}_3$	P
1	1	1	1	0	0	~	0	1	1	0	0	0
1	0	~	~	~	~	1	0	0	1	0	0	0
1	0	~	~	~	~	1	0	1	0	0	0	0
1	0	~	~	~	~	1	0	0	0	0	0	1
1	0	~	~	~	~	0	0	0	0	0	0	0
0	0	~	~	~	~	~	0	0	0	0	0	0
1	1	0	0	1	1	~	0	0	0	1	1	0
0	1	~	~	~	~	1	0	1	0	1	1	0
0	1	~	~	~	~	1	0	0	1	1	1	0
0	1	~	~	~	~	1	0	1	1	1	1	1
0	1	~	~	~	~	1	0	0	0	0	0	0

б)

Рис. 4.40. Реверсивный синхронизируемый четырехразрядный двоичный счетчик: *а* — условное изображение интегральной схемы счетчика; *б* — таблицы режимов работы и состояний

4.7. Универсальные арифметико-логические интегральные схемы

Во многих комплексах ИС, содержащих схемы повышенной степени интеграции, имеются универсальные арифметико-логические схемы (АЛС). Они предназначаются для построения операционных частей ариф-

метико-логических устройств процессоров. Среди них имеются схемы полностью комбинационные и схемы, содержащие триггерные элементы памяти. В ряде моделей ЭВМ ЕС в настоящее время используются комбинационные схемы: 4-разрядная АЛС и в дополнение к ней — схема организации ускоренного межгруппового переноса СМГП. Такие ИС имеются как в комплексах ТТЛ-элементов, так и элементов типа ЭСЛ. По функциональным схемам и реализуемым наборам арифметических, логических и арифметико-логических операций они во многом схожи.

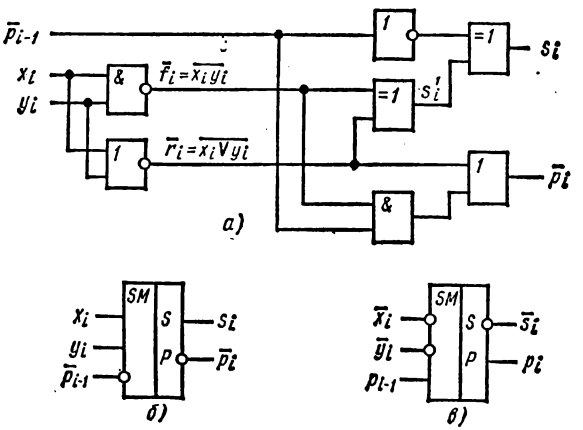


Рис. 4.41. Одноразрядный комбинационный сумматор на элементах *И*, *ИЛИ* и *Исключающее ИЛИ*: *а* — функциональная схема с инверсными цепями входного и выходного переносов; *б* — условное изображение схемы; *в* — условное изображение схемы, двойственной схеме *а*

В основе построения и тех и других схем лежит применение одноразрядных комбинационных сумматоров (рис. 4.41), содержащих два последовательно включенных сумматора по модулю 2 (две схемы *Исключающее ИЛИ*). Такие сумматоры могут работать как при кодовых сигналах высокого уровня, так и при сигналах низкого уровня.

На входы схемы, приведенной на рис. 4.41, *а*, подаются прямые значения x_i, y_i и инверсное значение входного переноса $\bar{p}_{i-1}$. Предположим, что единичным значениям x_i и y_i соответствуют высокие уровни потенциала, а на входе переноса единичному значению соответствует низкий уровень. Входные схемы *И—НЕ* и *ИЛИ—НЕ* формируют сигналы, соответствующие функциям $\bar{f}_i = \overline{x_i y_i}$ и $\bar{r}_i = \overline{x_i \vee y_i}$. По этим сигналам в схеме *Исключающее ИЛИ* вырабатывается полусумма

$$s'_i = \overline{x_i \vee y_i}\,\overline{\overline{x_i y_i}} \vee \overline{\overline{x_i \vee y_i}}\,\overline{x_i y_i} = x_i \bar{y}_i \vee \bar{x}_i y_i. \qquad (4.19)$$

Следующая схема *Исключающее ИЛИ* по s'_i и p_{i-1} формирует окончательное значение суммы

$$s_i = s'_i \bar{p}_{i-1} \vee \bar{s}'_i p_{i-1}. \qquad (4.20)$$

При этом на выходе переноса будет инверсное значение

$$\bar{p}_i = \overline{x_i \vee y_i} \vee \overline{x_i y_i} \bar{p}_i = \bar{x}_i \bar{y}_i \vee \bar{x}_i \bar{p}_{i-1} \vee \bar{y}_i \bar{p}_{i-1}. \qquad (4.21)$$

Все записанные выше функции самодвойственны. Поэтому ту же самую схему можно использовать двойственно — и при инверсных значениях

слагаемых $\overline{x}_i$, $\overline{y}_i$ и при прямом значении переноса p_{i-1}. В этом случае на выходах входных схем *И—НЕ* и *ИЛИ—НЕ* будут сигналы, соответствующие функциям

$$r_i = \overline{\overline{x_i}\,\overline{y_i}} = x_i \vee y_i \tag{4.22}$$

и

$$f_i = \overline{\overline{x_i} \vee \overline{y_i}} = x_i y_i. \tag{4.23}$$

Из них с учетом p_{i-1} две последовательно включенные схемы *Исключающее ИЛИ* сформируют инверсное значение суммы

$$\overline{s}_i = \overline{s_i \overline{p}_{i-1} \vee \overline{s}_i p_{i-1}}. \tag{4.24}$$

Сигнал переноса сформируется согласно соотношению

$$p_i = x_i y_i \vee (x_i \vee y_i)\, p_{i-1}. \tag{4.25}$$

Во втором случае — в двойственной схеме безусловное возникновение переноса определяется логическим произведением $x_i y_i$. Поэтому функцию f_i называют *функцией формирования переноса*. Прохождение через сумматор переноса p_{i-1} зависит от дизъюнкции $x_i \vee y_i$. Вследствие этого функцию r_i называют *функцией распространения переноса*. В первоначальном варианте использования схемы (см. рис. 4.41, *а*) для формирования непереноса $\overline{p}_i$ безусловное значение, наоборот, имеет $\overline{r}_i$, а условное — $\overline{f}_i$.

Функциональная схема 4-разрядной АЛС, реализуемой на ТТЛ-элементах, представлена на рис. 4.42. Она обеспечивает выполнение над кодами $X = x_4x_3x_2x_1$ и $Y = y_4y_3y_2y_1$ 16 поразрядных логических операций и арифметических операций: сложения, вычитания, удвоения (сдвиг влево на один разряд) и др. Перечень выполняемых операций, когда на входах x_i и y_i и выходах z_i сигналам кода 1 соответствуют высокие уровни потенциала, а на входе переноса p_0 — низкие, приведен в табл. 4.5.

Каждый из четырех одноразрядных сумматоров АЛС выполнен по типу схемы на рис. 4.41, *а*. В каждом разряде последовательно включены две схемы *Исключающее ИЛИ*. Логические операции в основном реализуются входными схемами *И—ИЛИ—НЕ* 1—8. Параллельный перенос в пределах четырех разрядов осуществляется с помощью схем *И—ИЛИ—НЕ* 10—12. Схемами *И—НЕ* 13 и 14 и схемой *И—ИЛИ—НЕ* 15 формируются сигналы: R_4^1, соответствующий функции распространения сквозного переноса через 4-разрядную группу, F_4^1 — функции формирования собственного переноса из группы и P_4^1 — полный выходной перенос из группы, определяемый соотношением

$$\overline{P}_4^1 = \overline{F}_4^1 \vee p_0 \overline{R}_4^1. \tag{4.26}$$

Переключение схемы АЛС на выполнение либо логических операций, либо арифметических осуществляется сигналом M. Если этот сигнал высокого уровня ($M = 1$), то низким уровнем потенциала ($\overline{M}$) с выхода входного инвертора закрываются вентили схемы *И—НЕ* 9 и схем *И — ИЛИ — НЕ* 10 — 12 в цепях формирования межразрядных переносов. Таким образом, цепи переносов отключаются. В этом состоянии схема АЛС выполняет только *логические операции*.

Таблица 4.5

Операции, выполняемые АЛС, когда сигналам кода 1 соответствуют высокие уровни потенциала

№ состояния	Набор управляющих сигналов				$M=1$	$M=0$
	$У_4$	$У_3$	$У_2$	$У_1$	Сигнал на выходе z_i соответствует логической функции	Результат арифметической или арифметико-логической операции
0	0	0	0	0	$\overline{x}_i$	$X+p_0$
1	0	0	0	1	$\overline{x_i \vee y_i}$	$(X \vee Y)+p_0$
2	0	0	1	0	$\overline{x}_i y_i$	$(X \vee \neg Y)+p_0$
3	0	0	1	1	0	2^4-1+p_0 (0 при $p_0=1$)
4	0	1	0	0	$\overline{x_i y_i}$	$X+(X \wedge \neg Y)+p_0$
5	0	1	0	1	$\overline{y}_i$	$(X \vee Y)+(X \wedge \neg Y)+p_0$
6	0	1	1	0	$x_i \oplus y_i$	$X-Y-1+p_0$
7	0	1	1	1	$x_i \overline{y}_i$	$(X \wedge \neg Y)-1+p_0$
8	1	0	0	0	$\overline{x_i \vee y_i}$	$X+(X \wedge Y)+p_0$
9	1	0	0	1	$x_i \oplus y_i$	$X+Y+p_0$
10	1	0	1	0	y_i	$(X \vee \neg Y)+(X \wedge Y)+p_0$
11	1	0	1	1	$x_i y_i$	$(X \wedge Y)-1+p_0$
12	1	1	0	0	1	$X+X+p_0$ (СДЛ 1 при $p_0=0$)
13	1	1	0	1	$x_i \vee \overline{y}_i$	$(X \vee Y)+X+p_0$
14	1	1	1	0	$x_i \vee y_i$	$(X \vee \neg Y)+X+p_0$
15	1	1	1	1	x_i	$X-1+p_0$

Коммутация АЛС на выполнение той или иной логической операции производится управляющими сигналами $У_1 - У_4$. В общем случае в каждом из четырех разрядов сигналы $\overline{r}_i^*$ и $\overline{f}_i^*$ на выходах входных схем *И–ИЛИ–НЕ* формируются в соответствии с логическими функциями:

$$\overline{r}_i^* = \overline{x_i \vee y_i У_1 \vee \overline{y}_i У_2}; \tag{4.27}$$

$$\overline{f}_i^* = \overline{x_i \overline{y}_i У_3 \vee x_i y_i У_4}. \tag{4.28}$$

Затем в первой по ходу распространения сигналов схеме *Исключающее ИЛИ* формируется сумма по mod 2:

$$s_i'^* = \overline{r}_i^* \overline{\overline{f}}_i^* \vee \overline{\overline{r}}_i^* \overline{f}_i^* =$$

$$= \overline{x_i \vee y_i У_1 \vee \overline{y}_i У_2}\ \overline{\overline{x_i \overline{y}_i У_3 \vee x_i y_i У_4}} \vee \overline{\overline{x_i \vee y_i У_1 \vee \overline{y}_i У_2}}\ \overline{x_i \overline{y}_i У_3 \vee x_i y_i У_4}. \tag{4.29}$$

В следующей схеме *Исключающее ИЛИ* сигнал, соответствующий $s_i'^*$, инвертируется. Это происходит потому, что при $M=1$ на выходах инверторов схем 9–12 устанавливаются сигналы высокого уровня, соответствующие коду 1. Поэтому суммирование по mod 2 в выходных схемах *Исключающее ИЛИ* дает

$$z_i = s_i'^* \overline{1} \vee \overline{s}_i'^* 1 = \overline{s}_i'^*. \tag{4.30}$$

Пользуясь соотношениями (4.29) и (4.30), можно проанализировать процессы выполнения всех логических операций, перечисленных в табл. 4.5.

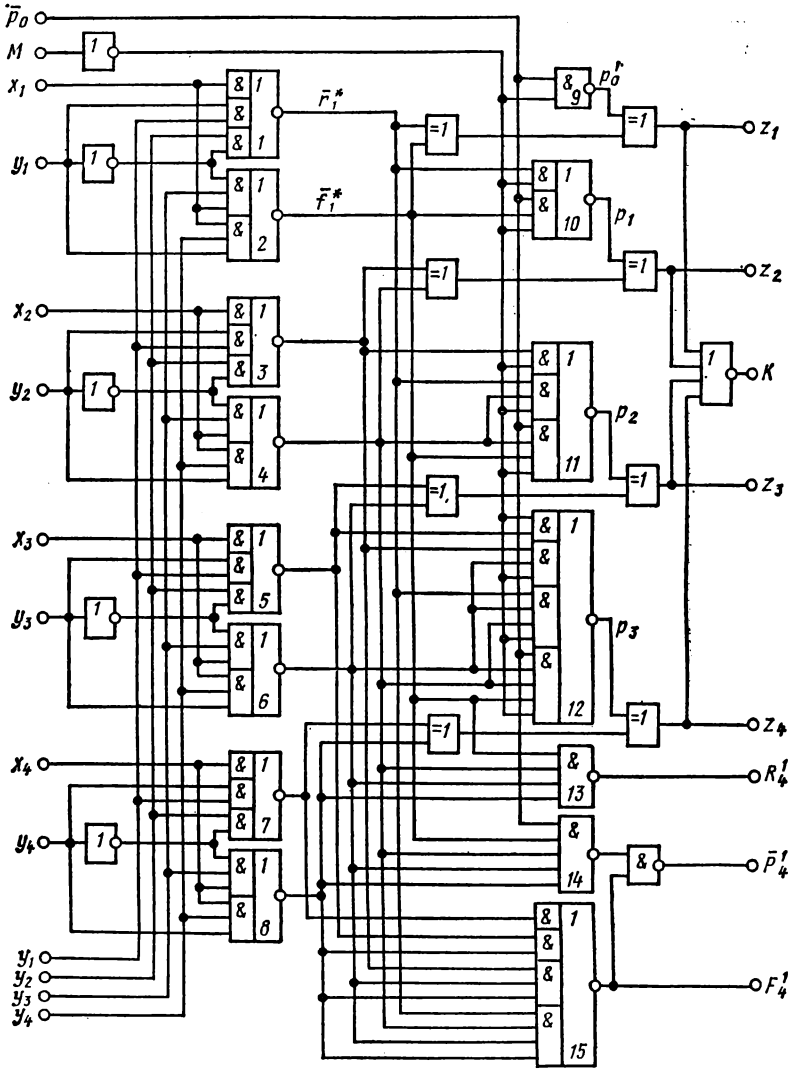


Рис. 4.42. Функциональная схема 4-разрядной АЛС

Арифметические операции схема АЛС выполняет при управляющем сигнале $M = 0$. При этом на выходе входного инвертора в цепи сигнала устанавливается сигнал высокого уровня, соответствующий $\overline{M} = 1$. Он открывает вентили схем *И—ИЛИ—НЕ* в цепях параллельного межразрядного переноса. Поэтому, если в каждом i-м разряде в первом каскаде суммирования по mod 2 согласно (4.29) образуется сумма

$$s_i^* = \bar{r}_i^* \oplus \bar{f}_i^*, \tag{4.31}$$

то во втором каскаде формируется результат, в общем случае соответствующий (4.20), т. е.

$$z_i = s_i'^* \oplus p_{i-1}. \tag{4.32}$$

Содержание арифметического действия определяется набором управляющих сигналов $У_1 - У_4$. Это может быть не только сложение, но и вычитание, удвоение (сдвиг влево) или более сложное действие, включающее к тому же поразрядное логическое преобразование кодов $X = x_4x_3x_2x_1$ и $Y = y_4y_3y_2y_1$. Виды возможных арифметических действий для указанных выше типов сигналов x_i, y_i и p_0 перечислены в табл. 4.5.

Микрооперация сложения кодов $X[1-4]$ и $Y[1-4]$ с учетом входного переноса $p_0[1]$ выполняется при наборе управляющих сигналов $У_4У_3У_2У_1$, равном 1001, т. е. (в записи на структурно-функциональном языке [29])

$$У_4 \wedge \neg| У_3 \wedge \neg| У_2 \wedge У_1) Z := X + Y + p_0, \text{ где } Z[1-4] = z_4z_3z_2z_1.$$

Если $У_2$ и $У_3$ равны нулю, то выражения (4.27) и (4.28) принимают вид

$$\overline{r}_i = \overline{x_i \vee y_i} \tag{4.33}$$

и

$$\overline{f}_i = \overline{x_iy_i}. \tag{4.34}$$

В каждом разряде АЛС первый каскад суммирования по mod 2 дает

$$s_i' = \overline{x_i \vee y_i}\,\overline{x_iy_i} \vee \overline{\overline{x_i \vee y_i}\,\overline{x_iy_i}} = x_i\overline{y} \vee \overline{x}_iy_i. \tag{4.35}$$

Кроме того, на выходах схем 9–12 формируются переносы согласно выражениям

$$p_0' = \overline{\overline{p}_0\overline{M}};$$

$$p_1 = \overline{\overline{r}_1\overline{M} \vee \overline{p}_0\overline{f}\,\overline{M}} = \overline{(\overline{x_1 \vee y_1} \vee \overline{p}_0\overline{x_1y_1})\,\overline{M}};$$

$$p_2 = \overline{\overline{r}_2\overline{M} \vee \overline{r}_1\overline{f}_2\overline{M} \vee \overline{p}_0\overline{f}_1\overline{f}_2\overline{M}} = \overline{(\overline{x_2 \vee y_2} \vee \overline{x_1 \vee y_1}\,\overline{x_2y_2} \vee \overline{p}_0\overline{x_1y_1}\,\overline{x_2y_2})\,\overline{M}};$$

$$p_3 = \overline{\overline{r}_3\overline{M} \vee \overline{r}_2\overline{f}_3\overline{M} \vee \overline{r}_1\overline{f}_2\overline{f}_3\overline{M} \vee \overline{p}_0\overline{f}_1\overline{f}_2\overline{f}_3\overline{M}} =$$

$$= \overline{(\overline{x_3 \vee y_3} \vee \overline{x_2 \vee y_2}\,\overline{x_3y_3} \vee \overline{x_1 \vee y_1}\,\overline{x_2y_2}\,\overline{x_3y_3} \vee \overline{p}_0\overline{x_1y_1}\,\overline{x_2y_2}\,\overline{x_3y_3})\,\overline{M}}. \tag{4.36}$$

После преобразования получим

$$p_0' = p_0 \vee M;$$

$$p_1 = x_1y_1 \vee (x_1 \vee y_1)p_0 \vee M = f_1 \vee p_0r_1 \vee M;$$

$$p_2 = f_2 \vee f_1r_2 \vee p_0r_1r_2 \vee M;$$

$$p_3 = f_3 \vee f_2r_3 \vee f_1r_2r_3 \vee p_0r_1r_2r_3 \vee M, \tag{4.37}$$

где $f_i = x_iy_i$ и $r_i = x_i \vee y_i$. Как следует из (4.37), при $M = 0$ и указанном выше наборе управляющих сигналов схема из элементов 9–12 функционирует как цепь параллельного распространения переносов. Вся схема на рис. 4.42 при наборе управляющих сигналов $У_1 - У_4$, равном 1001, представляет собой двоичный 4-разрядный сумматор с параллельным

переносом. На каждом его выходе формируется значение разряда суммы

$$z_i = s_i' \bar{p}_{i-1} \vee \bar{s}_i' p_{i-1}$$

где s_i' соответствует (4.35), а p_{i-1} – (4.37) при $i = \overline{1,4}$.

Схема АЛС превращается в обычный суммирующий счетчик, если при $M = 0$ на ее входы подать набор управляющих сигналов $У_4У_3У_2У_1$, равный 0000. Иными словами, при $\neg У_4 \wedge \neg У_3 \wedge \neg У_2 \wedge \neg У_1$ реализуется микрооперация $Z := X + p_0$. В этом случае выражения (4.27) и (4.28) вырождаются в $\bar{r}_i^* = \bar{x}_i$ и $f_i^* = 1$. На выходах первых каскадов суммирования по mod 2 в каждом разряде имеет место $s_i' = \bar{x}_i \oplus 1 = x_i$. При $M = 0$ на выходах инверторов схем 9–12 параллельно будут формироваться переносы согласно выражениям

$$\begin{gathered} p_0' = p_0; \\ p_1 = \overline{\bar{x}_1 \vee \bar{p}_0} = x_1 p_0; \\ p_2 = \overline{\bar{x}_2 \vee \bar{x}_1 \vee \bar{p}_0} = x_2 x_1 p_0; \\ p_3 = \overline{\bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1 \vee \bar{p}_0} = x_3 x_2 x_1 p_0. \end{gathered} \tag{4.38}$$

Они соответствуют логическим соотношениям, согласно которым формируются межразрядные переносы в счетчиках с цепями параллельных переносов.

В выходных ступенях суммирования по mod 2 в каждом разряде формируется значение разряда суммы $X + p_0$:

$$z_i = s_i' \oplus p_{i-1} = x_i \bar{p}_{i-1} \vee \bar{x}_i p_{i-1}. \tag{4.39}$$

Формулы для $z_1 - z_4$ из (4.39) с учетом (4.38) получаются в следующем виде:

$$\begin{gathered} z_1 = x_1 \bar{p}_0 \vee \bar{x}_1 p_0; \\ z_2 = x_2 \overline{x_1 p_0} \vee \bar{x}_2 x_1 p_0; \\ z_3 = x_3 \overline{x_2 x_1 p_0} \vee \bar{x}_3 x_2 x_1 p_0; \\ z_4 = x_4 \overline{x_3 x_2 x_1 p_0} \vee \bar{x}_4 x_3 x_2 x_1 p_0. \end{gathered} \tag{4.40}$$

В них в явном виде отражается функционирование параллельных цепей распространения межразрядных переносов.

Схема АЛС может работать и как вычитающий счетчик. Для этого при $M = 0$ следует подать на управляющие входы набор сигналов $У_4У_3У_2У_1$, равный 1111. В этом случае будет выполняться микрооперация: $(У_4 \wedge У_3 \wedge \wedge У_2 \wedge У_1) Z := X - 1 + p_2$. При $p_0 = 1$ число X [1 – 4] не меняется, а при $p_0 = 0$ из него вычитается единица младшего разряда.

Когда на всех управляющих входах $У_1 - У_4$ – сигналы высокого уровня (кода 1), то согласно (4.27) и (4.28) на выходах входных схем *И–ИЛИ–НЕ* (1–8) в каждом разряде будут формироваться сигналы

$$\bar{r}_i^* = \overline{x_i \vee y_i \vee \bar{y}_i} = \overline{x_i \vee 1} = 0; \tag{4.41}$$

$$\bar{f}_i^* = \overline{x_i \bar{y}_i \vee x_i y_i} = \overline{x_i 1} = \bar{x}_i. \tag{4.42}$$

Это равносильно тому, что на кодовые входы схемы АЛС поступают код $X = x_4x_3x_2x_1$ и код $Y = 1111$. Последний код есть дополнение -1 до 2^4, т. е. в двоичной системе $10000 - 0001 = 1111_{(2)}$.

Сложение по mod 16 кода 1111 с кодом X есть не что иное, как вычитание $X - 1$. Если в момент этого вычитания на вход переноса подать $p_0 = 1$, то число останется без изменения, так как $X - 1 + 1 = X$.

Процесс вычитания $X - 1 + p_0$ выполняется следующим образом. В каждом разряде на выходе первой схемы *Исключающее ИЛИ* формируется сигнал, соответствующий

$$s_i' = \bar{r}_i^* \bar{\bar{f}} \vee \bar{\bar{r}}_i^* \bar{f}_i^* = 0x_i \vee 1\bar{x}_i = \bar{x}_i. \tag{4.43}$$

Параллельно вырабатываются переносы:

$$p_0' = p_0;$$

$$p_1 = \overline{\bar{r}_1^* \vee \bar{f}_1^* \bar{p}_0} = \overline{\bar{x}_1 \bar{p}_0} = x_1 \vee p_0;$$

$$p_2 = \overline{\bar{r}_2^* \vee \bar{r}_1^* \bar{f}_2^* \vee \bar{p}_0 \bar{f}_1^* \bar{f}_2^*} = \overline{\bar{p}_0 \bar{x}_1 \bar{x}_2} = x_2 \vee x_1 \vee p_0;$$

$$p_3 = \overline{\bar{r}_3^* \vee \bar{r}_2^* \bar{f}_3^* \vee \bar{r}_1^* \bar{f}_2^* \bar{f}_3^* \vee \bar{p}_0 \bar{f}_1^* \bar{f}_2^* \bar{f}_3^*} = \overline{\bar{p}_0 \bar{x}_1 \bar{x}_2 \bar{x}_3} = x_3 \vee x_2 \vee x_1 \vee p_0. \tag{4.44}$$

В записях формул (4.44) не учтено M, равное 0.

С учетом переносов на выходах z_i формируются сигналы согласно соотношениям

$$z_1 = \bar{x}_1 \bar{p}_0 \vee \bar{\bar{x}}_1 p_0 = \bar{x}_1 \bar{p}_0 \vee x_1 p_0;$$

$$z_2 = \bar{x}_2 \bar{p}_1 \vee \bar{\bar{x}}_2 \bar{p}_1 = \bar{x}_2 \bar{x}_1 \bar{p}_0 \vee x_2 (x_1 \vee p_0);$$

$$z_3 = \bar{x}_3 \bar{p}_2 \vee \bar{\bar{x}}_3 p_2 = \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{p}_0 \vee x_3 (x_2 \vee x_1 \vee p_0);$$

$$z_4 = \bar{x}_4 \bar{p}_3 \vee \bar{\bar{x}}_4 p_3 = \bar{x}_4 \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{p}_0 \vee x_4 (x_3 \vee x_2 \vee x_1 \vee p_0). \tag{4.45}$$

Процесс функционирования вычитающего счетчика согласно соотношениям (4.45) можно иллюстрировать примерами:

$X[1-4]$	0111	0111	0110	0110	0000
-1	$+1111$	$+1111$	$+1111$	$+1111$	$+1111$
$p_0[1]$	1	0	1	0	0 и т. д.
$Z[1-4]$	0111	0110	0110	0101	1111

В рассматриваемой АЛС может осуществляться непосредственное вычитание $X - Y$. Согласно табл. 4.5 для этого при $M = 0$ на управляющие входы следует подать набор сигналов $У_4У_3У_2У_1$, равный 0110.

Возможность реализации непосредственного вычитания на том же оборудовании, на котором выполнен сумматор, основана на аналогии формирования сигналов заема из старших разрядов уменьшаемого при определении разности и сигналов переноса при сложении. Для уяснения этого следует рассмотреть табл. 4.6, соответствующую процессу вычитания из цифры уменьшаемого x_i цифры вычитаемого y_i при наличии или отсутствии сигнала заема d_{i-1} единицы из x_i, поступающего из $(i-1)$-го разряда.

Таблица 4.6

Таблица истинности для процесса определения разности i-х разрядов двоичных чисел

Цифра уменьшаемого x_i	Цифра вычитаемого y_i	Заем 1 из x_i (d_{i-1})	Цифра разности z_{ip}	Заем 1 из следующего разряда уменьшаемого (d_i)	Цифра уменьшаемого x_i	Цифра вычитаемого y_i	Заем 1 из x_i (d_{i-1})	Цифра разности z_{ip}	Заем 1 из следующего разряда уменьшаемого (d_i)
0	0	0	0	0	1	0	0	1	0
0	0	1	1	1	1	0	1	0	0
0	1	0	1	1	1	1	0	0	0
0	1	1	0	1	1	1	1	1	1

Логические функции для z_{ip} и d_i, соответствующие табл. 4.6, в совершенной дизъюнктивной форме имеют вид

$$z_{ip} = \overline{x}_i\overline{y}_i d_{i-1} \vee \overline{x}_i y_i \overline{d}_{i-1} \vee x_i\overline{y}_i\overline{d}_{i-1} \vee x_i y_i d_{i-1}; \tag{4.46}$$

$$d_i = \overline{x}_i\overline{y}_i d_{i-1} \vee \overline{x}_i y_i \overline{d}_{i-1} \vee \overline{x}_i y_i d_{i-1} \vee x_i y_i d_{i-1}. \tag{4.47}$$

Эти функции можно преобразовать следующим образом:

$$z_{ip} = (x_i\overline{y}_i \vee \overline{x}_i y_i)\,\overline{d}_{i-1} \vee (\overline{x}_i\overline{y}_i \vee x_i y_i)\,d_{i-1}; \tag{4.48}$$

$$d_i = \overline{x}_i y_i \vee \overline{x}_i d_{i-1} \vee y_i d_{i-1} = \overline{x}_i y_i \vee (\overline{x}_i \vee y_i)\,d_{i-1}. \tag{4.49}$$

Функции z_{ip} и d_i – самодвойственные, т. е. справедливы соотношения

$$\overline{z}_{ip} = (\overline{x}_i y_i \vee x_i\overline{y}_i)\,d_{i-1} \vee (x_i y_i \vee \overline{x}_i\overline{y}_i)\,\overline{d}_{i-1}; \tag{4.50}$$

$$\overline{d}_i = x_i\overline{y}_i \vee (x_i \vee \overline{y}_i)\,\overline{d}_{i-1}. \tag{4.51}$$

Если в (4.20) подставить значение s'_i, соответствующее (4.19), и сравнить полученное выражение с (4.48), то станет очевидно, что по структуре обе формулы одинаковы. Различие состоит в том, что в (4.48) вместо переноса p_{i-1} присутствует цифра заема d_{i-1}. Поэтому процесс нахождения цифры разности z_{ip}, как и суммы s_i, можно также реализовать с помощью схемы, состоящей из двух последовательно соединенных элементов *Исключающее ИЛИ*. Формула (4.49) отличается от (4.25) также цифрой заема d_{i-1} и, кроме того, наличием инверсий переменной. Для определения заема d_i, как и переноса p_i, требуются две схемы: *И* и *ИЛИ*. Однако в отличие от цепи переноса они должны вырабатывать сигналы, соответствующие функциям *формирования* $f_з = \overline{x}_i y_i$ и *распространения* $r_з = \overline{x}_i \vee y_i$ заема.

В рассматриваемой АЛС в каждом разряде эти сигналы формируются входными схемами *И–ИЛИ–НЕ* 1–8. При наборе управляющих сигналов $У_4У_3У_2У_1$, равном 0110, согласно (4.27) и (4.28) на выходах этих схем имеют место сигналы, соответствующие $\overline{r}_i^* = \overline{x_i \vee y_i} = \overline{x}_i y_i$ и $\overline{f}_i^* = \overline{x_i y_i} = \overline{x}_i \vee y_i$. В первых по ходу распространения сигналов схемах *Исключающее ИЛИ* они преобразуются согласно следующим выражениям:

$$s''_i = \overline{\overline{r}_i^*\overline{f}_i^*} \vee \overline{\overline{r}_i^*\overline{f}_i^*} = x_i y_i \vee \overline{x}_i\overline{y}_i. \tag{4.52}$$

По $\overline{r}_i^*$, $\overline{f}_i^*$ и $\overline{p}_0$ схемами 9–12 вырабатываются сигналы заема. Поэтому для данного случая выходные сигналы p_0', p_1, p_2 и p_3 обозначим через d_0', d_1, d_2 и d_3.

При анализе работы АЛС в режиме вычитания следует иметь в виду, что в ней реализуется микрооперация $\overline{\ }|\, Y_4 \wedge Y_3 \wedge Y_2 \wedge \overline{\ }|\, Y_1)\, Z := X - Y - 1 + p_0$. Заем из младшего разряда разности происходит при $p_0 = 0$. Поэтому сигналы заемов d_0', $d_1 - d_3$ формируются согласно выражениям

$$d_0' = \overline{\overline{p}_0} = p_0;$$

$$d_1 = \overline{\overline{r}_1^* \vee \overline{p}_0 \overline{f}_1^*} = \overline{\overline{x_1 \vee \overline{y}_1} \vee \overline{p}_0 \overline{x_1 \overline{y}_1}} = x_1 \overline{y}_1 \vee (x_1 \vee \overline{y}_1)\, p_0;$$

$$d_2 = \overline{\overline{r}_2^* \vee \overline{r}_1^* \overline{f}_2^* \vee \overline{p}_0 \overline{f}_1^* \overline{f}_2^*} = \overline{\overline{x_2 \vee \overline{y}_2} \vee \overline{x_1 \vee \overline{y}_1 x_2 \overline{y}_2} \vee \overline{p}_0 \overline{x_1 \overline{y}_1 x_2 \overline{y}_2}} =$$

$$= x_2 \overline{y}_2 \vee (x_2 \vee \overline{y}_2)\, x_1 \overline{y}_1 \vee (x_2 \vee \overline{y}_2)(x_1 \vee \overline{y}_1)\, p_0;$$

$$d_3 = \overline{\overline{r}_3^* \vee \overline{r}_2^* \overline{f}_3^* \vee \overline{r}_1^* \overline{f}_2^* \overline{f}_3^* \vee \overline{p}_0 \overline{f}_1^* \overline{f}_2^* \overline{f}_3^*} =$$

$$= \overline{\overline{x_3 \vee \overline{y}_3} \vee \overline{x_2 \vee \overline{y}_2 x_3 \overline{y}_3} \vee \overline{x_1 \vee \overline{y}_1 x_2 \overline{y}_2 x_1 \overline{y}_1} \vee \overline{p}_0 \overline{x_1 \overline{y}_1 x_2 \overline{y}_2 x_3 \overline{y}_3}} =$$

$$= x_3 \overline{y}_3 \vee (x_3 \vee \overline{y}_3)\, x_2 \overline{y}_2 \vee (x_3 \vee \overline{y}_3)(x_2 \vee \overline{y}_2)\, x_1 \overline{y}_1 \vee$$

$$\vee\, (x_3 \vee \overline{y}_3)(x_2 \vee \overline{y}_2)(x_1 \vee \overline{y}_1)\, p_0. \qquad (4.53)$$

В записях формул (4.53) опущено значение $M = 0$. При его учете, а также вводя обозначения f_i^* и r_i^*, эти логические соотношения можно представить в виде, аналогичном (4.37).

По сигналам, соответствующим выражениям (4.52) и (4.53), выходными схемами *Исключающее ИЛИ* вырабатываются сигналы

$$z_i = s_i'' \overline{d}_{i-1} \vee \overline{s}_i'' d_{i-1}, \qquad (4.54)$$

где $i = \overline{1,4}$. Подставив в (4.54) значения s_i'' и d_{i-1} из (4.52) и (4.53), можно получить развернутые выражения для z_i. При проверке их справедливости следует не забывать, что дополнительное вычитание единицы из разности $X - Y$ происходит в случае $p_0 = 0$.

Через АЛС можно пропускать коды как без сдвига, так и со сдвигом влево на один разряд (см. табл. 4.5). Микрооперация сдвига осуществляется при $M = 0$, наборе управляющих сигналов $Y_4 Y_3 Y_2 Y_1$, равном 1100, и $p_0 = 0$. В этом случае согласно (4.27) и (4.28) формируются $\overline{r}_i^* = \overline{x}_i$ и $f_i^* = \overline{x}_i$. Первое суммирование по mod 2 дает $s_i' = 0$. На выходах цепей межразрядных переносов вырабатываются сигналы:

$$p_0' = 0; \quad p_2 = x_2;$$

$$p_1 = x_1; \quad p_3 = x_3.$$

В результате их суммирования с нулями по mod 2 в выходных схемах *Исключающее ИЛИ* на выходах АЛС получается код $Z = z_4 z_3 z_2 z_1 = x_3 x_2 x_1 0$. Сигнал, соответствующий значению x_4, выдается через выход группового переноса $\overline{P}_4$.

Если $p_0 = 1$, то к сдвинутому коду добавляется единица. Таким образом, в общем случае реализуется микрооперация

$$Y_4 \wedge Y_3 \wedge \overline{\ }|\, Y_2 \wedge \overline{\ }|\, Y_1)\, Z := X\,[1-3].0 + p_0.$$

Аналогично рассмотренному можно проанализировать процессы выполнения остальных операций. При этом следует иметь в виду, что АЛС может функционировать и в инверсном режиме, когда на входах x_i и y_i сигналам кода 1 ставятся в соответствие низкие уровни потенциала, а на входе и выходе переноса (p_0 и P_4) – высокие. Выполняемые при этом арифметические, арифметико-логические и логические операции перечислены в табл. 4.7.

Таблица 4.7

Операции, выполняемые АЛС, когда сигналам кода 1 соответствуют низкие уровни потенциала

№ состояния	Набор управляющих сигналов				$M=1$	$M=0$
	y_4	y_3	y_2	y_1	Сигнал на выходе z_i соответствует логической функции	Результат арифметической или арифметико-логической операции
0	0	0	0	0	$\overline{x_i}$	$X-1+p_0$
1	0	0	0	1	$\overline{x_i y_i}$	$(X \wedge Y)-1+p_0$
2	0	0	1	0	$\overline{x_i} \vee y_i$	$(X \wedge \overline{\Pi} Y)-1+p_0$
3	0	0	1	1	1	2^4-1+p_0 (0 при $p_0=1$)
4	0	1	0	0	$\overline{x_i \vee y_i}$	$X+(X \vee \overline{\Pi} Y)+p_0$
5	0	1	0	1	$\overline{y_i}$	$(X \wedge Y)+(X \vee \overline{\Pi} Y)+p_0$
6	0	1	1	0	$\overline{x_i \oplus y_i}$	$X-Y-1+p_0$
7	0	1	1	1	$x_i \vee \overline{y_i}$	$(X \vee \overline{\Pi} Y)+p_0$
8	1	0	0	0	$\overline{x_i} y_i$	$X+(X \vee Y)+p_0$
9	1	0	0	1	$x_i \oplus y_i$	$X+Y+p_0$
10	1	0	1	0	y_i	$(X \wedge \Pi Y)+(X \vee Y)+p_0$
11	1	0	1	1	$x_i \vee y_i$	$(X \vee Y)+p_0$
12	1	1	0	0	0	$X+X+p_0$ (СДЛ 1 при $p_0=0$)
13	1	1	0	1	$x_i \overline{y_i}$	$(X \wedge Y)+X+p_0$
14	1	1	1	0	$x_i y_i$	$(X \wedge \Pi Y)+X+p_0$
15	1	1	1	1	x_i	$X+p_0$

Принятое условное обобщенное изображение 4-разрядной АЛС (ALU) дано на рис. 4.43. Схема на рис. 4.43, *а* соответствует случаю, когда на разрядных входах сигналу кода 1 ставится в соответствие высокий уровень потенциала, а на входе и выходе переноса – низкий. На рис. 4.43, *б* дано условное изображение АЛС для кодовых сигналов и сигналов переноса, инверсных по отношению к предыдущим. На обеих схемах также даны условные изображения схем ускоренного межгруппового переноса (GRP). Их входы и выходы также обозначаются в зависимости от того, какие уровни сигналов переноса ставятся в соответствие коду 1.

Для того чтобы можно было пользоваться схемами ускоренного переноса, в АЛС предусмотрены выходы F_4^1 и R_4^1 (см. рис. 4.42). На этих выходах формируются сигналы согласно логическим соотношениям

$$F_4^1 = \overline{\overline{r}_4 \vee \overline{r}_3 \overline{f}_4 \vee \overline{r}_2 \overline{f}_3 f_4 \vee \overline{r}_1 \overline{f}_2 \overline{f}_3 \overline{f}_4}; \qquad (4.55)$$

$$R_4^1 = \overline{\overline{f}_1 \overline{f}_2 \overline{f}_3 \overline{f}_4}, \qquad (4.56)$$

где $r_i = \overline{x_i \vee y_i}$ и $f_i = \overline{x_i y_i}$, $i = \overline{1,4}$. Действие цепей межгруппового переноса рассматриваем на примере операции сложения, когда $M = 0$, набор управляющих сигналов $У_4У_3У_2У_1$ соответствует 1001 и в цепях переноса значение 1 представляется потенциалом низкого уровня.

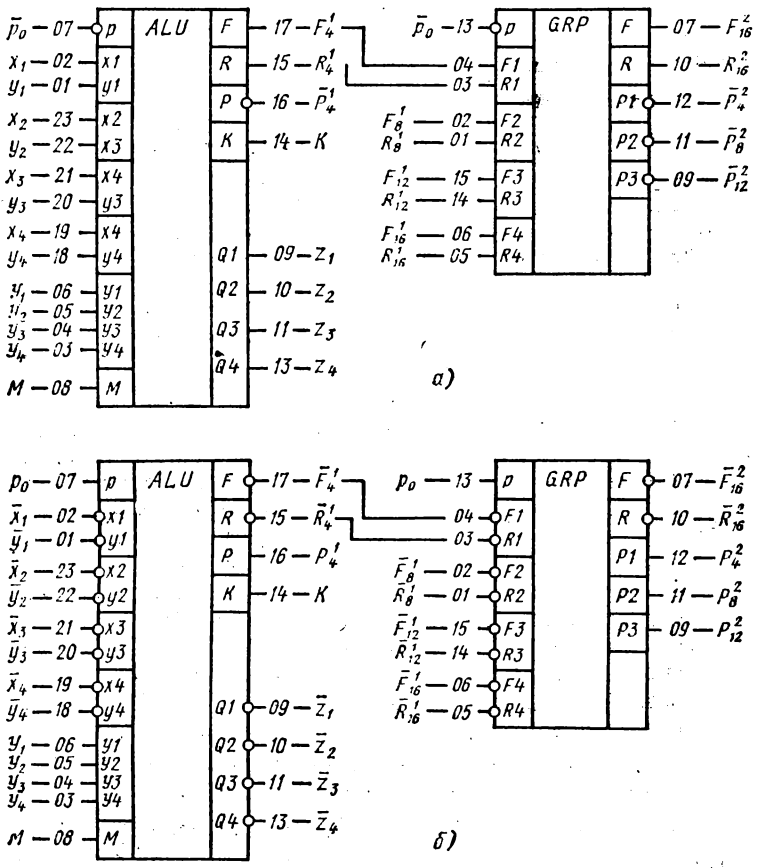


Рис. 4.43. Условные изображения схем АЛС и ускоренного межгруппового переноса: *а* — для кодовых разрядных сигналов высокого уровня; *б* — для кодовых разрядных сигналов низкого уровня

Сигналы F_4^1 и $\overline{\overline{p}_0 \overline{R}_4^1}$ в АЛС объединяются схемой *И—НЕ*. На ее выходе формируется сигнал в соответствии с соотношением (4.26), т. е.

$$\overline{P}_4^1 = \overline{F_4^1 \overline{\overline{p}_0 \overline{R}_4^1}} = \overline{F}_4^1 \vee \overline{p}_0 \overline{R}_4^1.$$

В силу самодвойственности функции переноса это выражение инверсно прямой функции параллельного распространения переноса.

Пользуясь выходами АЛС, на которых формируются сигналы $\overline{P}_4^1$ (см. рис. 4.43, *а*), можно организовывать последовательные межгрупповые переносы. Через эти выходы также передаются сигналы между группами при сдвигах в сторону старших разрядов. Для осуществления параллельных межгрупповых переносов сигналы F_4^1 и R_4^1 со схем АЛС подаются на входы схемы ускоренного переноса (см. рис. 4.43, *а*). Функциональная схема формирования параллельного межгруппового переноса приведена на рис. 4.44.

На трех ее выходах параллельно вырабатываются переносы:

$$\overline{P}_4^2 = \overline{R_4^1 F_4^1 \vee p_0 F_4^1} = F_4^1 \vee \overline{p}_0 \overline{R}_4^1;$$

$$\overline{P}_8^2 = \overline{R_8^1 F_8^1 \vee R_4^1 F_4^1 F_8^1 \vee p_0 F_4^1 F_8^1} = \overline{F}_8^1 \vee \overline{F}_4^1 \overline{R}_8^1 \vee \overline{p}_0 \overline{R}_4^1 \overline{R}_8^1;$$

$$\overline{P}_{12}^2 = \overline{R_{12}^1 F_{12}^1 \vee R_8^1 F_8^1 F_{12}^1 \vee R_4^1 F_4^1 F_8^1 F_{12}^1 \vee p_0 F_4^1 F_8^1 F_{12}^1} =$$

$$= \overline{F}_{12}^1 \vee \overline{F}_8^1 \overline{R}_{12}^1 \vee \overline{F}_4^1 \overline{R}_8^1 \overline{R}_{12}^1 \vee \overline{p}_0 \overline{R}_4^1 \overline{R}_8^1 \overline{R}_{12}^1. \qquad (4.57)$$

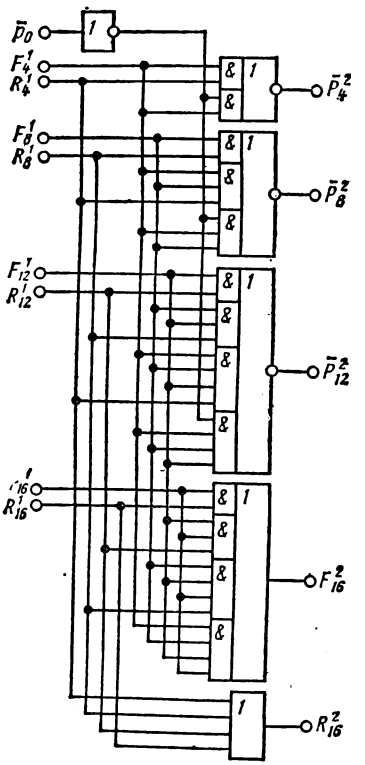


Рис. 4.44. Схема ускоренных межгрупповых переносов

В (4.57) первое выражение аналогично (4.26).

Если необходима организация третьего и т. д. каскадов переносов, то в рассматриваемой схеме для этой цели имеются выходы, на которых формируются сигналы

$$F_{16}^2 = R_{16}^1 F_{16}^1 \vee R_{12}^1 F_{16}^1 \vee R_8^1 F_8^1 F_{12}^1 F_{16}^1 \vee$$
$$\vee F_4^1 F_8^1 F_{12}^1 F_{16}^1$$

и

$$R_{16}^2 = R_4^1 \vee R_8^1 \vee R_{12}^1 \vee R_{16}^1, \qquad (4.58)$$

соответствующие инверсным функциям формирования и распространения переноса.

Примеры построения сумматоров с числом разрядов, большим четырех, даны на рис. 4.45. Схема на рис. 4.45, *а* иллюстрирует построение 8-разрядного сумматора с последовательным межгрупповым переносом. В изображенных схемах АЛС показаны только входы и выходы переносов, соответствующие рис. 4.43, *а*.

На рис. 4.45, *б* приведен пример построения 16-разрядного сумматора с параллельными межгрупповыми переносами. Здесь в условных изображениях сумматоров и схемы ускоренного переноса также показаны только входы и выходы переносов. Сокращенное условное изображение 16-разрядного сумматора (рис. 4.45, *б*) дано на рис. 4.45, *в*.

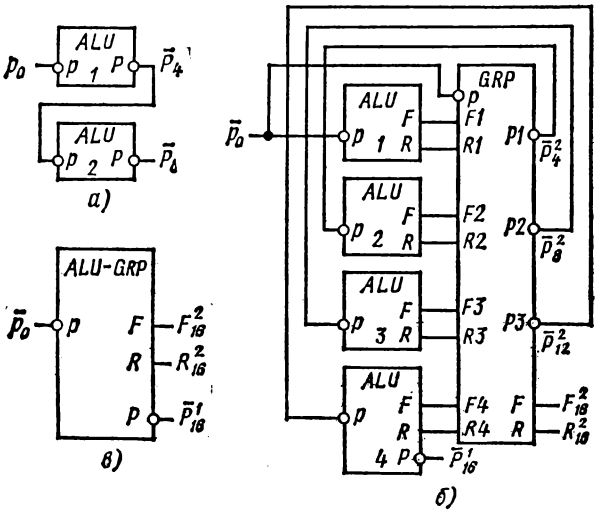


Рис. 4.45. Схемы соединения входов и выходов переносов: *а* — в сумматоре с последовательными межгрупповыми переносами; *б* — в 16-разрядном сумматоре с каскадно-параллельными переносами; *в* — условное сокращенное обозначение схемы *б*

В схеме АЛС (см. рис. 4.42) имеется еще один выход, обозначенный буквой K. На этом выходе формируется сигнал

$$K = \overline{z_1 \vee z_2 \vee z_3 \vee z_4}. \qquad (4.59)$$

Он используется в качестве индикатора равенства сравниваемых кодов $X[1-4]$ и $Y[1-4]$. Сравнение осуществляется в режиме вычитания, когда $M = 0$ и на управляющих входах $У_4У_3У_2У_1$ — комбинация 0110. Если на разрядных входах и выходах сигналам кода 1 ставятся в соответствие высокие уровни потенциала, то при $X = Y$ на выходах $z_1 - z_4$ при вычитании будут потенциалы низкого уровня. В этом случае на выходе K будет высокий уровень потенциала.

Выход инвертора схемы индикатора равенства кодов построен по схеме «с открытым коллектором». Это дает возможность посредством монтажных перемычек между выходами K схем АЛС увеличивать количество входов индицирующих схем *ИЛИ—НЕ*. Таким образом, реализуются схемы анализа на нуль выходов сумматора для количества разрядов, большего четырех.

Рассмотренная схема АЛС выполняется в 24-выводном корпусе.

Глава 5

ОПЕРАТИВНАЯ И ПОСТОЯННАЯ ПАМЯТЬ

5.1. Принципы построения оперативной памяти типа 2,5 *D*

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 3*D* (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 полутоков;

б) двухмерные (двухкоординатные) МОЗУ, или МОЗУ типа 2*D*, иначе называемые МОЗУ с линейной выборкой, МОЗУ типа *Z*; в таких устройствах запись производится по принципу совпадения полутоков, а считывание осуществляется полным током;

в) 2,5-мерные МОЗУ, или МОЗУ типа 2,5*D*, 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 [53]. Рассмотрим сначала общие принципы построения МОЗУ типа 2,5*D*. Это удобнее сделать на конкретном примере построения МОЗУ небольшой емкости [28].

Пусть необходимо построить схему МОЗУ типа 2,5*D* для хранения 32 трехразрядных чисел ($E = 32$, $n = 3$). В трехмерном МОЗУ такой емкости адресные шины по каждой координате проходили бы через три сердечника, а разрядные шины и шины запрета – через 32 сердечника. В двухмерном МОЗУ каждая числовая шина (шина z) проходила бы через три 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 $k = \sqrt{(E/n)}$ и затем округляется до значения 2^p (p – целое число), что необходимо для удобства дешифрации.

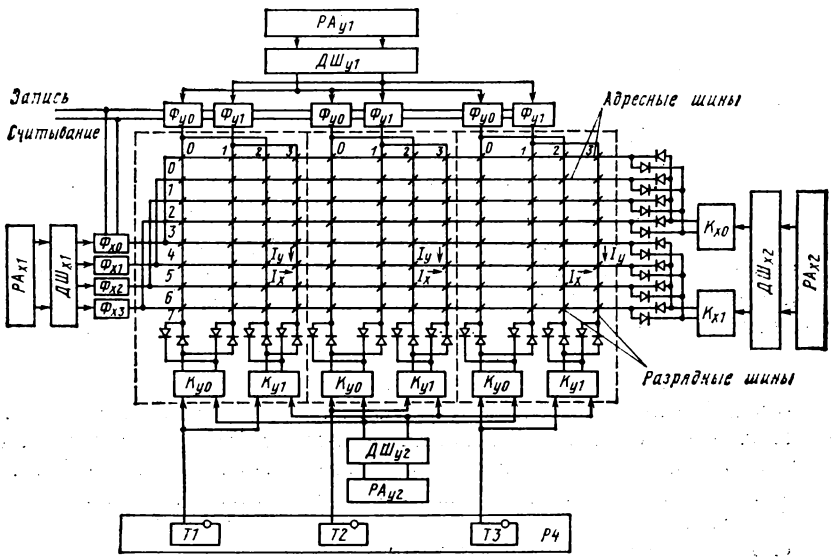


Рис. 5.1. МОЗУ типа 2,5 D малой емкости

В нашем примере $k = \sqrt{(32/3)} \approx 3{,}3$ 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. Число сердечников, прошитых каждой адресной шиной, будет точно соответствовать числу сердечников, через которые проходит каждая разрядная шина, если имеет место равенство

$$\frac{E}{n} = 2^{2p}. \qquad (5.1)$$

Например, если $E = 8192$, $n = 32$, то $p = 4$, и каждая адресная и разрядная шины 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. Прямоугольную матрицу, составленную 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. На рисунке сердечники одной ячейки памяти обведены кружками.

2. Регистр адреса РА, составленный из четырех частей, обозначенных $РА_{x1}$, $РА_{x2}$, $РА_{y1}$, $РА_{y2}$. Число разрядов регистра РА равно 5, так как $E = 32$. Регистры $РА_{x2}$, $РА_{y1}$, $РА_{y2}$ являются одноразрядными, а регистр $РА_{x1}$ – 2-разрядным.

3. Дешифраторы адреса $ДШ_{x1}$, $ДШ_{x2}$, $ДШ_{y1}$, $ДШ_{y2}$. В рассматриваемой схеме дешифраторы $ДШ_{x2}$, $ДШ_{y1}$, $ДШ_{y2}$, связанные с одноразрядными регистрами, включены для общности, так как управление работой соответствующих ключей и формирователей может осуществляться непосредственно сигналами, поступающими с регистров. В МОЗУ большой емкости эти дешифраторы обязательны. Например, для указанного выше примера, когда $E = 8192$ 32-разрядных числа, каждый из дешифраторов $ДШ_{x2}$, $ДШ_{y1}$, $ДШ_{y2}$ 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 $РА_{x1}$, $РА_{x2}$, $РА_{y1}$, $РА_{y2}$ 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 полутоков используется не только при записи, но и при считывании.

При считывании числа, записанного в выделенной ячейке памяти, в результате дешифрации адреса числа возбуждаются формирователи $Ф_{x1}$, $Ф_{y1}$ и открываются ключи $К_{x1}К_{y1}$. Через пятую адресную и третью разрядные шины всех разрядов проходят импульсы токов I_{x1} и I_{y1}, амплитуда которых равна $0{,}5I_m$ (направление токов показано стрелками), где I_m – ток, обеспечивающий создание перемагничивающего поля. Если в избранном сердечнике хранился код 1, то под действием магнитного поля, создаваемого токами I_x и I_y, он перемагничивается в нулевое

положение, и в выходной обмотке (на рис. 5.1 она не показана), которая проходит через все сердечники одного разряда, наводится ЭДС считывания. Если в избранном сердечнике хранился код 0, его магнитное состояние не изменяется, и в идеальном случае в выходной обмотке ЭДС 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 $К_y$, 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:

$$I_{x\,\min} + I_{y\,\min} \approx 2I_{x\,\min} \geqslant I_m; \tag{5.2}$$

$$I_{x\,\max} \approx I_{y\,\max} \leqslant I_п, \tag{5.3}$$

где $I_п$ – максимальное (пороговое) значение тока, при котором еще не происходит перемагничивания сердечника по полному циклу.

Выполнение условия (5.2) обеспечивает надежное перемагничивание 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

$$\delta_x = \frac{I_{x\,\max} - I_x}{I_x} = \frac{I_x - I_{x\,\min}}{I_x}; \tag{5.4}$$

$$\delta_y = \frac{I_{y\,\max} - I_y}{I_y} = \frac{I_y - I_{y\,\min}}{I_y}. \tag{5.5}$$

Тогда для случая максимальных допусков соотношения (5.2) и (5.3) можно записать в виде

$$(1 - \delta_x) I_x + (1 - \delta_y) I_y = I_m; \tag{5.6}$$

$$(1 + \delta_x) I_x = I_п; \tag{5.7}$$

$$(1 + \delta_y) I_y = I_п. \tag{5.8}$$

Совместное решение уравнений (5.6)–(5.8) с учетом обозначения $\alpha = I_m/I_ц$ позволяет получить соотношение

$$\frac{1-\delta_x}{1+\delta_x}+\frac{1-\delta_y}{1+\delta_y}=\alpha, \qquad (5.9)$$

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 $\delta_z = \delta_{зп} = 24\,\%$ и в трехмерном $\delta_x = \delta_{зпр} = 7{,}5\,\%$. 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 число 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 3*D* или разрядной шиной считывания (записи) в МОЗУ типа 2*D*;

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 импульса тока запрета.

5.2. Состав и назначение основных блоков оперативной памяти ЭВМ ЕС-1022

Структуру и организацию функционирования 2,5-мерной оперативной памяти большой емкости рассмотрим на примере ОЗУ машины ЕС-1022 (шифр этого устройства ЕС-3222-02). Устройство ЕС-3222-02 предназначено для приема, хранения и выдачи информации при работе совместно с процессором ЕС-2622 модели ЕС-1022.

Оперативная память ЕС-3222-02 состоит из двух типов логически самостоятельных частей: основной памяти (ОП ОЗУ) и мультиплексной памяти (МП ОЗУ). Локальная память, которая в запоминающем устройстве ЕС-3220 машины ЕС-1020 представляет собой третью логически самостоятельную часть, в ЭВМ ЕС-1022 оформлена в виде отдельной сверхбыстродействующей регистровой памяти в составе процессора, что наряду с другими мерами обеспечило повышение быстродействия этой машины. В стойке оперативной памяти ЕС-3222-02 конструктивно расположено также запоминающее устройство ключей памяти (ЗУКП).

Таблица 5.1

Емкость оперативной памяти модели ЕС-1022

Модификации		Объем памяти			Количество магнитных блоков ЕС-3222/К002
процессора	оперативной памяти	ОП, байт	МП, байт	ЗУКП, 6-разрядные слова	
ЕС-2622-01	ЕС-3222-02 (одно устройство)	256К	2К	128	4
ЕС-2622-02	ЕС-3222-02 (два устройства)	512К	2К	256	8

В зависимости от модификации процессора и оперативной памяти емкость ОП, МП и ЗУКП соответствует данным табл. 5.1.

В каждый данный момент обращение может осуществляться только к одному из двух указанных типов памяти — основной или мультиплексной — в соответствии с поступившим из процессора признаком по одной из шин ОП или МП. При обращении к памяти производится запись или считывание одновременно четырех байт информации в параллельном коде. Возможна совмещенная по времени работа ОП и ЗУКП. При отключении питания данные в ОП и МП сохраняются без разрушения, а в ЗУКП информация разрушается. Собственно запоминающие части ОП и МП, построенные по системе выборки 2,5*D* на термостабильных ферритовых сердечниках типа М3ВТ — К 0,6 × 0,55 × 0,22 мм, совместно с оконечными каскадами адресно-разрядной дешифрации и усилителями считанных сигналов объединены в магнитные блоки (МБ). Для ОП и МП является общими схемы управления их работой, формирования разрядно-адресных токов и усиления считанных сигналов.

Полный цикл обращения к памяти делится на два самостоятельных такта: такт считывания и такт записи, начало которых определяется сигналами СЧИТЫВАНИЕ и ЗАПИСЬ, поступающими из процессора. По сигналу СЧИТЫВАНИЕ в соответствии с адресом ячейки в одном из двух типов памяти выдается четырехбайтное число на кодовые шины (КШЧ). При этом информация в выбранной ячейке памяти стирается и не восстанавливается. По сигналу ЗАПИСЬ осуществляется запись четырех байт информации, поступившей по кодовым шинам записи (КШЗ), в ту ячейку, из которой перед этим произошло 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 число из регистра РЧ переписывается 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 0,95 мкс.

Основными функциональными блоками оперативной памяти являются (рис. 5.2) блок формирования и распределения управляющих сигналов

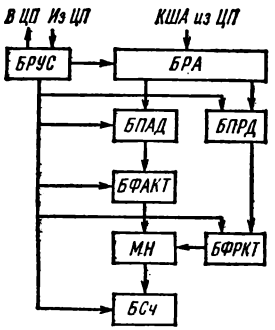


Рис. 5.2. Схема оперативной памяти ЕС-3222-02

БРУС, блок регистра адреса БРА, блок предварительной адресной дешифрации БПАД, блок предварительной адресной дешифрации БПРД, блок формирования адресных координатных токов БФАКТ, блок формирования разрядных координатных токов БФРКТ, магнитный накопитель МН, блок считывания БСч.

Блок формирования и распределения управляющих сигналов выполняет функции блока местного управления оперативной памяти. В зависимости от значения управляющих сигналов, поступающих в этот блок из центрального процессора ЕС-2422, в нем производится формирование требуемой временной диаграммы и выдача необходимых сигналов управления в остальные блоки памяти. Из блока формирования и распределения управляющих сигналов в процессор подаются сигналы, представляющие собой признак отсутствия сбоя по адресации и признак сбоя регистра адреса.

Блок регистра адреса предназначен для приема кода адреса, по которому осуществляется обращение из процессора, контроля этого кода на нечетность и хранения его на протяжении тактов считывания и записи. Адрес в блок регистра адреса поступает импульсным однофазным кодом по 22 кодовым шинам адреса КША. Регистр адреса является 22-разрядным, причем разряды объединены в три байта. Первые два байта 9-разрядные, включающие восемь информационных разрядов и один контрольный. Третий байт неполный, 4-разрядный, причем три разряда информационных и один контрольный.

Код адреса управляет работой адресных и разрядных дешифраторов. Для адресации основной памяти используются 0–18-й разряды БРА, а для адресации МП – 0 – 10-й разряды БРА. Нулевой и первый разряды в дешифрации не участвуют, они служат для выбора одного из четырех байт считанной информации.

Блоки предварительной адресной и предварительной разрядной дешифрации обеспечивают управление адресной и разрядной дешифрацией путем расшифровки сигналов, поступающих из блоков БРА и БРУС. Основные узлы, входящие в состав блоков БПАД и БПРД,— линейные дешифраторы.

Блок формирования разрядных координатных токов служит для выбора (в соответствии с адресом) одной из 64-разрядных шин во всех 36 разрядах четырех МБ и формирования стабилизированных токовых импульсов записи в тех разрядах, в которых необходимо восстановить (при регенерации считанной информации) или записать заново код 1. В составе блока имеются 36 одинаковых разрядных дешифраторов ДШР и 22 генератора тока.

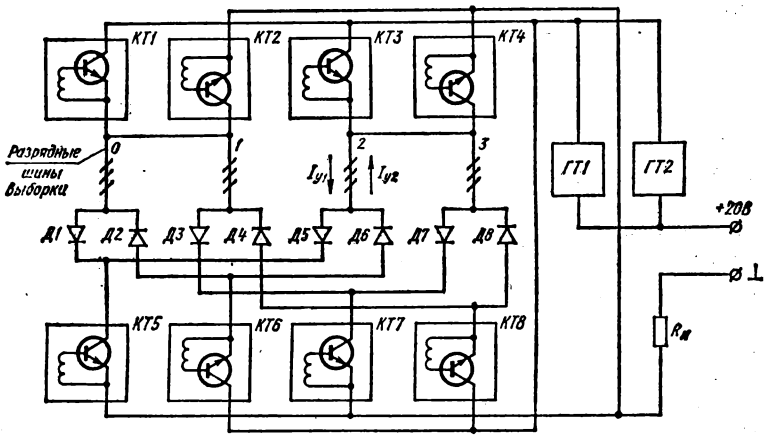


Рис. 5.3. Токовый дешифратор на четыре выхода

Разрядные токовые дешифраторы построены по принципу диодно-матричных переключателей с раздельными транзисторными ключами для токовых импульсов считывания и записи и двумя разделительными диодами на каждую разрядную шину выборки. Принцип построения дешифратора рассмотрим на примере дешифратора на четыре выхода (рис. 5.3).

На рис. 5.3 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 регистр $R_и$, источники питания + 20 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 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 $R_н$ и на шину «Земля». 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 $R_н$ и на шину «Земля». 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 переключении 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).

По описанному принципу построен каждый из 36 разрядных дешифраторов. В составе одного разрядного дешифратора имеются *8* пар координатных ключей и 64 пары разделительных диодов, оформленных в виде 16 интегральных диодных матриц — восьми с общим анодом и восьми с общим катодом.

Блок формирования адресных координатных токов предназначен для выбора адресной шины, соответствующей адресу и признаку памяти, формирования стабилизированных токовых импульсов считывания и записи и посылки их в выбранную адресную шину. В состав блока входят восемь одинаковых адресных токовых дешифраторов ДША и четыре генератора тока ГТА. Каждый магнитный блок обслуживается двумя такими дешифраторами, расположенными на сторонах *А* и *В* (ДША — *А* и ДША — *В*). С помощью адресного дешифратора обеспечивается формирование и коммутация стабилизированных по амплитуде разнополярных токовых импульсов считывания и записи в одну из 256 адресных шин выборки при обращении к ОП или в одну из четырех дополнительных шин выборки при обращении к МП. Адресные дешифраторы построены по тому же принципу, что и разрядные дешифраторы.

Магнитный накопитель служит для непосредственного хранения информации. Он состоит из запоминающих матриц на ферритовых сердечниках. Основная память ЕС-3222-02 объемом 256К байт конструктивно оформлена в виде четырех магнитных блоков. Емкость одного магнитного блока равна 64К байт.

Рассмотрим более подробно структуру магнитного блока (рис. 5.4). Запоминающий массив МБ разбит на девять ферритовых полей по числу разрядов байта (восемь информационных и один контрольный). Каждое ферритовое поле служит для хранения одноименных разрядов запоминаемых слов. Оно состоит из (64 + 1) К сердечников и оформлено в виде прямоугольной матрицы (1024 + 16) × 64 координат. Матрица разде-

лена на две равные части (полуматрицы) по $(215+8)\times 64$ координат, которые конструктивно расположены на двух сторонах (A и B) магнитного блока и связаны с адресными дешифраторами ДША – A и ДША – B на 256 выходов каждый.

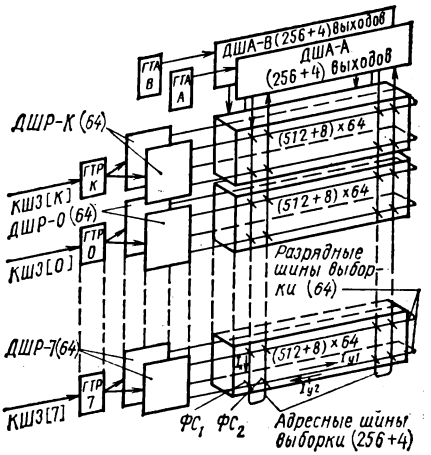


Рис. 5.4. Магнитный блок оперативной памяти ЕС-3222-02

Ферритовые сердечники полуматриц последовательно пронизываются 512 координатными шинами одной стороны МБ. Эти шины, будучи объединены попарно, образуют 256 адресных шин выборки. Каждая адресная шина выборки пронизывает в каждом разряде 128 сердечников – 64 прямой ветвью и 64 обратной.

Кроме адресных дешифраторов, имеется девять разрядных дешифраторов ДШР-0 – ДШР-7 и ДШР-К на 64 выхода каждый. Выходы разрядного дешифратора связаны с разрядными шинами соответствующего ферритового поля. Ферритовое поле любого разряда содержит 64 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 данными, поступающими по кодовым шинам записи КШЗ[0] – КШЗ[7], КШЗ[К].

Адресные и разрядные дешифраторы возбуждаются сигналами с адресного и разрядного генераторов тока.

Выборка одного из 64К сердечников в каждом ферритовом поле осуществляется следующим образом. Пусть выбранными оказываются адресная и разрядная шины, проходящие через ферритовые сердечники $ФС_1$ и $ФС_2$. 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 $ФС_2$, а во втором случае – сердечник $ФС_1$. Таким образом, за счет изменения направления разрядных токов в выбранной разрядной шине число выходов дешифратора ДША – A сокращается вдвое (256 вместо 512). То же самое относится к дешифратору ДША – B.

В каждой из двух плоскостей магнитного блока, кроме 256 адресных шин выборки основной памяти, имеются четыре дополнительные адресные шины, позволяющие получить необходимую емкость мультиплексной памяти.

Блок считывания предназначен для приема и усиления сигналов, поступающих из магнитного накопителя, отделения их от помех с помощью временного стробирования и амплитудной дискриминации, формирования выходных сигналов и посылки этих сигналов по кодовым шинам числа в процессор. В блок считывания входят обмотки считывания ферритовых матриц, расположенные в четырех МБ, усилители считывания, схемы развязки сигналов, управляющих работой усилителей считывания, и общие для всех четырех МБ схемы выдачи данных на КШЧ. Амплитудная селекция кодовых сигналов считывания производится после усиления и выпрямления сигналов; она осуществляется заданием порога срабатывания усилителя. Временная селекция обеспечивается сигналом СТРОБ, который формируется в БРУС индивидуального для каждого МБ и поступает в блок считывания.

5.3. Оперативная память машины ЕС-1033

Устройство памяти ЕС-3203 машины ЕС-1033 содержит собственно 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 128 кбайт. 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 2,5*D*. 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.

Устройства ОП1 и ОП2 полностью идентичны как по составу, так и по техническим данным и принципу действия. Каждое из них состоит из запоминающего магнитного блока (БЗМ) 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.

Устройства питания ОП1 и ОП2 конструктивно выполнены 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

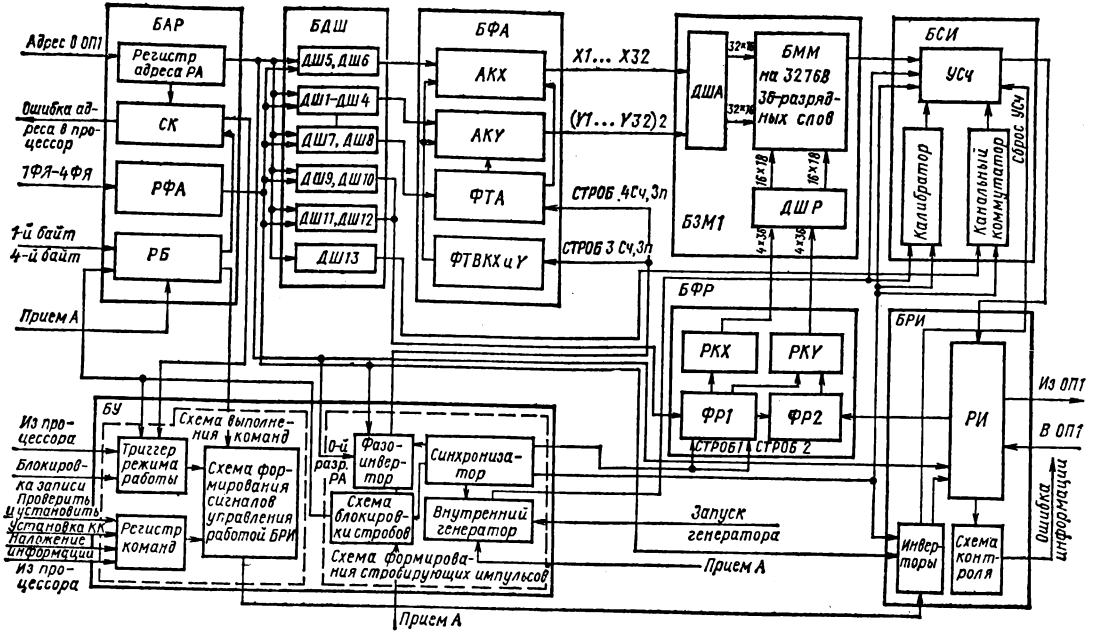


Рис. 5.5. Структурная схема запоминающего устройства ОП1 машины ЕС-1033

может быть подключено два устройства ЕС-3203, т. е. четыре запоминающих устройства ОП1 — ОП4.

Устройство ОП1 (рис. 5.5) содержит следующие функциональные блоки:

блок адресных регистров БАР;

блок дешифраторов БДШ;

блок адресных формирователей БФА;

блок разрядных формирователей БФР;

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. Блок 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 два 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-й байт — 4-й 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 mod 2. При наличии ошибки сигнал ОШИБКА АДРЕСА выдается в процессор и в блок управления..

Блок дишифраторов (БДШ). Блок 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, 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*;

ДШ13 – для управления работой усилителей считывания. Управление осуществляется через дешифратор секций на восемь выходов (канальный коммутатор), расположенный в блоке считывания информации.

Блок адресных формирователей (БФА). Блок 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* (АК*X*) (32 пары ключей), адресные ключи *Y* (АК*Y*) (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 имеются 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 задаются 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). Блок 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 РК*X* и РК*Y* (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 ФР1, которые, в свою очередь, запускаются сигналами СТРОБ 1 Сч и СТРОБ 1 Зп с блока управления. Формирователи

тока ФР1 по координате X управляются сигналами с дешифраторов ДШ11, ДШ12, а формирователи тока ФР1 по координате Y — сигналами с дешифраторов ДШ9, ДШ10.

Управление формирователями разрядного тока считывания-записи ФР2 осуществляется сигналами с регистра информации блока БРИ, причем в случае записи или регенерации нуля ток записи не формируется. Для возбуждения формирователей ФР2 используются импульсы с формирователей тока ФР1, 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 ФР2 данного разряда через возбужденные 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 ФР2.

Блок запоминающий магнитный (БЗМ). Этот блок 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 × 0,4 × 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-1022.

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). Блок 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 БЗМ 1) 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. Блок 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 mod 2 (по нечетности). При обнаружении ошибки формируется сигнал ОШИБКА инф., который посылается в процессор.

Блок управления БУ. Этот блок 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, обеспечивающую сохранение информации в ОП1 при отключении питания;

внутренний генератор, вырабатывающий стробирующие и запускающие импульсы для усилителей считывания и калибратора БСИ. С его помощью осуществляется запуск калибратора и стробирование калибровочного сигнала в УСч. Внутренний генератор содержит линию задержки с инвертором, формирователи стробирующих и запускающих импульсов и управляющий триггер; 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 и ОП2. Для устройства ОП1 (ОП2) предусматриваются следующие режимы работы.

1. «Запись» – в этом режиме выполняется 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 в РИ и выдача затем из 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 байтов на регистр байтов (1-й байт – 4-й байт).

По сигналу ПРИЕМ А триггер режима работы устанавливается в состояние, соответствующее режиму «Запись», осуществляется прием кода адреса в регистр блока адресных регистров, запускается синхронизатор и выключается внутренний генератор блока управления. В соответствии с кодом адреса в блоке дешифраторов вырабатываются управляющие сигналы, которые поступают в адресные и разрядные формирователи тока, а также в ключи *X* и *Y* и в канальный коммутатор блока БСИ. Синхронизатор блока управления в определенные моменты времени посылает стробирующие сигналы в формирователи токов в блоках БФА и БФР, в канальный коммутатор и усилители считывания.

В регистр информации импульсы приема информации поступают только для тех байт, признаки которых переданы в регистр байт. Для других байт в регистр информации поступают сигналы управления регенерацией информации. В такте считывания в режиме «Запись» стробирующие импульсы на входы усилителей считывания и соответственно кодовые сигналы с выходов усилителей в регистр информации поступают только для тех байт, по которым нет записи новой информации, а производится лишь регенерация считанной. 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 в 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» в 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 байт.

В ОП1 предусмотрены четыре диагностических режима. В режиме «Установка контрольного кода» принимается информация 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 в 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 регистра 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-1033 осуществляется с пульта управления модели с помощью схем контроля оперативной памяти. Для этой цели используется также соответствующий программный тест.

5.4.

Оперативная память старших моделей ЕС ЭВМ

Оперативное запоминающее устройство ЕС-3205, входящее в состав старших моделей ЕС ЭВМ (ЕС-1050 и др.), включает собственно ОП и ее систему питания, 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 2,5*D*. 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 кабельную магистраль. К ней могут быть подключены четыре устройства ЕС-3205.

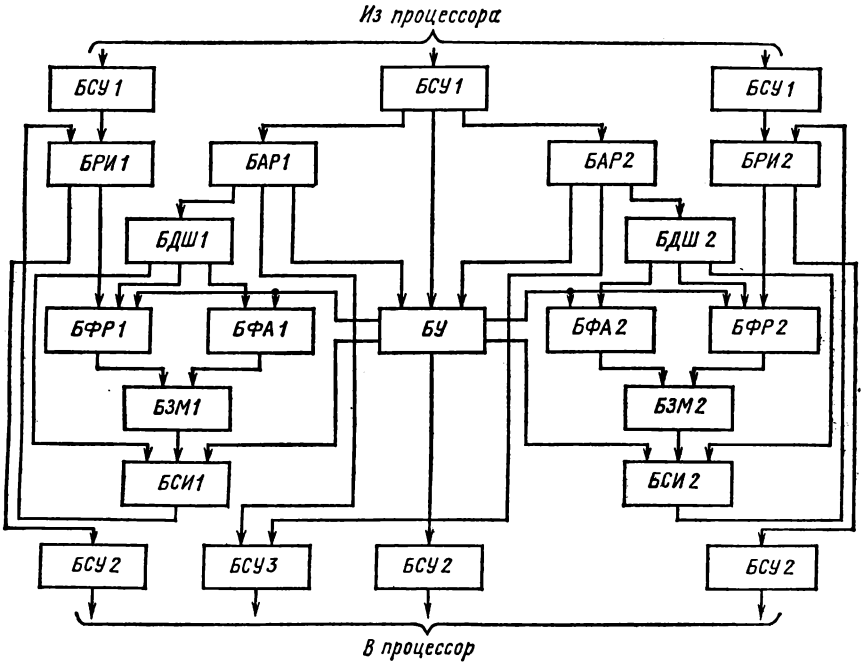


Рис. 5.6. Структурная схема оперативной памяти машины ЕС-1050

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. Она содержит следующие основные функциональные блоки (рис. 5.6):

адресных регистров БАР1 и БАР2;
дешифраторов БДШ1 и БДШ2;
адресных формирователей БФА1 и БФА2;
разрядных формирователей БФР1 и БФР2;
запоминающие магнитные БЗМ1 и БЗМ2;
считывания информации БСИ1 и БСИ2;
регистра информации БРИ1 и БРИ2;
согласователей уровней БСУ1, БСУ2, БСУ3;
управления БУ.

Эти блоки по своей структуре и выполняемым функциям аналогичны соответствующим блокам устройства ЕС-3203, поэтому ограничимся их краткой характеристикой.

Блоки адресных регистров имеют в своем составе два регистра РА1 и РА2 (для каждого БЗМ – свой регистр), регистр маркеров и схемы контроля адреса. Регистр РА1 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 РА2 имеет 17 разрядов (15 информационных и 2 контрольных); он выполняет те же функции по отношению к БДШ2 и БЗМ2. Регистр маркеров служит для приема и хранения признаков байт и контрольного разряда, а также для управления работой регистров информации в режиме записи по байтам и в режиме «Проверить и установить». Контроль РА1 и РА2 и регистра маркеров осуществляется по схеме непрерывного контроля по mod 2.

Каждый из блоков 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 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 32768 35-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 (ОК – 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, во входные сигналы процессора (стандартные сигналы интегральных схем серии 137), а также для преобразования выходных сигналов процессора во входные сигналы ОП. Для устройства ЕС-3205 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.

5.5.

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-1022. Основные характеристики этой памяти: емкость 6144 64-разрядных слова, время выдачи информации (слова) 500 нс, напряжение питания + 5 В, –5 В, потребляемый ток от этих источников питания соответственно равен 6 и 1,7 А.

Для хранения информации используются ферритовые сердечники с непрямоугольной петлей гистерезиса (оксиферовые сердечники), работающие как линейные трансформаторы. Основой для построения ПЗУ трансфор-

маторного типа служит числовая линейка (рис. 5.7, *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. 5.7, *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. 5.7, *а* отсутствует). Таким образом, при записи информации во время монтажа ПЗУ машины ЕС-1022 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 индуктируются 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 ЭДС сигнала. В выходных обмотках сердечников, хранящих нуль, сигнал отсутствует.

В ПЗУ машины ЕС-1022 одна числовая линейка служит для хранения 128 128-разрядных слов, т. е. в ее состав входит 128 ферритовых сердечников (ФС1 — ФС128), прошитых 128 входными шинами. Такая числовая линейка представляет собой модуль постоянной памяти. Общее количество модулей в ПЗУ равно 24, нумерация модулей — от 0 до 23. При обращении к ПЗУ из выбранного модуля считывается 128-разрядное слово, а так как передача в другие блоки процессора осуществляется 64-разрядными словами (микрокомандами), то считанное слово разбивается на два машинных слова — четное и нечетное. Выходные обмотки одноименных разрядов модулей соединены параллельно (рис. 5.7, *б*).

Схема ПЗУ машины ЕС-1022 приведена на рис. 5.8. Основные ее составные части следующие: регистр адреса РА, дешифратор 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.

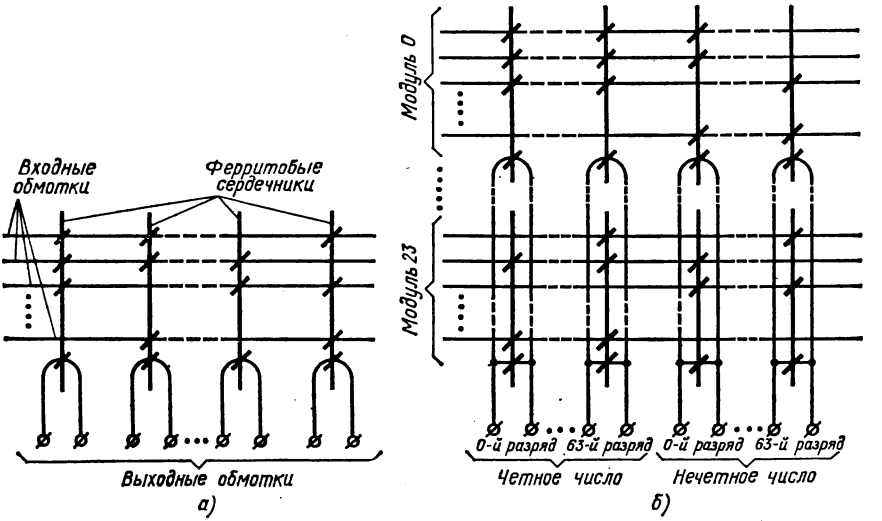


Рис. 5.7. Числовая линейка *(а)* и схема соединения выходных обмоток сердечников числовых линеек *(б)* в ПЗУ

Первая ступень дешифрации имеет дешифраторы ДШУ, ДШХ и ДШМ, причем управление этими дешифраторами осуществляется сигналами, поступающими соответственно с [1—3] разрядов, с [4—7] разрядов и с [8—12] разрядов регистра адреса. Дешифратор модуля ДШМ с выходами М[0] — М[23] определяет номер выбираемого модуля ПЗУ. Дешифраторы ДШХ и ДШУ обеспечивают выборку входной обмотки в избранном модуле.

Вторая ступень дешифрации предназначена для формирования импульса тока опроса в выбранной входной обмотке избранного модуля. Она состоит из 24 идентичных дешифраторов провода ДШП[0] — ДШП[23].

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 4 или СТРОБ 5, 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 три блока модулей БМ1 — БМ3 по восемь

модулей в каждом. В состав каждого модуля, кроме 128 сердечников и 128 входных обмоток, входит соответствующий дешифратор провода. Входные обмотки выполнены методом печатного монтажа на фторопластовой подложке.

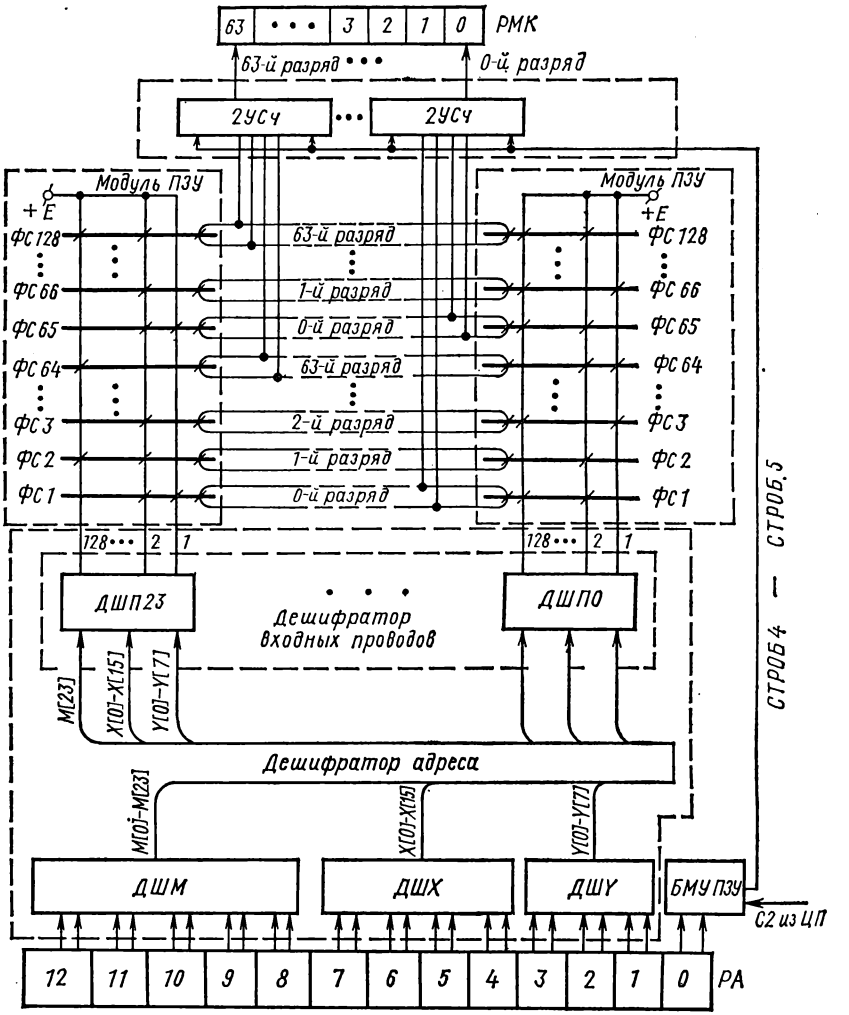


Рис. 5.8. Схема ПЗУ машины ЕС-1022

Узел усилителей считывания предназначен для разделения четного и нечетного слов, усиления и дискриминации 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 4 или СТРОБ 5, 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 mod 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. Формирование временной диаграммы управляющих сигналов осуществляется с приходом импульса *С2* из центрального процессора (ЦП) машины.

Глава 6

АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА ПРОЦЕССОРОВ

6.1. 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 Р, предназначенных для приема и размещения надлежащим образом операндов, над которыми должны производиться действия при выполнении очередной операции;

операционную часть О, где осуществляются преобразования операндов согласно машинным алгоритмам арифметических, логических и других операций, на выполнение которых рассчитано АЛУ;

схемы контроля К, 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-1022 и ЕС-1033 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 (52) 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.

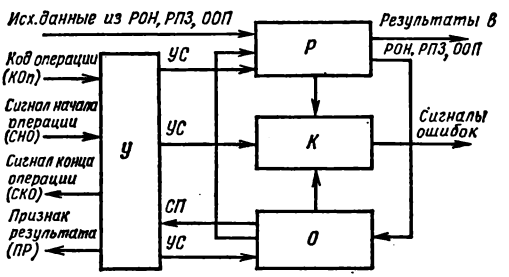


Рис. 6.1. Обобщенная структурная схема АЛУ процессоров

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. 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-1022 в основном регистры 16- и 32-разрядные, в ЕС-1033 – 32-разрядные, в ЕС-1050 (52) – 64-разрядные. Это является одной из причин, обусловливающих различия 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 (52) операционная часть состоит из двух блоков с явно выраженными сумматорами в каждом из них (рис. 6.2, *а*): блока арифметического сумматора БАС и 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 прини-

мать и обрабатывать операнды всех фиксированных форматов (от полуслова до двойного слова).

В блоке БАЦ производится только побайтная 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.

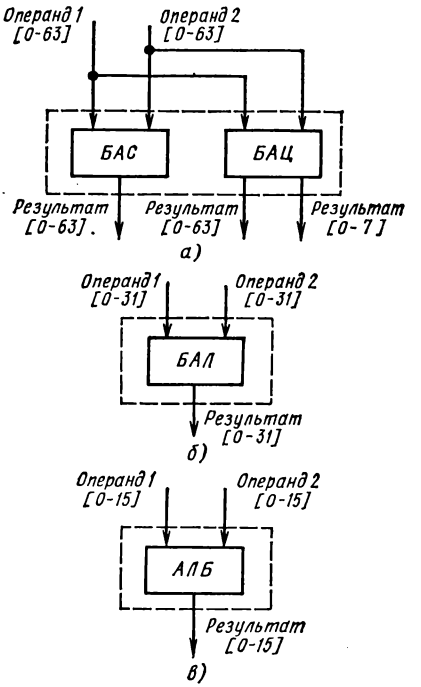


Рис. 6.2. Состав операционной части АЛУ процессора:
а — ЕС-2050 (52);
б — ЕС-2433;
в — ЕС-2622

Два дополнительных 64-разрядных регистра позволяют БАЦ принимать для обработки от каждого операнда одновременно по 8 байт. Блок может выдавать результат либо побайтно, либо параллельно по 64 разряда.

В модели ЕС-1033 основу операционной части (рис. 6.2, *б*) составляет 32-разрядный универсальный блок арифметических и логических операций БАЛ. В нем выполняются все арифметические и логические действия над двоичными и двоично-кодированными (в том числе и десятичными) операндами фиксированных и переменных форматов.

В АЛУ модели ЕС-1022 операционная часть представляет собой универсальный коммутируемый 16-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 сложения по mod 2, а также операций сквозной передачи операндов через блок. В результате процессы выполнения операций над многобайтными операндами слагаются из последовательного выбора и преобразования пар их двухбайтных частей по микропрограммам этих операций. При этом из основной оперативной памяти операнды выбираются или записываются туда по четыре байта за одно обращение к ней, а обмен с локальной памятью осуществляется полусловами.

Различия в составе и принципе действия АЛУ явились результатом разных требований к производительности моделей и желаемому объему

их оборудования, предъявленных разработчиками 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 их контроля.

6.2. Структуры и основные информационные связи арифметико-логических устройств процессоров

АЛУ процессора ЕС-2050(52). Исключая схемы контроля и цепи управляющих сигналов, структурная схема АЛУ процессора ЕС-2050(52) 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

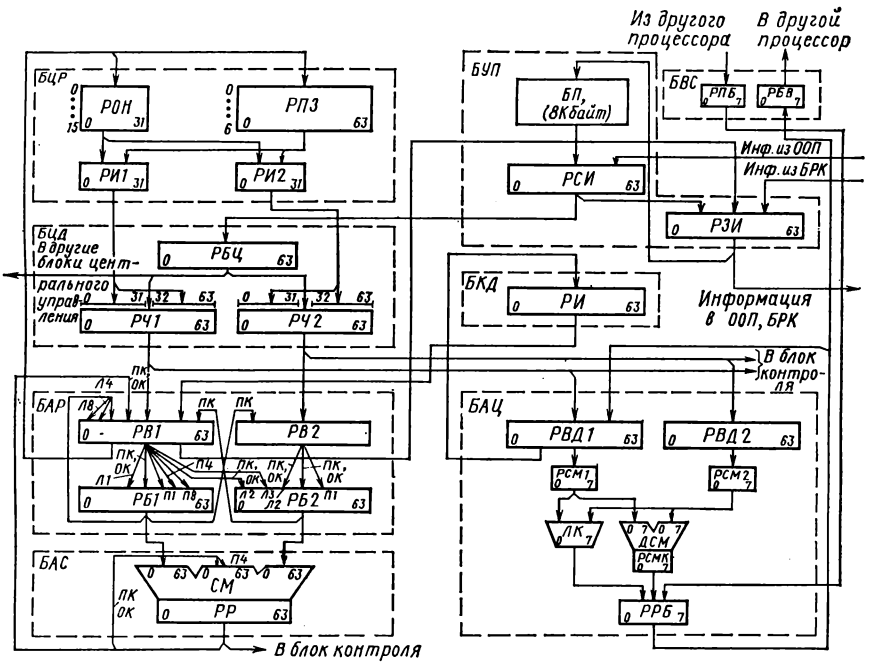


Рис. 6.3. Структурная схема АЛУ и его информационных связей с другими блоками процессора ЕС-2052

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 БАР ЕС-2050 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 блоке управления памятью (БУП) процессора ЕС-2052 в отличие от процессора ЕС-2050 имеется буферная память емкостью 8К байт, выполненная на интегральных схемах. Она является быстродействующей буферной сверхоперативной памятью БСОП между ОПП, буферными регистрами канала БРК и центральным процессором. Наличие такой БСОП обеспечивает повышение производительности процессора ЕС-2052 по сравнению с процессором ЕС-2050. В процессоре ЕС-2050 (52) 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 блока выбора данных БЦД процессора. Через этот блок проходит вся информация, которая должна обрабатываться в АЛУ.

Информация, запрашиваемая из основной оперативной памяти, заблаговременно вводится в буферную память БП блока управления оперативной памятью. Оттуда она, по мере необходимости, через регистр считанной информации РСИ подается в буферный регистр центрального управления РБЦ. В этот регистр поступают не только исходные операнды. Из ООП через БП в регистр поступают 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 БП.

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 операнды 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 буферную память БП через регистр записываемой информации РЗИ 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 умещающимися полностью в РВД1 и РВД2, они вызываются в БАЦ по частям, каждая из которых не превышает 8 байт. В этом случае результат также вырабатывается по частям. Каждые очередные 8 байт результата отсылаются в БП 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 РБВ блока БВС. Наоборот, принимаемые на обработку байты поступают на входы РРБ через приемный регистр байта РПБ.

С отсылкой результата в местную или буферную 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 БАК. Проверка правильности работы оборудования блока БАЦ основана на применении сквозного контроля по mod 2.

Рассмотренные связи АЛУ с местной регистровой и буферной оперативными памятями 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 (52) приняты три уровня совмещения, одним из которых является выполнение действий над операндами в АЛУ и запись результата в местную или буферную сверхоперативную память.

Одновременно с этим на других уровнях подготовляются операнды на регистрах РЧ1 и РЧ2 для выполнения очередной команды и производится выбор в процессор следующей за ней команды. Большинство арифметических операций кончается записью в местную память. Запись в буферную память, как и отсылка в основную оперативную память, в большинстве случаев производится по командам загрузки, при исполнении которых не требуется работа АЛУ. Поэтому время выполнения команды на первом из указанных уровней в основном определяется временем выполнения операции в АЛУ (табл. 6.1).

Таблица 6.1

Сравнительные временные данные для АЛУ различных моделей ЭВМ ЕС

Тип операции	Время выполнения операции, мкс		
	ЕС-1050	ЕС-1033	ЕС-1022
Короткие операции	0,56	1	3,3
Сложение-вычитание:			
с фиксированной запятой	0,56	1	3,3
с плавающей запятой	0,72–1,4	2,8–5,21	14,04–15,84
Умножение:			
с фиксированной запятой	2,32	8,5	29,1
с плавающей запятой	2,0–3,2	13,02–32,44	39,33–168
Деление:			
с фиксированной запятой	6,48	29,53	65,1
с плавающей запятой	5,76–12	17,72–48,36	51–144

На выполнение действий на всех уровнях отводится столько времени, сколько требуется. Это обеспечивается независимой работой счетчиков тактов.

Счетчик тактов уровня начинает работу, когда данный уровень освобождается (закончилось преобразование или хранение информации) и появляется признак готовности предыдущего уровня. Минимально возможный цикл счетчиков тактов всех уровней составляет шесть полутактов (полутакт в ЕС-1050 равен 80 нс). Этот цикл определяется временем обращения к РОН или РПЗ местной памяти.

При выполнении операции в БАС запись в БП 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 БП. В то же время из БП через блок БЦД принимаются на обработку продолжения операндов.

АЛУ процессора ЕС-2433. Для выборки из программы очередной команды и формирования исполнительных адресов операндов в процессоре ЕС-2050 (52) 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 оперативной памяти, и при действиях над кодами адресов предыдущей команды.

В процессоре ЕС-2433 практически нет такого совмещения выполнения указанных действий. Выполнение очередной команды здесь слагается из последовательности следующих друг за другом этапов: выборки из ООП кода команды, формирования исполнительных адресов операндов и их выборки из регистровой или основной оперативной памяти, выполнения операций и запоминания результата. При такой последовательности действий одно и то же оборудование процессора в разные интервалы времени используется для различных целей.

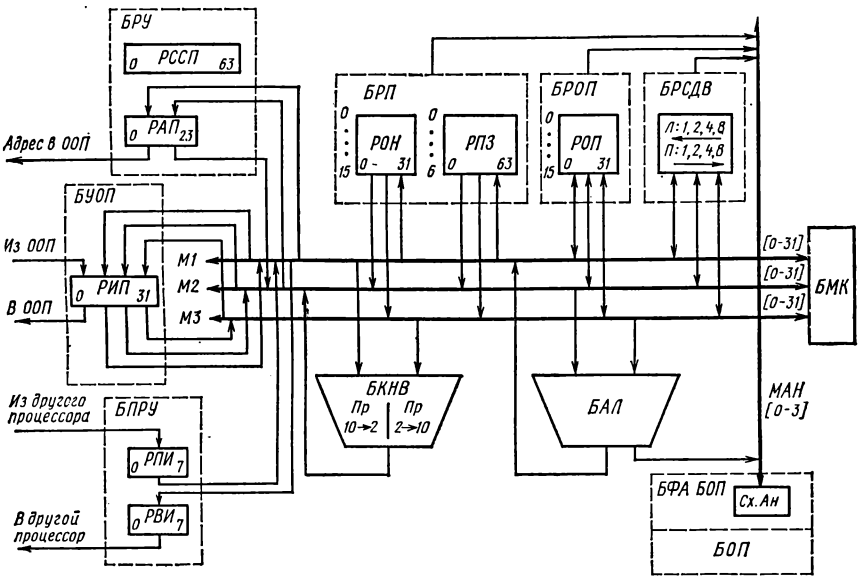


Рис. 6.4. Структурная схема информационных связей блоков АЛУ с регистрами процессора и с оперативной памятью в модели ЕС-1033

Это в первую очередь относится к АЛУ, основу которого составляет комбинационный арифметико-логический блок БАЛ. В начале исполнения команды с его помощью формируются исполнительные адреса операндов, если они находятся в ООП. Затем он участвует в выполнении арифметических и логических операций.

Для удобства многоцелевого использования узлов и блоков процессора ЕС-2433 в нем применена магистральная структура связей. Все блоки и их узлы связаны друг с другом через три 4-байтные информационные магистрали, как показано на рис. 6.4. Над магистралями М1, М2 и М3 изображены регистровые блоки, регистры которых образуют единое поле памяти. Перед выполнением каждой операции в ней размещаются необходимая командная информация и преобразуемые данные. Она используется также для промежуточного хранения информации на

этапах выполнения операции и результата. На схеме выделены основные регистры. Вспомогательные малоразрядные регистры и триггеры, предназначенные для запоминания различных кодов, сигналов, отражающих различные состояния, условия, результаты анализа преобразуемой информации, необходимые для правильного выполнения арифметических, логических и других операций, на схеме не показаны.

Коммутацию связей между блоками процессора осуществляет блок магистрального коммутатора БМК. При этом он обеспечивает поддержание информационных сигналов в магистральных шинах в течение времени, необходимого для преобразования информации в комбинационных схемах и ввода результатов в регистры. Регистровая оперативная память сгруппирована в два блока. В блоке программных регистров БРП находятся программно адресуемые регистры общего назначения РОН и регистры для операндов с плавающей запятой РПЗ. Вспомогательную сверхоперативную процессорную память образуют 16 операционных регистров РОП, скомпонованных в блок БРОП. Они используются для временного хранения кодов команд, адресов, операндов и промежуточных результатов в ходе выполнения микропрограмм операций. В блоке регистров сдвига БРСДВ находятся регистры, обеспечивающие сдвиги преобразуемых операндов вправо или влево на необходимое число разрядов, регистр результата деления и регистр параметра нормализации, хранящий число разрядов, на которое осуществляется сдвиг.

Для преобразования чисел из десятичной системы счисления в двоичную ($10 \rightarrow 2$) и обратно ($2 \rightarrow 10$) в процессоре ЕС-2433 имеется блок конвертирования БКНВ. Он выполнен на интегральных схемах типа постоянных запоминающих устройств ПЗУ и работает по принципу комбинационных схем. В циклических процессах преобразования по преобразуемым кодам из ПЗУ последовательно выбираются и формируются их эквиваленты противоположной системы счисления. Преобразование осуществляется согласно алгоритмам, иллюстрируемым примерами 3.17 и 3.18. Блок БКНВ состоит из двух преобразователей: преобразователя в двоичную систему счисления Пр"$10 \rightarrow 2$" и преобразователя в десятичную систему Пр"$2 \rightarrow 10$".

Адреса команд и операндов выдаются в ООП через регистр адреса процессора РАП, расположенный в блоке управляющих регистров БРУ. Обмен информацией между процессором и ООП осуществляется через регистр информации процессора РИП, входящий в состав блока управления оперативной памятью БУОП.

Выполнение любой арифметической или логической операции начинается с выборки в процессор соответствующей команды. Очередная выбранная команда через РИП блока БУОП заносится в общем случае в регистры РОП1 и РОП3 блока БРОП. Размещение команды в регистрах зависит от формата команды и от того, в какой половине очередного выбираемого командного слова (32-разрядного) расположены ее старшие байты. Если один или оба операнда, участвующие в операции, находятся в ООП, то с помощью БАЛ формируются их исполнительные адреса. Они последовательно через магистраль М1 и РАП выдаются в регистр адреса ООП. Адреса $R1$ и $R2$ операндов при выборке из ООП очередной команды заносятся в регистры-счетчики СЧ1 РОН и СЧ2 РОН блока

БРП (на рис. 6.4 не показаны). По этим адресам происходит обращение к регистрам РОН или РПЗ за исходными операндами и при записи результата, если он отсылается по адресу *R1*.

В зависимости от исполняемой операции исходные операнды поступают на входы БАЛ либо из регистров блока БРП, либо непосредственно с регистра РИП блока БУОП, либо из регистров блока БРОП. В последнем случае они соответствующим образом непосредственно перед выполнением операции размещаются в регистрах РОП.

Например, при сложении и вычитании 32-разрядных операндов с фиксированной запятой по командам формата RR первый операнд поступает на вход БАЛ из регистра РОН с адресом *R1* по магистрали М2, а второй – из регистра РОН с адресом *R2* по магистрали М3. Результат по магистрали М1 отсылается в РОН с адресом *R1*. При командах формата RX второй операнд подается в магистраль М3 и на вход БАЛ непосредственно из регистра РИП блока БУОП.

Для операций над операндами с плавающей запятой регистры РОП используются как для размещения мантисс исходных операндов, так и для их характеристик. Например, перед сравнением порядков характеристика первого операнда заносится в РОП2, а второго – в РОП3. Разность характеристик запоминается в РОП1. Перед действиями сложения и вычитания мантиссы длинных операндов размещаются в регистрах РОП11, РОП12 (мантисса первого операнда) и в РОП13, РОП14 (мантисса второго операнда). Это осуществляется при выполнении команд как формата RR, так и формата RX. Если порядки слагаемых или уменьшаемого и вычитаемого равны, то мантиссы, сначала младшие половины, а потом старшие, подаются на входы БАЛ для выполнения над ними операции сложения или вычитания. Младшая половина мантиссы результата помещается в РОП5, а старшая – в РОП4. При коротких операндах мантисса результата полностью помещается в РОП4.

Если результат получился нормализованным, то к старшим разрядам его мантиссы приформировывается характеристика и он отсылается в РОН на место первого операнда – сначала старшая половина, а затем – младшая. В случае отрицательного результата, что определяется по значению М2 [0] = 1, мантисса результата перед окончательным формированием результата и отсылкой его в РОН переводится из дополнительного кода в прямой.

При необходимости сдвиги исходных операндов с целью выравнивания порядков и результата (для нормализации) осуществляются с помощью регистра сдвига блока БРСДВ. Сдвигающие регистры этого блока в общем случае позволяют сдвигать числа вправо и влево на 1, 2, 4 и 8 разрядов за один машинный такт. В зависимости от выполняемых операций сдвигаемые коды либо предварительно запоминаются в РОП, либо подаются на входы БАЛ непосредственно из БРСДВ. Например, при умножении целых двоичных чисел с фиксированной запятой множимое и множитель предварительно размещаются соответственно в РОП3 и РОП4. Для хранения старших разрядов сумм частных произведений используется РОП5. Младшие разряды постепенно формируются в сдвигателе. В конце умножения младшие 32 разряда 64-разрядного произведения оказываются сформированными в сдвигателе блока БРСДВ, откуда они переносятся

в РОП4, где находился множитель. Затем старшая половина произведения из РОП5 пересылается в БРП по адресу *R1* в регистр РОН с четным номером, а младшая половина из РОП4 – в следующий за ним регистр РОН с нечетным номером.

В блоке БАЛ могут непосредственно выполняться операции как сложения, так и вычитания. Поэтому нет необходимости в предварительном преобразовании кодов операндов. При этом действия могут производиться как над двоичными, так и над десятичными числами. В последнем случае для внесения избытков 6 в тетрады второго операнда 32-разрядные его части предварительно складываются с помощью БАЛ с константой $6666\,6666_{10}$, специальным образом подаваемой на вторую магистраль М2. Изъятие избытков 6 из тетрад, из которых не было переноса при сложении операндов, осуществляется с помощью специальных потетрадных комбинационных корректирующих схем "Коррекция $+10_{10}$". Таких схем в 32-разрядном блоке БАЛ – восемь (по количеству тетрад). Они работают только при десятичных операциях.

При десятичных операциях сложения и вычитания адреса первого и второго операндов помещаются соответственно в регистры РОП14 и РОП15. Второй операнд полностью выбирается из ООП пословно перед операцией, начиная с младшего слова, и размещается в регистрах РОП6–РОП2. При этом он выравнивается по правой границе первого операнда. Первый операнд выбирается из ООП пословно в процессе выполнения операции. Очередное выбранное слово заносится в РОП7. После сложения или вычитания очередное слово результата замещает использованное слово второго операнда в соответствующем регистре РОП. При обработке очередных пар слов операндов запоминается значение переноса из старшего (0-го) разряда сумматора.

Логические операции также выполняются с помощью блока БАЛ. Обрабатываться могут целиком целые слова, располагающиеся в регистрах РОН. Если же логические действия выполняются над полями данных, то необходимые байты из поступающих пословно данных из ООП выбираются путем коммутации цепей связи регистра РИП блока БУОП с магистралью М1. Имеется возможность входы и выходы любой 8-разрядной группы [И0 – 7], [8 – 15], [16 – 23] и [24 – 31] регистра РИП соединять с [24–31] разрядами магистрали М1. Выбор нужного байта из поступившего в РИП слова осуществляется соответствующим дешифрированием кода адреса байта. Аналогичным образом байт результата отсылается в нужную 8-разрядную группу РИП из [24–31] разрядов М1. Выделение нужных байтов из содержимых регистров блоков БРП и БРОП при подаче на входы БАЛ осуществляется путем соответствующего маскирования входов магистралей. Для этого в блоке магистрального коммутатора БКМ имеются два регистра для кодов масок и схема коммутации масок. Если надо открыть какой-либо вход магистрали, в соответствующем разряде кода маски, предназначенного для этой магистрали, должна быть единица. Коды масок вводятся в регистры блока БКМ микрокомандами в процессе выполнения микропрограмм операций.

Сигналы – признаки результатов действий, выполняемых в БАЛ, а также различные необходимые признаки информации, находящейся в ре-

гистрах блоков БРП, БРОП, БРСДВ и др., выдаются в 4-разрядную магистраль анализа МАН [0—3]. По этой магистрали они поступают в схему анализа Сх.АН блока формирования адресов БФА БОП блока односторонней (постоянной) памяти. В этой памяти хранятся микропрограммы операций. Анализ признаков в Сх.АН дает возможность определять последующие действия в зависимости от промежуточных результатов. Программные признаки — коды условия формируются непосредственно в БАЛ и заносятся в 34-й и 35-й разряды РССП — регистра слова состояния программы.

Непосредственный обмен информацией с другой ЭВМ может осуществляться через блок прямого управления БПРУ. Прием и выдача информации осуществляются соответственно через 8-разрядные регистры РПИ и РВИ. В модели ЕС-1033 применен сквозной контроль по mod 2.

Применение быстродействующих интегральных схем арифметико-логических устройств обеспечивает формирование в БАЛ результатов, например, сложения с учетом максимального распространения переносов за время, не превышающее 120 нс. Однако из-за последовательного характера действий, производимых при исполнении каждой команды, затрачиваемое на выполнение операций время больше в модели ЕС-1033, чем в модели ЕС-1050 (52). В табл. 6.1 указаны средние времена выполнения операций над операндами фиксированных форматов (с учетом времени обращения к оперативным запоминающим устройствам).

АЛУ процессора ЕС-2622. Процессор ЕС-2622 имеет регистровую структуру, в которой для выполнения арифметических и логических операций предусмотрен 16-разрядный универсальный арифметико-логический блок АЛБ комбинационного типа. В процессоре есть локальная оперативная память ЛП емкостью 256 полуслов. Она используется для хранения содержимого 16 регистров общего назначения РОН и четырех регистров с плавающей запятой РПЗ, информации, необходимой для работы мультиплексного и селекторного каналов. Четвертая часть ЛП отведена под рабочее поле вычислительно-управляющего устройства ВЧУ, где во время выполнения операций хранятся промежуточные результаты и вспомогательные данные, например кратные множимого. Запоминающая часть ЛП выполнена на полупроводниковых интегральных схемах, содержащих триггерные матрицы.

Регистры процессора имеют в основном вполне определенное назначение. Это — либо адресные регистры, либо информационные, где размещаются операнды и результаты, либо регистры специального назначения, такие как регистр прерываний, регистр слова состояния программы ССП и т. п. Системой информационных магистралей большинство регистров связаны друг с другом через общий АЛБ, например, как показано на рис. 6.5. На схеме изображены только основные информационные регистры: РД и РДБ — основной и буферный регистры данных блока связи с ООП, РЛ — регистр данных локальной памяти, РУ — регистр общего назначения, РА и РБ — входные регистры АЛБ. Управление перемещением информации и выполнением различных действий в цикле каждой операции осуществляется, как и в ЕС-2433, микропрограммным способом. Суть его и более полная структура процессора ЕС-2622 рассматриваются в гл. 7.

Пары байт очередных операндов, предназначенные для преобразования в комбинационных схемах АЛБ, заносятся в регистры РА и РБ. Эти регистры являются входными для АЛБ. Информация в них может поступать почти из всех регистров процеессора. АЛБ может работать в двух режимах: в режиме одновременного преобразования пар байт, поступивших в РА и РБ, и в режиме побайтного преобразования информации.

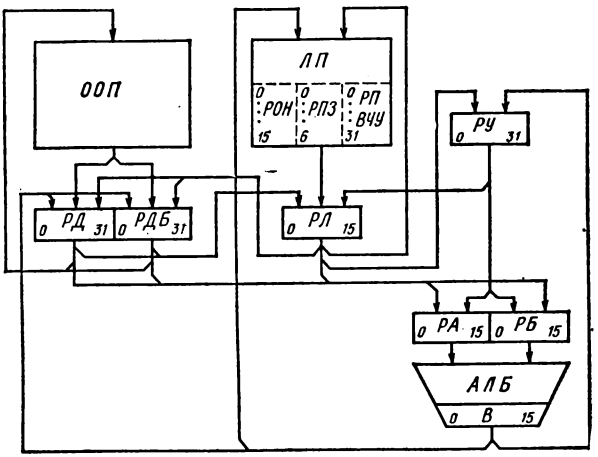


Рис. 6.5. Структурная схема информационных связей АЛБ с регистрами процессора и с оперативной памятью в модели ЕС-1022

Универсальность схемы АЛБ обеспечивает возможность выполнения им всех арифметических и логических операций над двоичными и десятичными операндами. Кроме того, АЛБ производит операции над адресами, через него также осуществляется обмен информацией между регистрами процессора. Для того чтобы можно было передавать информацию из регистра в регистр без изменений, в АЛБ предусмотрена возможность выполнения микроопераций транзитной пересылки (*А* ТРАНЗИТ и *Б* ТРАНЗИТ). С выхода В АЛБ информация может быть передана либо в любой регистр процессора, либо, минуя регистр РЛ, записана в ЛП, либо через РД – в ООП.

Обмен с ООП осуществляется словами, а с ЛП – полусловами. Принимаемые из этих запоминающих устройств операнды предварительно размещаются в регистрах РД, РДБ и РУ. Например, для операций типа сложения над 32-разрядными операндами с фиксированной запятой в РУ помещается первый операнд. Второй операнд засылается и хранится в регистре РД. Из этих регистров операнды по частям подаются соответственно в регистры РА и РБ. Вначале складываются полуслова, соответствующие младшим разрядам операндов. Результат непосредственно с выхода В блока АЛБ записывается в ЛП на место младших 16 разрядов первого операнда. Затем подаются на сложение старшие полуслова операндов. Старшие 16 разрядов результата аналогично предыдущему записываются в ЛП на месте старших 16 разрядов первого операнда.

Перед умножением 32-разрядных чисел с фиксированной запятой множимое помещается в РУ, а множитель – в РД. В начале операции формируются и запоминаются в рабочем поле ЛП кратные множимого *1А*, *2А*, *3А* и *6А* (см. табл. 3.3). Кратные множимого *1А* и *2А* – 32-разрядные, а *3А* и *6А* – 64-разрядные. В процессе умножения регистр РУ используется как рабочий регистр, в котором хранятся 32 старших разряда очередной суммы частных произведений. Младшие 32 разряда 64-разрядного произведения накапливаются в регистре РДБ, старшие – в РД, на месте использованных разрядов множителя.

При делении чисел с фиксированной запятой в РД хранится делитель, старшие 32 разряда делимого находятся в РУ, а младшие – в РДБ. В процессе деления в регистре РДБ накапливается частное.

В операциях сложения и вычитания чисел с плавающей запятой для хранения мантисс первого и второго операндов также используются соответственно регистры РУ и РД. В операциях с плавающей запятой, а также в операциях десятичной арифметики для хранения частей операндов, не умещающихся в этих регистрах, используются ячейки рабочего поля ЛП.

В АЛБ за один машинный такт длительностью 550–600 нс выполняется одна микрооперация (сложение двух полуслов, сдвиг байта или полуслова на разряд вправо или влево, транзитная передача полуслова из регистра в регистр и т. п.). Промежуточные и окончательные результаты, как правило, отсылаются прямо в ЛП, минуя РЛ. Последовательное использование регистров процессора вначале для действий над очередной командой, а затем для выполнения определяемой ею операции, преобразование информации в АЛБ парами байтов или побайтное приводят к необходимости значительной затраты времени на выполнение операций. Для сравнения времена выполнения операций в модели ЕС-1022 над операндами фиксированных форматов даны 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 и II) – комбинационные. В первой из них формируется двухрядный код, состоящий из кода S поразрядных полусумм (сумм по mod 2) и кода P поразрядных переносов, получаемых в результате сложения трех двоичных чисел. С выходов этой ступени коды S и P 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 ПК со сдвигом на четыре разряда вправо (П4).

В сумматоре второй ступени полное приведение переносов выполняется по каскадно-параллельной схеме. Переносы как внутри групп (тетрад и байтов), так и между группами (межбайтные) 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 байтные разрешения переносов. Эти

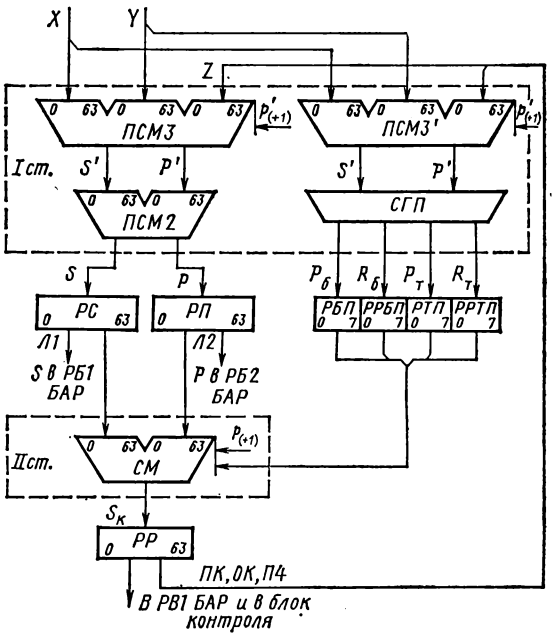


Рис. 6.6. Структурная схема БАС

переносы $P_б$ и $P_т$ и разрешения $R_б$ и $R_т$ в виде 8-разрядных кодов заносятся в регистры байтных и тетрадных переносов РБП и РТП и регистры разрешения байтных 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. Например, 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.

Формирование двухрядного кода S, P происходит в два этапа. В первой ступени при сложении трех 64-разрядных кодов $X = x_0x_1 \ldots x_i \ldots x_{63}$, $Y = y_0y_1 \ldots y_i \ldots y_{63}$ и $Z = z_0z_1 \ldots z_i \ldots z_{63}$ в каждом i-м разряде вначале вырабатываются промежуточные сумма s'_i и перенос p'_i, а затем — окончательные значения суммы s_i и переноса p_i, т. е.

$\ldots x_i \quad x_{i+1} \ldots$ — первое слагаемое

$\ldots y_i \quad y_{i+1} \ldots$ — второе слагаемое

$\ldots z_i \quad z_{i+1} \ldots$ — третье слагаемое

$\ldots s'_i \quad s'_{i+1} \ldots - S'$ } — промежуточный

$\ldots p'_i p'_{i+1} p'_{i+2} \ldots - P'$ } двухрядный код

$\ldots s_i \quad s_{i+1} \ldots - S$ } — окончательный

$\ldots p_i \quad p_{i+1} \ldots - P$ } двухрядный код

Коды S', P', S и P — 64-разрядные. В них значения i-х разрядов получаются в соответствии с формулами

$$s'_i = x_i \oplus y_i \oplus z_i; \tag{6.1}$$

$$p'_i = x_i y_i \vee x_i z_i \vee y_i z_i; \tag{6.2}$$

$$s_i = s'_i \oplus p'_{i+1}; \tag{6.3}$$

$$p_i = s'_i p'_{i+1}. \tag{6.4}$$

Рассмотрим пример. 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:

$$+\begin{cases} 0010\,1101 \\ 0100\,1000 \\ \underline{1011\,0000} \end{cases}$$

S' 1101 0101

P' 0101 000 $\boxed{1}$

S 1000 0100 } — окончательный

P 0101 0001 } двухрядный код

Промежуточные коды S' и P' вырабатываются входным трехвходовым полусумматором ПСМ3 (рис. 6.6). При этом 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 P' поразрядных переносов. В примере эта единица выделена рамкой, а на рис. 6.6 показана как $p'_{(+1)}$.

Коды S' и P' поступают в двухвходовый полусумматор ПСМ2, где согласно (6.3) и (6.4) формируются значения s_i и p_i, образующие окончательный двухрядный код, состоящий из кодов S и P. Цена единицы переноса p_i соответствует соседнему старшему двоичному разряду $i-1$. Это означает, что при подаче, например, кодов S и P снова на входы схемы суммирования первой ступени код переносов надо сдвинуть на разряд влево по отношению к коду полусумм. Поэтому, при передаче кода S в РБ1 БАР со сдвигом на один разряд влево, как показано на рис. 6.6, код P в РБ2 БАР передается со сдвигом влево на два разряда. С выходов РБ1 и РБ2 эти коды затем подаются на входы

ПСМ3. Такая передача двухрядного кода со сдвигом влево используется, например, при делении для сдвига очередного остатка.

Необходимость двухэтапного формирования кодов S и P 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.

На рис. 6.7 представлена функциональная схема i-го разряда первой ступени двоичного сумматора, реализованного на полупроводниковых интегральных элементах типа ЭСЛ. При построении схемы принято, что единичным значениям x_i, y_i, z_i, s'_i, p'_i, s_i и p_i соответствуют сигналы низкого уровня.

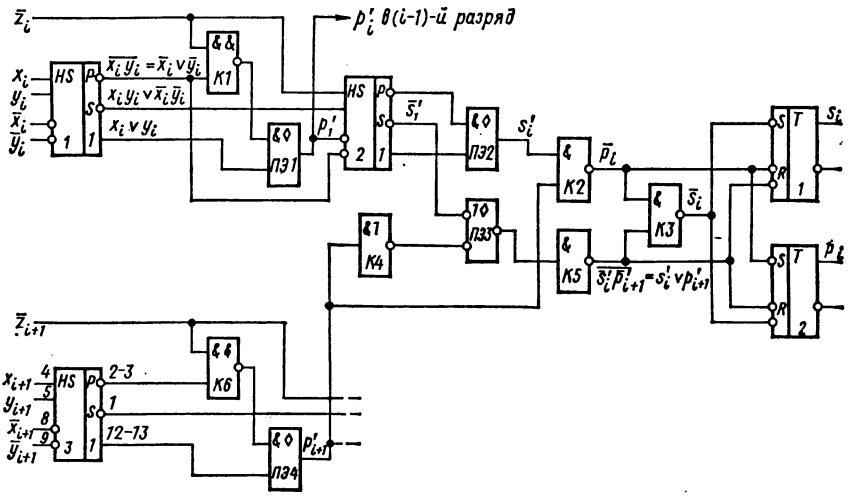


Рис. 6.7. Функциональная схема одного разряда первой ступени двоичного сумматора

С помощью полусумматоров HS1 и HS3 и конъюнкторов K1 и K6 формируются переносы p'_i и p'_{i+1}, а также инверсии сумм по mod 2 $x_iy_i \vee \bar{x}_i\bar{y}_i$ и $x_{i+1}y_{i+1} \vee \bar{x}_{i+1}\bar{y}_{i+1}$. Перенос $p'_i(p'_{i+1})$ получается на объединенных выходах конъюнктора K1 и полусумматора HS1 (K6 и HS3). В полусумматорах используются прямые дизъюнктивные выходы *1*, на которых формируются дизъюнкции $x_i \vee y_i$ и $x_{i+1} \vee y_{i+1}$. На выходе K1 (K6) сигналы соответствуют логическому соотношению

$$\overline{\overline{x_iy_i}\bar{z}_i} = \overline{(\bar{x}_i \vee \bar{y}_i)\bar{z}_i} \quad \overline{(\overline{x_{i+1}y_{i+1}}\bar{z}_{i+1} = \overline{(\bar{x}_{i+1} \vee \bar{y}_{i+1})\bar{z}_{i+1}}).}$$

Соединение накоротко (в точку) выходов *1* HS1 и K1 дает возможность получить перенос p'_i в виде

$$p'_i = \overline{(\bar{x}_i \vee \bar{y}_i)\bar{z}_i}(x_i \vee y_i) = x_iy_i \vee x_iz_i \vee y_iz_i.$$

Аналогично вырабатывается и p'_{i+1}.

Инверсия $\overline{s}'_i$ формируется непосредственно на выходе S полусумматора HS_2, т. е.

$$\overline{s}'_i = (x_i y_i \vee \overline{x}_i \overline{y}_i)\overline{z}_i \vee \overline{(\overline{x}_i \vee \overline{y}_i)\overline{z}_i}(x_i \vee y_i)(\overline{x}_i \vee \overline{y}_i), \ldots,$$

а s'_i – путем соединения в точку его выходов P и 1:

$$s'_i = \overline{(x_i y_i \vee \overline{x}_i \overline{y}_i \vee \overline{z}_i)(\overline{x}_i \vee \overline{y}_i \vee \overline{(\overline{x}_i \vee \overline{y}_i)\overline{z}_i}(x_i \vee y_i))}.$$

Элементы функциональной схемы рис. 6.7, с помощью которых вырабатываются p'_i, s'_i и $\overline{s}'_i$, относятся к ПСМ3 (см. рис. 6.6). Схему полусумматора ПСМ2 в каждом разряде составляют элементы, с помощью которых образуются $\overline{s}_i$ и $\overline{p}_i$. Получаемое на выходе К3 значение $\overline{s}_i$ соответствует выражению

$$\overline{s}_i = \overline{s'_i p'_{i+1} \overline{s}'_i \overline{p}'_{i+1}},$$

а на выходе К2

$$\overline{p}_i = \overline{s'_i p'_{i+1}}.$$

Нетрудно убедиться, что эти логические функции представляют собой инверсии выражений (6.3) и (6.4).

Сигналами, соответствующими $\overline{s}_i$ и $\overline{p}_i$, 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 p_i, или в целом S и P осуществляются 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 P'. Для этой цели оказалось удобнее использовать дополнительный полусумматор на три входа ПСМ3′ (см. рис. 6.6), в общих чертах повторяющий ПСМ3, но имеющий несколько большую скорость работы. По полученным для всех разрядов значениям s'_i и p'_i для каждой j-й 8-разрядной группы сумматора вырабатываются групповой перенос из ее второй тетрады $p_{T_2 j}$, тетрадное разрешение переноса $r_{T_2 j}$, групповой перенос из j-го байта p_{6j} и байтовое разрешение переноса r_{6j} (рис. 6.8). Сигналы, соответствующие этим значениям, формируются в схеме групповых переносов СГП согласно формулам

$$p_{T_2 j} = s'_{j4} p'_{j5} \vee (s'_{j4} \vee p'_{j5}) s'_{j5} p'_{j6} \vee (s'_{j4} \vee p'_{j5})(s'_{j5} \vee p'_{j6}) s'_{j6} p'_{j7} \vee \vee (s'_{j4} \vee p'_{j5})(s'_{j5} \vee p'_{j6})(s'_{j6} \vee p'_{j7}) s'_{j7} p'_{(j+1),0}; \quad (6.5)$$

$$r_{T_2 j} = (s'_{j4} \vee p'_{j5})(s'_{j5} \vee p'_{j6})(s'_{j6} \vee p'_{j7})(s'_{j7} \vee p'_{(j+1),0}); \quad (6.6)$$

$$p_{6j} = s'_{j0} p'_{j1} \vee (s'_{j0} \vee p'_{j1}) s'_{j1} p'_{j2} \vee (s'_{j0} \vee p'_{j1})(s'_{j1} \vee p'_{j2}) s'_{j2} p'_{j3} \vee \vee (s'_{j0} \vee p'_{j1})(s'_{j1} \vee p'_{j2})(s'_{j2} \vee p'_{j3}) s'_{j3} p'_{j4} \vee (s'_{j0} \vee p'_{j1})(s'_{j1} \vee p'_{j2})(s'_{j2} \vee \vee p'_{j3})(s'_{j3} \vee p'_{j4}) p_{T_2 j}; \quad (6.7)$$

$$r_{6j} = (s'_{j0} \vee p'_{j1})(s'_{j1} \vee p'_{j2})(s'_{j2} \vee p'_{j3})(s'_{j3} \vee p'_{j4}) r_{T_2 j}, \quad (6.8)$$

где цифровой индекс, следующий за индексом j, указывает номер разряда в j-м байте; $j = 0, 1, 2, \ldots, 7$.

Величина $p'_{(+1)}$ 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.

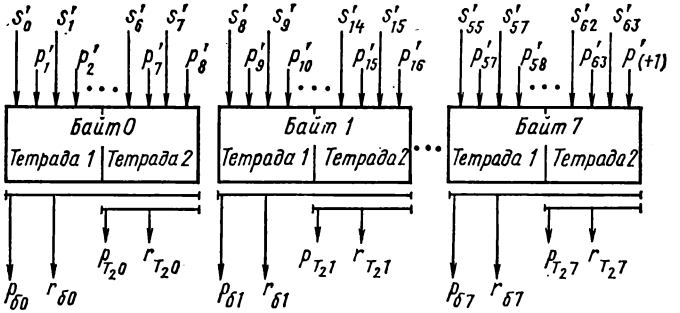


Рис. 6.8. Распределение входных и выходных сигналов по байтам сумматора в схеме формирования групповых переносов СГП

Значения тетрадных и байтовых переносов, а также разрешения тетрадных и байтовых переносов, образующие 8-разрядные коды соответственно $P_Т = p_{Т_20}p_{Т_21}\ldots p_{Т_27}$, $P_б = p_{б0}p_{б1}\ldots p_{б7}$, $R_Т = r_{Т_20}r_{Т_21}\ldots r_{Т_27}$ и $R_б = r_{б0}r_{б1}\ldots r_{б7}$, заносятся в соответствующие регистры (см. рис. 6.6). При определении окончательной суммы в сумматоре СМ ступени II двоичного сумматора БАС эти значения совместно с полусуммой S и переносами P используются для формирования по каскадно-параллельной схеме входных переносов $p_{вх.i}$ в каждый разряд сумматора, где $i = 0, 1, 2, \ldots, 63$. По полученным входным переносам $p_{вх.i}$ и значениям полусумм s_i полное приведение переносов осуществляется путем суммирования по mod 2 s_i и $p_{вх.i}$, т. е.

$$s_{iк} = s_i \oplus p_{вх.i}. \tag{6.9}$$

Значения $s_{iк}$, полученные для всех разрядов, образуют код окончательной суммы $S_к$, который заносится в регистр результата РР (см. рис. 6.6).

По каскадно-параллельной схеме нулевого разряда сумматора СМ может достичь, например, либо любой перенос p_1, p_2 или p_3, возникающий в первой тетраде нулевого байта (рис. 6.8), либо групповой тетрадный перенос $p_{Т_20}$ из второй тетрады этого же байта, либо входной перенос нулевого байта $p_{вх.60}$, переданный в данный байт из других групп, т. е.

$$p_{вх.0} = p_1 \vee s_1p_2 \vee s_1s_2p_3 \vee s_1s_2s_3p_{Т_20} \vee s_1s_2s_3r_{Т_20}p_{вх.60}, \tag{6.10}$$

где

$$p_{\text{вх. б0}} = p_{б1} \vee r_{б1}p_{б2} \vee r_{б1}r_{б2}p_{б3} \vee \ldots \vee r_{б1}r_{б2}r_{б3}r_{б4}r_{б5}r_{б6}p_{б7} \vee$$

$$\vee\ r_{б1}r_{б2}r_{б3}r_{б4}r_{б5}r_{б6}r_{б7}p_{(+1)};$$

$p_{(+1)}$ – дополнительная единица в 63-й разряд, которую можно рассматривать как входной перенос 7-го байта.

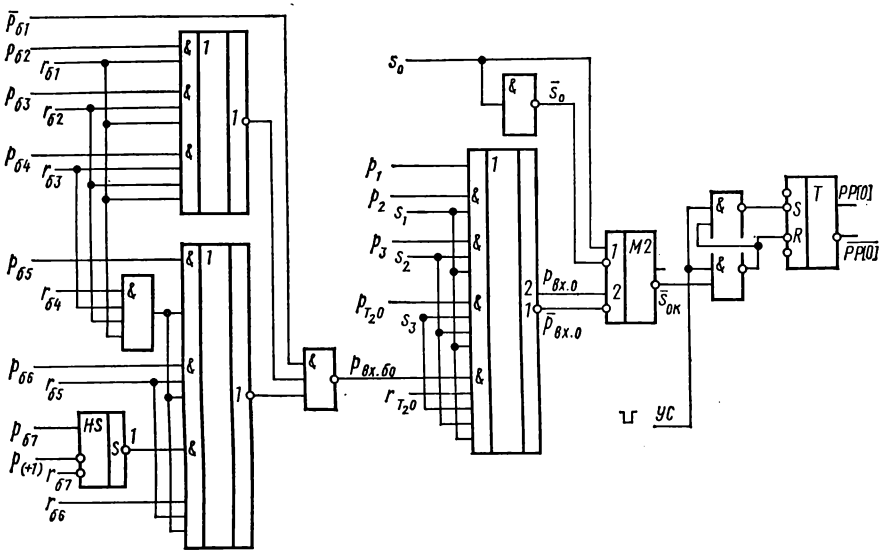


Рис. 6.9. Схема формирования результатов в нулевом разряде сумматора второй ступени

Функциональная схема формирования окончательного значения $s_{ок}$ в нулевом разряде суммы $S_к$ приведена на рис. 6.9. Левая часть этой схемы с полусумматором HS предназначена для образования входного переноса нулевого байта $p_{вх.б0}$, правая – для получения входного переноса нулевого разряда $p_{вх.0}$ и его суммирования по mod 2 с s_0 согласно (6.9). 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 рассмотренной каскадно-параллельной схеме СМ ступени II 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.

Сумматор с полным приведением переносов по каскадно-параллельной схеме может быть также построен на интегральных схемах АЛС и СМГП типа рассмотренных в гл. 4. Пример многоразрядного арифметико-логического устройства такого типа рассматривается в следующем параграфе. При использовании интегральных схем повышенной степени

интеграции на токовых переключателях схема суммирования получается более быстродействующей при меньших аппаратурных затратах.

В блоке десятичной арифметики БАЦ основу операционной части составляет десятичный сумматор ДСМ (рис. 6.10). Это – 8-разрядный комбинационный сумматор, позволяющий складывать как положительные, так и отрицательные десятичные числа, представленные в двоично-десятичной форме в коде с избытком 6. Отрицательные числа при сложении и вычитании представляются 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 $p_{(+1)} = 1$. 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 одинаковых 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 значение межтетрадного p_{T_2} и межбайтного $p_б$ 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 $p_б$ и p_{T_2} 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 ёе старшего разряда. Поэтому, если равен единице p_{T_2} или $p_б$, 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

$$s_{iк} = (s'_i \bar{p}_{i+1} \vee \bar{s}'_i p_{i+1}) c_1, \tag{6.11}$$

где $s'_i = x_i y_i \vee \bar{x}_i \bar{y}_i$ берется непосредственно с выхода S полусумматора HS1, а $s'_j = (x_i \vee y_i)\overline{x_i y_i} = x_i \bar{y}_i \vee \bar{x}_i y_i$ получается путем соединения выходов P и 1 (монтажное *И*) того же полусумматора; p_{i+1} — входной перенос в i-й разряд, формируемый из предыдущих разрядов параллельными цепями; c_1 — синхронизирующий сигнал.

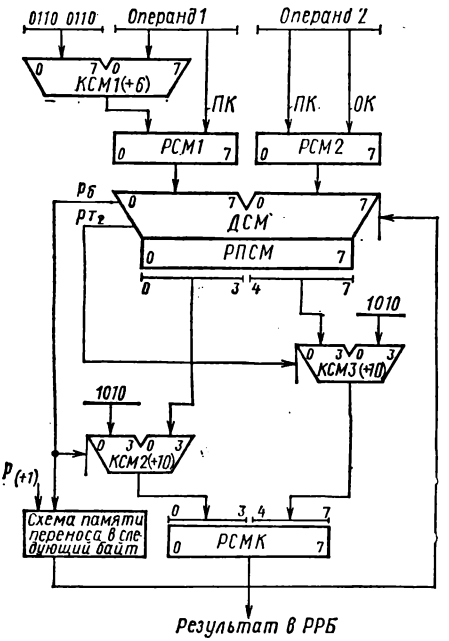


Рис. 6.10. Структурная схема десятичного сумматора

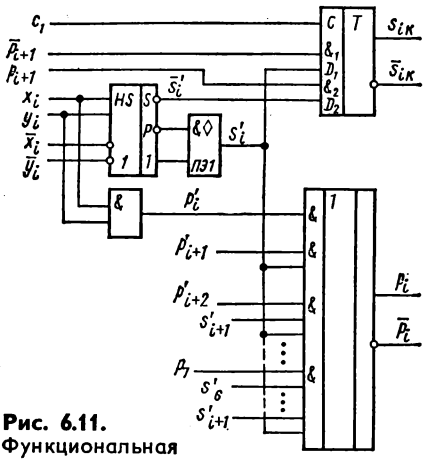


Рис. 6.11. Функциональная схема одного разряда ДСМ

Структура параллельных цепей переноса показана на примере формирования окончательного значения переноса в i-м разряде. Как следует из схемы,

$$p_i = p'_i \vee s'_i p'_{i+1} \vee s'_i s'_{i+1} p'_{i+2} \vee \ldots \vee s'_i s'_{i+1} \ldots s'_6 p_7, \tag{6.12}$$

где

$$p'_i = x_i y_i;$$

$$p_7 = x_7 y_7 \vee s'_7 (p_6 \vee p_{(+1)}).$$

Для выполнения логических операций, таких как *И*, *ИЛИ*, *Исключающее ИЛИ*, 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:

$$F_{1i} = x_i y_i; \tag{6.13}$$

$$F_{2i} = x_i \vee y_i; \tag{6.14}$$

$$F_{3i} = x_i\bar{y}_i \vee \bar{x}_i y_i. \tag{6.15}$$

Результаты логических действий над одноименными разрядами очередной пары байтов операндов заносятся непосредственно в регистр результата обработки байтов РРБ.

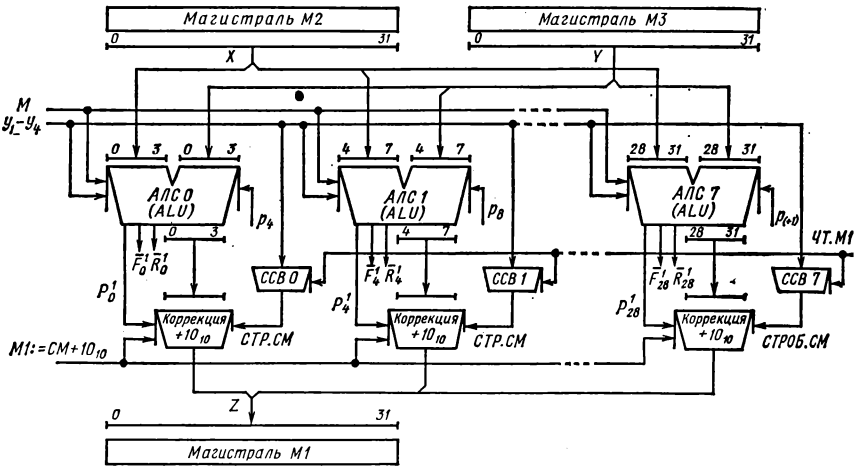


Рис. 6.12. Структурная схема БАЛ

6.4. Операционная часть арифметико-логического устройства процессора ЕС-2433

Все арифметические и логические действия над операндами (за исключением преобразования из одной системы счисления в другую) в процессоре ЕС-2433 выполняются с помощью 32-разрядного блока БАЛ (рис. 6.4) комбинационного типа. Этот блок реализован на арифметико-логических интегральных схемах АЛС, рассмотренных в п. 4.7. Применение этих схем обусловливает групповую структуру БАЛ. Разряды с 0-го по 31-й распределены по восьми 4-разрядным группам, как показано на рис. 6.12.

Двоичные и двоично-кодированные десятичные числа подаются с магистралей М2 и М3 на входы АЛС0—АЛС7. С каждой магистрали одновременно может поступать до 32 разрядов. Арифметико-логические схемы принимают кодовые сигналы в инверсной форме. Поэтому они функционируют согласно табл. 4.7 и их включение соответствует условному изображению рис. 4.44, *б*.

Действие, выполняемое блоком БАЛ, определяется управляющими сигналами $У_1 - У_4$, поступающими из регистра микрокоманд блока задания режимов работы процессора. При этом переключение на действие — арифметическое или логическое — осуществляется сигналом *М*, также

выдаваемым блоком режимов работы. Принципы выполнения различных арифметических и логических микроопераций в АЛС также подробно рассмотрены в п. 4.7.

Результаты микроопераций с выходов АЛС0—АЛС7 выдаются в магистраль М1 через блоки "Коррекция $+10'_{10}$. При выполнении действий

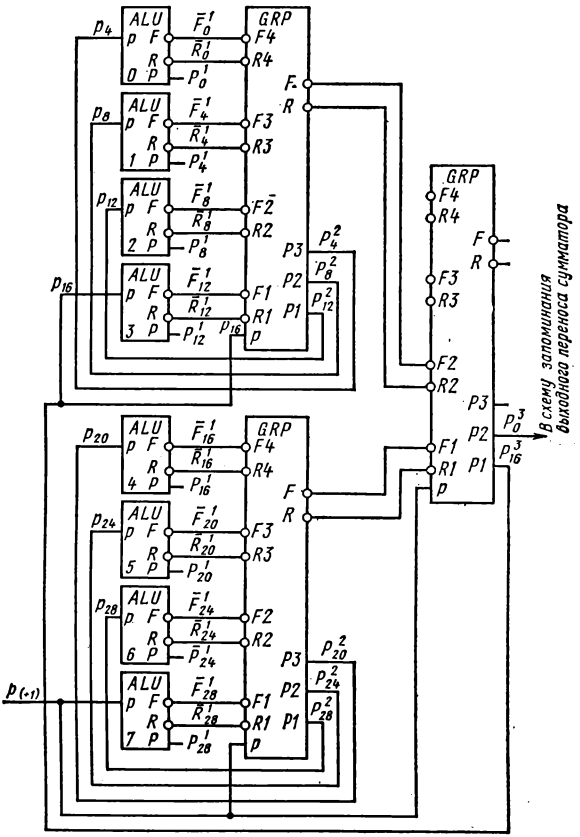


Рис. 6.13. Схема цепей межтетрадных переносов

над двоичными числами эти блоки работают в пассивном режиме. В этом случае тетрады результатов проходят через них по сигналам СТРОБ СУММАТОРА *СТРОБ СМ*, не претерпевая изменений. Сигналы стробов выдаются схемами стробов выдачи ССВ0—ССВ7. Для каждой тетрадной группы имеется своя схема ССВ. Управляют работой этих схем сигналы $У_1—У_4$ и сигнал ЧТЕНИЕ в первую магистраль ЧТ М1. Выдача в магистраль М1 запрещается, когда выполняются микрооперации приема в магистраль М1 информации из операционного РОП*i* или информационного РИП регистров процессора (в обоих случаях нет разрешающего сигнала ЧТ М1), либо из блока режимов работы поступает комбинация управляющих сигналов $У_4У_3У_2У_1 = 1101$. Эта комбинация служит для блокировки выдачи результата БАЛ в магистраль М1.

Если производятся действия над двоично-кодированными десятичными операндами, то результаты в магистраль М1 выдаются по сигналу $М1 := СМ + 10_{10}$, поступающему из блока режимов работы процессора. Этим сигналом схемы коррекции активизируются. При отсутствии сигнала переноса P_i^1 из i-й АЛС к тетраде результата прибавляется двоичный код 1010 ($+10_{10}$), что равносильно изъятию избытка 6.

Для ускорения распространения переносов при микрооперациях суммирования (заемов – при вычитании) межтетрадные переносы P_i^2 формируются с помощью схем межгрупповых переносов, как показано на рис. 6.13. Выходы инверсий сигналов формирования $\overline{F_i^1}$ и распространения $\overline{R_i^1}$ переносов арифметико-логических схем (ALU) соединены с входами схем межгрупповых переносов (GRP) первой ступени. В свою очередь, аналогичные выходы схем GRP первой ступени соединены с входами схемы межгрупповых переносов второй ступени. На выходе последней формируется выходной перенос P_0^3 32-разрядного сумматора. Этот перенос используется схемами анализа процессора при выработке признака результата (кода условия), либо просто запоминается и подается как сигнал $p_{(+1)}$ на вход сумматора при сложении 32-разрядных чисел и чисел более высоких разрядов. На схеме при изображении блоков ALU не показаны информационные и управляющие входы и выходы. Номера схем ALU и сигналов даны в соответствии с нумерацией разрядов операндов, приведенной на рис. 6.12.

Так как в противоположность номерам разрядов веса их возрастают справа налево, то распространение переносов идет в направлении от разрядов со старшими номерами к разрядам с младшими номерами. Организация каскадно-параллельного переноса обеспечивает завершение процессов сложения 32-разрядных операндов не более чем за 120 нс после подачи их разрядов на входы арифметико-логических схем.

6.5. Арифметико-логический блок процессора ЕС-2622

В 16-разрядном 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 мультиплексного канала.

Арифметико-логический блок рассчитан на выполнение 16 элементарных операций (микроопераций), перечисленных в табл. 6.2. Информация в АЛБ может обрабатываться либо побайтно, либо от каждого операнда в микрооперации одновременно могут участвовать по два байта. Поэтому в табл. 6.2 под X и Y можно понимать либо коды байт соответственно 1-го и 2-го операндов, либо пары байт. Перед началом очередной микрооперации эти коды заносятся соответственно в регистры А и Б – входные регистры АЛБ (см. рис. 6.5).

Микропрограммы всех операций, определяемых системой команд ЕС ЭВМ, действий над адресами, обмена и пр., составляются из микрокоманд перечисленных выше элементарных операций. Одна такая микрооперация

Таблица 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	$Y - \underline{X}$ (дв)	Вычитание (двоичное) 1-го операнда
1000	$X \vee \overline{Y}$	Поразрядное логическое сложение прямого кода 1-го операнда с обратным кодом 2-го
1001	Y ТРАНЗИТ	Транзитная пересылка 2-го операнда
1010	$\underline{X} \oplus Y$	Поразрядная сумма по модулю 2
1011	$\overline{X} \vee Y$	Поразрядное логическое сложение обратного кода 1-го операнда с прямым кодом 2-го
1100	Сдв. Y 1П	Сдвиг кода Y в РБ на один разряд вправо
1101	Сдв. Y 1Л	Сдвиг кода Y в РБ на один разряд влево
1110	$X + Y$ (дес)	Сложение (десятичное)
1111	$X + Y$ (дв)	Сложение (двоичное)

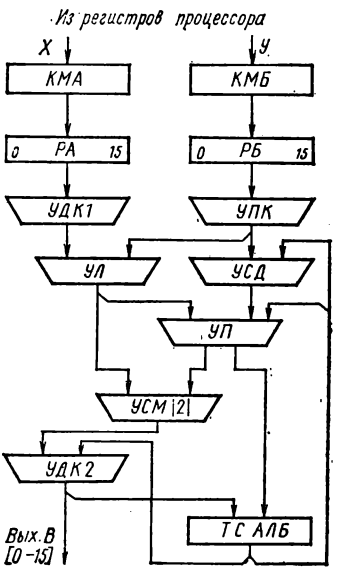


Рис. 6.14. Арифметико-логический блок

выполняется в АЛБ за один машинный такт, равный 550–600 нс. 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):

коммутаторы магистралей КМА и КМБ;

входные регистры РА и РБ;

первый и второй десятичные корректоры УДК1 и УДК2;

узел перекосов УПК;

логический узел УЛ;

узел сдвига УСД;

узел переносов УП;

узел суммирования по mod 2 УСМ |2|;

триггеры состояния (признаков результата) арифметико-логического блока ТС АЛБ.

Кроме регистров, все узлы, составляющие АЛБ,— комбинационные.

Коммутаторы магистралей КМА и КМБ обеспечивают выделение и передачу из регистров процессора в регистры РА и РБ очередных байтов операндов, подлежащих обработке в АЛБ.

Входные двухбайтные регистры РА и РБ предназначаются для хранения очередных операндов, обрабатываемых в АЛБ. Каждый из них состоит из двух однобайтных частей соответственно РА0, РА1 и РБ0, РБ1.

Первый десятичный корректор УДК1 при выполнении операций над десятичными данными используется для внесения в тетрады байтов 1-го операнда, поступающих с РА0 и РА1, избытков +6 (0110). В остальных арифметических и логических микрооперациях над двоичными операндами информация при передаче через УДК1 не претерпевает изменений.

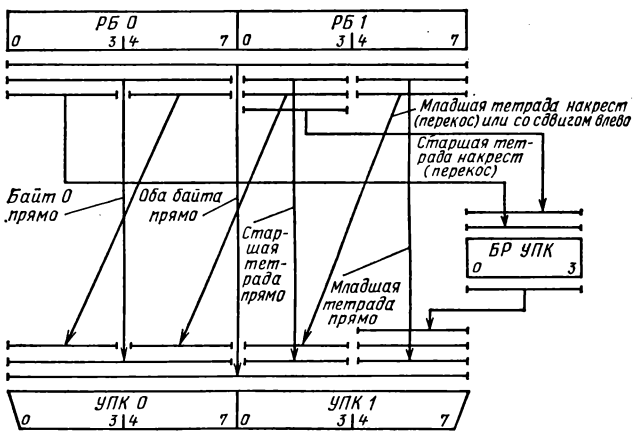


Рис. 6.15. Информационные связи регистров Б с УПК

Узел перекосов УПК служит для передачи информации из регистра РБ либо без изменений (прямо), либо с изменением взаимного расположения и даже сочетания тетрад байтов (рис. 6.15). Тетрады байта 1 могут меняться либо только местами, либо только одна из них может передаваться прямо или накрест, а вместо другой при этом следуют нули. Кроме этого, может осуществляться пересылка тетрад обоих байтов со смещением, получившим название перекоса. При выполнении его младшие три тетрады, выдаваемые из РБ, смещаются влево на четыре разряда, а старшая запоминается в буферном регистре БР УПК. Буферный регистр — двухступенчатый. Поступающая в него старшая тетрада запоминается в ступени I, а в ступени II в это время еще хранится тетрада из предыдущего байта. Она в данном такте выдается на входы младших разрядов УПК. По окончании выдачи информации в УПК содержимое ступени I БР переписывается в ступень II. Передача с перекосом используется для осуществления сдвигов влево сразу на четыре разряда. Такой сдвиг может осуществляться только в пределах младшего байта, если производится побайтная обработка информации.

Логический узел УЛ предназначен для выполнения поразрядных логи-

ческих 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 полусумма $x_i \oplus y_i$. 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 просто передачи.

Узел сдвига УСД служит для организации сдвигов вправо на два разряда байт данных, поступающих в АЛБ через регистр Б. Полностью эта микрооперация реализуется узлом сдвига совместно с узлом переносов УП. При арифметическом сложении байты второго слагаемого проходят через УСД без изменения, а при вычитании типа $X - Y$ они поразрядно инвертируются. В логических операциях узел сдвига для информации всегда закрыт (на выходах всех его разрядов выдаются сигналы кода 0).

Узел переносов УП в первую очередь предназначается для выработки межразрядных и межбайтных переносов. Эти переносы затем используются в узле суммирования по mod 2 для получения окончательных значений арифметической суммы или разности.

Межразрядные переносы p_i вырабатываются согласно логическим формулам

$$p_i = \overline{s}'_{i+1} y_{i+1} \vee s'_{i+1} p_{i+1}, \qquad (6.16)$$

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; p_{i+1} — перенос в $(i+1)$-й разряд; $i = 0, 1, 2, \ldots, 15$.

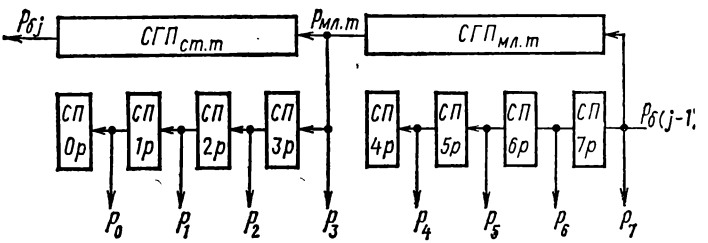


Рис. 6.16. Схема распространения переносов

Переносы между разрядами (между разрядными схемами формирования переносов СП на рис. 6.16) в пределах тетрад распространяются последовательно. Кроме того, имеются схемы групповых переносов тетрад (например, $СГП_{мл.т}$ и $СГП_{ст.т}$ в пределах одного из байт). Межтетрадный ($p_{мл.т}$) и межбайтный ($p_б$) переносы формируются этими схемами по принципу параллельного распространения переносов. Например, для младшей тетрады формула, согласно которой получается $p_{мл.т}$, имеет вид

$$p_{мл.т} = \overline{s}'_4 y_4 \vee s'_4 \overline{s}'_5 y^5 \vee s'_4 s'_5 \overline{s}'_6 y_6 \vee s'_4 s'_5 s'_6 \overline{s}'_7 y_7 \vee s'_4 s'_5 s'_6 s'_7 p_{7}. \qquad (6.17)$$

где p_7 — перенос в 7-й разряд, являющийся либо межбайтным переносом $p_{б(j+1)}$ из предыдущего $(i+1)$-го байта, либо дополнительной единицей, подсуммируемой к младшему разряду при образовании дополнительного кода. Перенос $p_{бj}$, формируемый схемой групповых переносов старшей тетрады, является межбайтным переносом из j-го байта в $(j-1)$-й. Он запоминается в триггере регистра состояния арифметико-логического блока РС АЛБ и при сложении кодов следующих байт передается в младший разряд как p_7. Цепи формирования и распределения всех переносов — парафазные.

Кроме выработки переносов, узел УП совместно с УСД используется для сдвига кодов, поступающих на вход АЛБ через РБ, на один разряд 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 удалился автоматически при действии над двоично-десятичными кодами в узлах УЛ, УП и УСМ |2|.

Во всех остальных микрооперациях, не относящихся к десятичным, УДК2 не меняет значение проходящей через него информации.

Триггеры состояния арифметико-логического блока ТС АЛБ предназначены для запоминания признаков, характеризующих результат действия в АЛБ над байтами данных. Эти признаки — межбайтные переносы, знак результата, указатели переполнения, правильности десятичных данных и т. п. — используются для обеспечения правильной обработки последующих байтов как при непрерывной работе АЛУ, так и после прерывания. В случае прерывания исполняемой микропрограммы содержимое ТС через УДК2 заносится в специальный 4-байтный регистр состояния РС процессора, а оттуда либо в локальную память, либо через АЛБ в основную оперативную память. Если необходимо продолжить прерванную микропрограмму, то состояние АЛБ восстанавливается засылкой в ТС запомненного ранее состояния. Признаки состояния вводятся в АЛБ через регистр РА и, следуя транзитом через УДК2, заносятся в ТС.

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 пере-

сылочных 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. Рассмотрим работу АЛБ на примере побайтной обработки информации. Пусть, например, складываются два 8-разрядных двоичных числа: $X[0-7] = 0111\,1101$ и $Y[0-7] = 1111\,1011$. Сложение будет происходить так, как показано на рис. 6.17.

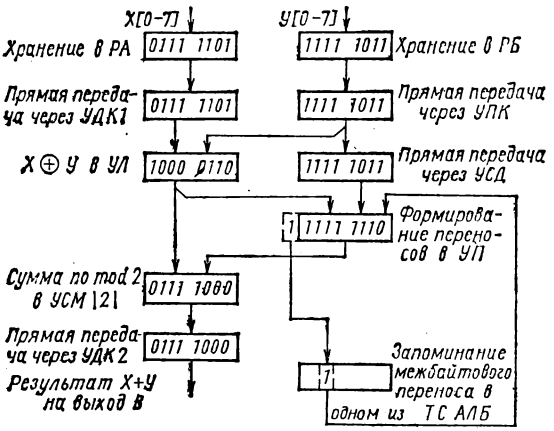


Рис. 6.17. Схема процесса сложения в АЛБ 8-разрядных двоичных кодов

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 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 $X \oplus \overline{Y}$ или $\overline{X} \oplus Y$ определяется следующим образом.

Так как

$$x_i \oplus \overline{y}_i = \overline{x}_i \oplus y_i = x_i y_i \vee \overline{x}_i \overline{y}_i = \overline{s}'_i, \tag{6.18}$$

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

$$p_i = \overline{s}'_{i+1} \overline{y}_{i+1} \vee \overline{s}'_{i+1} p_{i+1}. \tag{6.19}$$

Пусть, например, требуется вычесть из двоичного числа $X[0-7] = 0111\,1111$ двоичное число $Y[0-7] = 0000\,0111$. В результате выполнения в УЛ поразрядной операции равнозначности получится

$$\begin{array}{r} 0111\,1111 \\ \sim\ 0000\,0111 \\ \hline \overline{S'[0-7]} = 1000\,0111. \end{array}$$

По этому результату совместно с инвертированным значением $[Y]_{обр} = \overline{Y[0-7]} = 1111\,1000$ согласно (6.19) сформируется код переносов:

$$\begin{array}{r} 1000\,011\,1 \text{ – код на выходе УЛ } (\overline{S'[0-7]}) \\ 1111\,100\,0 \text{ – код на выходе УСД } (\overline{Y[0-7]}) \\ \hline \boxed{1}\,1111\,111\boxed{1} \text{ – код переносов на выходе УП.} \end{array}$$

Выделенная рамкой младшая единица в коде переноса является дополнительной единицей, которую необходимо учитывать при образовании дополнительного кода $[Y]_{обр}$. Старшая выделенная единица – межбайтовый перенос. В результате сложения в УСМ |2| получится следующий результат:

$$\begin{array}{r} 1000\,0111 \text{ – код на выходе УЛ} \\ \oplus\ 1111\,1111 \text{ – код на выходе УП} \\ \hline 0111\,1000 \text{ – результат операции } X - Y \text{ (дв).} \end{array}$$

При выполнении операции $Y - X$ (дв) код уменьшаемого Y проходит через УСД, не инвертируясь. Межразрядные переносы в УП формируются согласно формуле

$$p_i = \overline{s}'_{i+1} y_{i+1} \vee \overline{s}'_{i+1} p_{i+1}. \qquad (6.20)$$

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.

$$\overline{S'[0-7]} = X[0-7] \sim Y[0-7] = 1000\,0111.$$

Согласно (6.20) в УП сформируется код переносов:

$$\begin{array}{r} 1000\,011\,\ 1 \text{ – код на выходе УЛ } (\overline{S'[0-7]}) \\ 0000\,011\,\ 1 \text{ – код на выходе УСД } (Y[0-7]) \\ \hline \boxed{0}\,0000\,111\boxed{1} \text{ – код переносов на выходе УП.} \end{array}$$

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:

$$\begin{array}{r} 1000\,0111 \text{ – код на выходе УЛ} \\ \oplus\ 0000\,1111 \text{ – код на выходе УП} \\ \hline 1000\,1000 \text{ – результат операции } Y - X \text{ (дв).} \end{array}$$

Выполнение операций над десятичными операндами во многом аналогично рассмотренному выше. Добавляются лишь дополнительные кор-

ректирующие 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 РБ, 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_i следующим через УСД. На выходах УП согласно (6.16) при $s'_i = 0$ имеем

$$p_i = \overline{0} y_{i+1} = y_{i+1}.$$

Например, пусть сдвигается код $Y_2 = 1111\,1101$. В результате действий в разрядах УП получим

0000 000 0 — код на выходе УЛ

1111 110 1 — код на выходе УСД

1 1111 101 0 — результат сдвига влево.

$p_{бj}$ $p_{б(j+1)}$

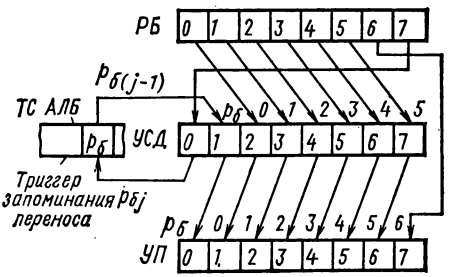


Рис. 6.18. Схема сдвига на разряд вправо

При сдвиге влево кодов, состоящих из нескольких байтов, значение разряда, соответствующего межбайтному переносу $p_{бj}$, запоминается в одном из триггеров ТС 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 код, проходя УСД, предварительно смещается на два разряда вправо, как показано на рис. 6.18. При этом содержимое 6-го разряда РБ сразу поступает в цепь входящего переноса УП, а 7-го — в 0-й разряд УСД. С выхода УСД оно заносится в схему памяти межбайтового переноса как $p_{бj}$ и будет использоваться при сдвиге соседнего справа $(j-1)$-го байта. В то же время значение 7-го разряда предыдущего $(j-1)$-го байта как $p_{б(j-1)}$ подается из схемы памяти на вход 1-го разряда УСД. Для того чтобы можно было одновременно запоминать $p_{бj}$ и выдавать $p_{б(j-1)}$, 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 операции 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 режим работы АЛБ — однобайтный или двухбайтный — и номера регистров, из которых должна приниматься информация в регистры РА и РБ и куда должен выдаваться результат с выхода *В*. Имеется в ней также специальное поле, код в котором определяет способ пересылки очередного байта из РБ через УПК на входы УЛ и УСД (например, осуществить ли прямую передачу или передать тетрады байта 1 накрест, или передать их с перекосом и т. п.).

Глава 7

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 программного обеспечения и рабочие программы, организует все необходимые действия по приему, оценке и преобразованию исходной информации, по получению результирующих данных и выдаче их потребителям.

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 главным образом центральным устройством управления (ЦУУ). В связи с этим ЦУУ может считаться преобразователем первичной командной информации, представляемой командами программы, во вторичную командную информацию, представляемую формируемыми ЦУУ исполнительными адресами, кодами и управляющими сигналами, воздействие которых на соответствующие узлы и блоки устройств машины приводит к выполнению заданных операций. Помимо команд, к первичной командной информации относятся также сигналы и коды, характеризующие состояние отдельных блоков и устройств.

Выполнение, или реализация команды обычно проводится в следующей последовательности:

выборка команды из оперативной памяти;

формирование исполнительных адресов операндов по информации, содержащейся в коде команды;

выборка операндов из оперативной или местной (локальной) памяти;

выполнение действий в арифметических блоках (устройствах), именуемых также арифметическо-логическими;

отсылка результата выполнения операции в оперативную или местную память.

Каждый из указанных этапов действий несколько различается для команд разных форматов и разных моделей ЭВМ; различия в выполнении поэтапных действий разных моделей ЭВМ вызываются принимаемыми принципами построения процессоров, форматами слов на входе и выходе оперативной памяти, т. е. шириной выборки, 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 рассматриваемых машинах используется в основном двухадресная система команд, причем только для формата RR принята прямая адресация. Компоненты исполнительных адресов для случаев относительной адресации (форматы RX, RS, SI, SS) частично содержатся в кодах команд, а большей частью размещаются, как и операнды для формата RR, в местной (локальной) памяти, имеющей в общем случае 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 некоторых моделях младших номеров (например, ЕС-1020) она представляет собой часть оперативной памяти с начальными адресами; в моделях старших номеров, а также ЕС-1022, ЕС-1033, она выполняется либо в виде набора регистров общего назначения, либо в виде полупроводниковой памяти малой емкости, часто называемой локальной памятью.

Если все действия, связанные с преобразованием кодов команд в наборы управляющих сигналов и исполнительные адреса, выполняются непосредственно в центральном устройстве управления, то это устройство в общем случае включает 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, непосредственно обеспечивающих преобразование и реализацию кодов команд, в состав ЦУУ еще включаются пульт управления ПУ, блок прерываний БП, блок управления памятью БУП или блок сопряжения с памятью БСП, 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.

Блок выборки команд служит для формирования адреса очередной команды (или очередного участка программы) с помощью суммирующего счетчика и размещения кодов команд в своих регистрах. В машинах Единой системы команды могут быть двухбайтными (формат RR), четырехбайтными (форматы RX, RS, SI) и шестибайтными (формат SS). В то же время выборка информации из оперативной памяти в ЕС-1020 производится по полусловам (2 байта), в ЕС-1022, ЕС-1030, ЕС-1033 – по словам (4 байта), в ЕС-1050, ЕС-1052 – по двойным словам (8 байт). Таким образом, только в случае команд формата RR в ЕС-1020 и форматов RX, RS, SI в ЕС-1022, ЕС-1030 и ЕС-1033 выбираемые из оперативной памяти

коды представляют собой коды соответствующих команд. Во всех остальных случаях код, выбранный из оперативной памяти, представляет собой либо часть команды, либо несколько команд — от двух до четырех, т. е. участок программы.

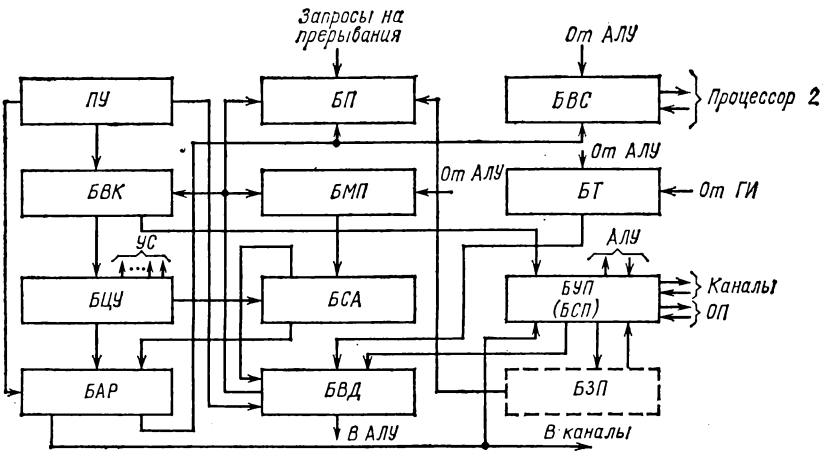


Рис. 7.1. Общая структура центрального устройства управления

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 один или несколько дополнительных циклов обращения к ОП для получения остальных частей команды. Если же реальная длина команды не превышает 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.

Явно выраженные блоки выборки команд имеют центральные устройства управления моделей старших номеров. В моделях младших номеров функции БВК выполняют общие регистры процессора совместно с арифметическо-логическим блоком, обеспечивающим формирование адресов очередных команд. Это связано с тем, что с целью уменьшения общего аппаратурного состава в моделях младших номеров осуществлено совмещение функций отдельных частей процессоров.

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 АЛУ, блок выборки команд, а также в блок прерываний и блок местной памяти. Выполняется БВД в отдельных моделях ЭВМ Единой системы аналогично блоку выборки команд.

В БАР 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; в последнем случае рассматриваемый блок относится к числу неявно выраженных блоков.

Блок сумматора адреса служит для формирования исполнительных адресов операндов по информации, содержащейся в коде команды. В предельном случае исполнительный адрес формируется путем сложения трех составляющих: базового адреса, выбираемого из местной (локальной) памяти, 12-разрядного кода смещения, содержащегося в коде команды, и индекса, или модификатора, выбираемого из местной (локальной) памяти. Поэтому основу рассматриваемого блока при его выполнении в виде отдельной явно выраженной функциональной части составляет суммирующая схема, обеспечивающая сложение кодов трех чисел. Исполнительные адреса, формируемые в БСА, направляются в блок управления памятью или блок сопряжения с 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 УС этот блок строится либо в виде схем реализации микропрограмм с базовой постоянной памятью, либо в виде дешифратора кода операции и схем формирования этих сигналов. Для многих моделей ЭВМ Единой системы принят микропрограммный принцип формирования УС, а для ЕС-1050 (ЕС-1052) — схемный. Более подробно эти принципы формирования УС и их реализация в различных моделях рассматриваются ниже, в пп. 7.2 и 7.3.

Пульт управления содержит переключатели, кнопки, а также электронные схемы и средства индикации, обеспечивающие управление работой машины со стороны оператора, визуальный контроль состояния отдельных устройств и проведение профилактических мероприятий.

Блок прерываний, функционально выделяемый практически во всех моделях, обеспечивает реализацию запросов на прерывания в соответствии с их приоритетами. Включение в состав центрального устройства управления блока прерываний подчеркивает системные возможности и характеризует, в частности, машины Единой системы как машины третьего поколения.

Особое место в составе центрального устройства управления моделей старших номеров занимает блок управления памятью. Дело в том, что в машинах Единой системы 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 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. В моделях младших номеров аналогичные, но упрощенные функции выполняет блок сопряжения с памятью.

Блок внешних связей и блок таймеров относятся к так называемым системным средствам управления. Первый из них, относимый также к средствам прямого управления, обеспечивает обмен управляющей информацией с другим процессором при построении и работе многопро-

цессорной системы. Блок таймеров служит для подсчета временных интервалов и управления работой процессора в системе по принимаемым временным соотношениям.

Блок защиты памяти обеспечивает выявление случаев обращения к запрещенным для данной рабочей программы участкам или страницам оперативной памяти и организацию соответствующих прерываний в реализации программ. Иногда этот блок включают в состав оперативной памяти.

Рассмотренная общая структура центрального устройства управления является наиболее полной, отвечающей основным принципам организации управления работой машин Единой системы. Реализация же этой структуры различна для различных моделей ЭВМ вследствие различия принимаемых уровней совмещения функциональных частей и принципов формирования управляющих сигналов. При этом действует общий для всех цифровых вычислительных машин закон: чем выше уровень совмещения функциональных частей, т. е. чем меньше аппаратурный состав, тем ниже при прочих равных условиях уровень совмещения во времени отдельных действий при выполнении команд и программ, т. е. тем меньше, в конечном итоге, быстродействие машины. Действие этого закона наглядно иллюстрируется возможностями моделей ЕС-1022 и ЕС-1033, имеющих общую элементную базу и устройства управления, реализующие принципы микропрограммного управления, но различающихся по быстродействию.

Синхронизация. Отличительной особенностью всех моделей ЭВМ Единой системы является использование принципов жесткой синхронизации работы их процессорных частей. Для этого применяются отдельные блоки синхронизации и специальные системы, включающие в свой состав узлы генерирования, распределения и ретрансляции синхронизирующих сигналов.

В машине ЕС-1022, например, в состав центрального устройства управления включен отдельный блок синхронизации, в котором формируются две серии синхроимпульсов, обеспечивающих совместную работу блоков процессора: исходные синхроимпульсы серии С и рабочие (тактовые) импульсы серии ТИ. Каждая серия состоит из четырех основных и четырех так называемых задержанных импульсов; период их следования составляет один машинный такт и равен 550—600 нс. В каждой серии импульсы нумеруются с 1-го по 4-й, например С1, С2, С3, С4, $С1_з$, $С2_з$, $С3_з$, $С4_з$.

Синхроимпульсы серии С вырабатываются непрерывно и используются для формирования тактовых импульсов при наличии потенциалов разрешения РТИ1 и РТИ2, поступающих из узла пуска-останова, входящего в состав блока синхронизации. Длительность, период следования и последовательность тактовых импульсов всегда определяются только параметрами исходных синхроимпульсов серии С. В качестве базовой для всех синхроимпульсов принята длительность, равная 90 нс; задержанные импульсы сдвигаются относительно основных на 70 ± 10 нс; длительность всех импульсов может изменяться в пределах 10 нс; базовый сдвиг между соседними импульсами одной серии, например между С1 и С2, составляет 150 нс.

Формирование импульсов серии С происходит в схемах генератора

исходных синхроимпульсов, включающего в свой состав задающий генератор (ЗГ), который представляет собой автогенератор с линией задержки (ЛЗ) в цепи обратной связи, делители частоты на два ДЧ1, ДЧ2, ДЧ3, триггер формирования длительности такта, или триггер длительности интервала (ТДИ), и выходную вентильную схему, непосредственно обеспечивающую выделение различных импульсов серии С, как это показано на рис. 7.2. Генератор исходных синхроимпульсов не синхронизирован; он начинает формировать синхроимпульсы $C1-C4$, $C1_з-C4_з$ сразу же после подачи на него питающего напряжения; при действии сигнала гашения системы (ГШС) формирование исходных синхроимпульсов не производится.

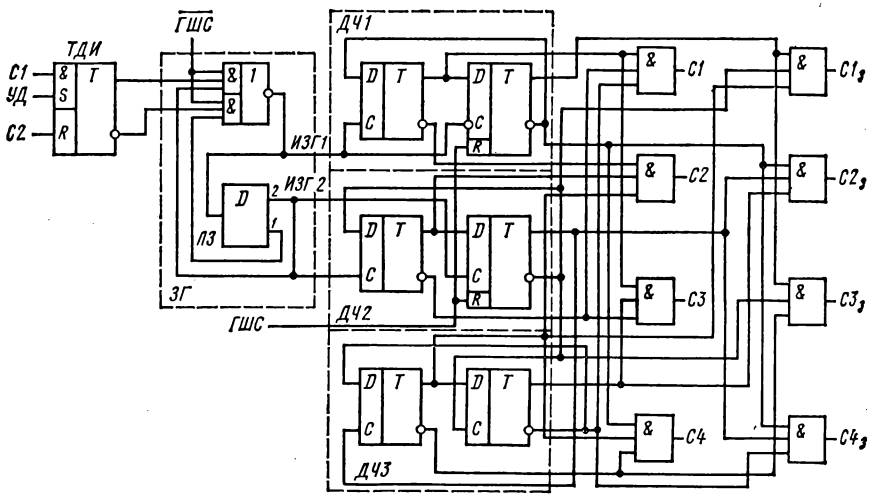


Рис. 7.2. Схема генератора исходных синхроимпульсов

Существенную роль в формировании импульсов серии С по длительности и в организации их временного сдвига играет линия задержки с двумя выходами и общим входом. Сигналы по тракту *1* задерживаются на время порядка 70 нс, а по тракту *2* – порядка 75 нс. В обычном режиме, когда триггер ТДИ находится в нулевом состоянии, работу задающего генератора определяет характер задержки сигналов по тракту *1*; выходные сигналы ЗГ, т. е. ИЗГ1 и ИЗГ2, имеют длительность, равную времени задержки сигналов по этому тракту; сдвиг между сигналами серий ИЗГ1 и ИЗГ2 определяется временем задержки сигналов по тракту *2*. При переключении триггера ТДИ в единичное состояние сигналом УД длительность выходных сигналов ЗГ определяется временем задержки сигналов по тракту *2* линии задержки.

Делители частоты ДЧ1 и ДЧ2 по сигналам серий ИЗГ1 и ИЗГ2 соответственно формируют сигналы длительностью 140 или 150 нс (при единичном состоянии триггера ТДИ); длительность сигналов на выходах ДЧ3 равна 280 или 300 нс. По выходным сигналам делителей частоты

и производится непосредственное формирование исходных синхроимпульсов серии С. Тактовые импульсы вырабатываются по импульсам серии С с помощью двух групп элементов *И* на два входа каждый.

Элементы первой группы, управляемые сигналом РТИ1, по импульсам С1, С2, $С1_3$, $С2_3$ на своих выходах образуют тактовые импульсы ТИ1, ТИ2, $ТИ1_3$ и $ТИ2_3$. Элементы второй группы, управляемые сигналом РТИ2, по импульсам С3, С4, $С3_3$ и $С4_3$ на своих выходах образуют тактовые импульсы ТИ3, ТИ4, $ТИ3_3$, $ТИ4_3$. Сигналы РТИ1 и РТИ2 формируются узлом пуска-останова, отнесенным в ЕС-1022 к блоку синхронизации.

В машинах ЕС-1050 и ЕС-1052 синхроимпульсы используются для жесткого тактирования работы запоминающих элементов типа триггеров и синхронизации работы логических элементов, для которых длительность информационных и управляющих сигналов принята кратной длительности основного такта, т. е. периоду синхроимпульсов. С целью 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 осуществляется централизованно, от единого задающего генератора. Для ретрансляции сигналов синхронизации в каждой стойке и каждой панели процессора машины ЕС-1050 имеется специальный ТЭЗ. Задающий генератор непрерывно формирует в процессе нормальной работы машины начальные синхроимпульсы серии С длительностью 35—40 нс и периодом 80 нс, т. е. $T_{зг} = 12{,}5$ МГц. Одновременно он формирует вспомогательные синхросигналы С0, имеющие ту же длительность, но период, равный 320 нс. При проведении профилактических мероприятий частота задающего генератора с помощью кнопок, расположенных на пульте управления, может изменяться на $\pm 10\,\%$.

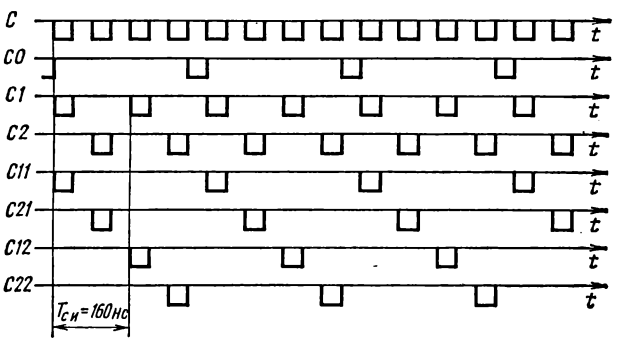


Рис. 7.3. Временная диаграмма сигналов синхронизации

Основными сигналами синхронизации являются сигналы серий С1 и С2, формируемые панельными ТЭЗ синхронизации по сигналам серии С, получаемым из задающего генератора. Эти сигналы имеют частоту, равную 6,25 МГц, при длительности 35—40 нс; длительность такта по основным синхроимпульсам $T_{си} = 160$ нс (см. временную диаграмму на рис. 7.3). Сигналы С0 служат для синхронизации схем выработки последовательностей С1 и С2, находящиеся в панельных ТЭЗ синхронизации.

Сигналы серий С1 и С2 сдвинуты относительно друг друга на полутакт, т. е. на половину периода их следования. Поэтому они непосредственно могут использоваться для синхронизации последовательно включенных триггеров. Но их оказывается недостаточно при организации передач информации по длинным цепочкам. Для этого используются сигналы, имеющие ту же фазу и длительность, что и сигналы серий С1 и С2, но разнесенные по времени на четыре полуконтакта; такими сигналами являются синхроимпульсы серий С11, С21, С12, С22, имеющие увеличенный по сравнению с сигналами серий С1 и С2 период следования и формируемые в панельных схемах формирования синхросигналов.

Организация и совмещение циклов. Под циклом обычно понимают временной интервал, в течение которого осуществляются действия по выборке и выполнению одной команды. Формат отдельных команд, как отмечалось выше, больше формата кодов, выбираемых из оперативной памяти моделей ЭВМ Единой системы младших номеров. Поэтому в течение цикла возможно неоднократное обращение к ОП только по выборке команды. Последовательность действий, выполняемых в течение цикла, зависит еще от уровня совмещения функциональных частей процессора. Таким образом, организация циклов зависит от характеристик процессоров и оперативных запоминающих устройств моделей ЭВМ, а также от вида реализуемых команд и принципов организации местной памяти.

При достаточно высоком уровне совмещения функциональных частей процессора и представлении местной памяти в виде участка оперативной памяти, как это сделано в ЕС-1020, цикл выполнения команды оказывается растянутым и включает многократные обращения к оперативной памяти и арифметическо-логическому блоку. Если, например, выполняется команда формата RR, определяющая некоторую арифметическую операцию, то в течение цикла требуется не менее 4 раз производить обращение 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, включая действия по формированию исполнительного адреса второго операнда. Все действия должны выполняться последовательно в одних и тех же блоках и устройствах, поэтому совмещение циклов невозможно.

В машинах ЕС-1022 и ЕС-1033 двукратное обращение к оперативной памяти для выборки кода команды производится только при выполнении команд формата SS. Но и в случае выполнения команд формата RX или RS в машине ЕС-1022, например, цикл оказывается несколько растянутым из-за необходимости последовательного выполнения действий в АЛБ.

Совмещение действий внутри циклов и самих циклов возможно тогда, когда выполнение каждого этапа реализации команды обеспечивается отдельным функциональным узлом или блоком, а местная память выполнена в виде сверхоперативного запоминающего устройства. Такая возможность имеется, например, в машине ЕС-1050 (ЕС-1052), 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.

Для случая совмещения трех циклов действия по выполнению одной команды (инструкции процессора) разделяются на три уровня:

выборка участка программы из оперативной памяти и его распаковка, т. е. выборка инструкции из этого участка с размещением ее в регистре команд;

формирование исполнительных адресов операндов и выборка операндов из местной или оперативной памяти с размещением их в специальных регистрах ЦУУ;

выполнение действий над операндами в АЛУ и запись результата операции в местную или оперативную память.

Каждому уровню отвечают отдельные функционально независимые узлы и блоки ЦУУ. 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 АЛУ.

Выполнение 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 один от другого и определяющие продолжительность действий на каждом уровне в соответствии с особенностями выполняемой инструкции. Минимальный подцикл обработки инструкции на одном уровне равен шести полутактам; при этом минимальные подциклы одновременно на всех уровнях возможны только при реализации инструкции 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 выполнению инструкции на данном уровне может начинаться только тогда, когда триггер занятости 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 на нем обработки информации, необходимой для рассматриваемого уровня.

Установка триггеров занятости и готовности уровней в соответствующие состояния производится 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 АЛУ, сигнал отключения каналов и сигнал окончания записи в оперативную память. Так, например, при обработке инструкции 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 данному моменту времени первый операнд уже выбран из местной памяти и размещен в соответствующем регистре блока выборки данных.

Принципы совмещения циклов при обработке инструкции по трем уровням иллюстрируются диаграммами, приведенными на рис. 7.4. Диаграмма на рис. 7.4, *а* построена применительно к случаю последовательной обработки инструкций формата RR. На временном интервале от t_1

до t_2 производится выборка участка программы, так как восьмибайтный код, принимаемый из ОП, содержит четыре инструкции формата RR. Распаковка, т. е. выделение инструкций, начинается с момента времени t_2; инструкции на рис. 7.4, *а* 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.

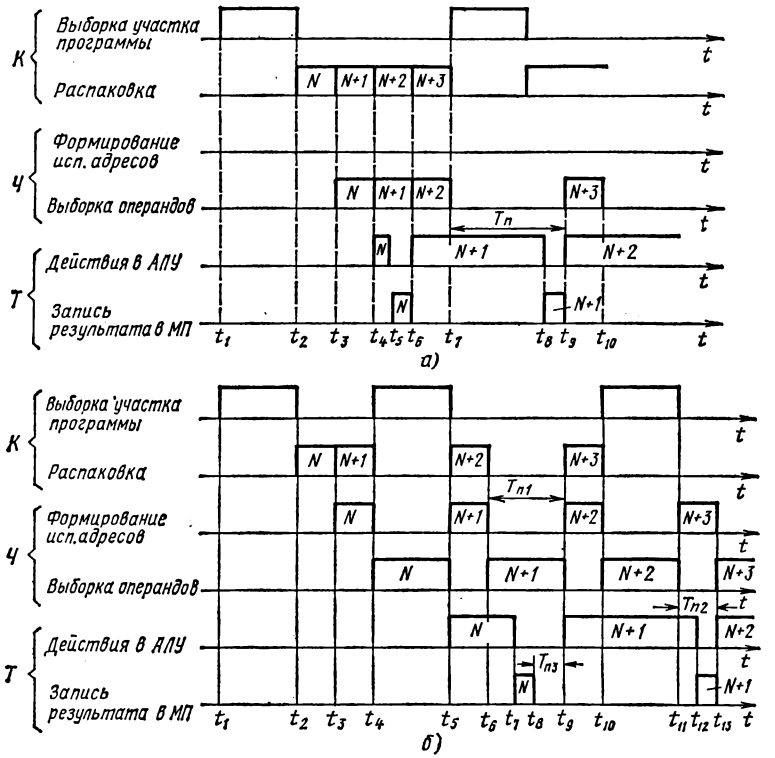


Рис. 7.4. Диаграммы совмещения циклов: *а* — при выполнении команд формата RR; *б* — при выполнении команд формата RX

На уровне Ч в соответствии с особенностями инструкций формата RR производится только выборка операндов из местной памяти. 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 выделение инструкции $N+2$, на уровне Ч — выборка операндов по инструкции $N+1$, на уровне Т — выполнение операции и запись результата в местную память по инструкции N. С момента времени t_7 приостанавливается работа на уровне Ч, так как до момента времени t_9 уровень Т занят выполнением действий по инструкции $N+1$. В момент времени t_9 на уровень Т передаются операнды по инструкции $N+2$, а на уровне Ч после простоя, равного $T_п$, начинается выборка операндов по инструкции $N+3$.

Диаграмма, приведенная на рис. 7.4, *б*, отвечает случаю последовательного выполнения инструкций формата RX, содержащихся по две в каждом восьмибайтном коде, принимаемом из оперативной памяти. В зависимости от длительности выполняемых в АЛУ операций возможны приостановки работ на уровнях Ч и Т; необходимость формирования исполнительных адресов на уровне Ч и выборки вторых операндов из оперативной памяти обычно приводит к приостановке работ на уровне К. Так, выделение инструкции $N + 3$ задерживается на время $T_{п1}$, так как уровень Ч занят действиями по инструкции $N + 1$. Время простоя $T_{п2}$ уровня Ч вызвано большой длительностью операции, выполняемой в АЛУ по инструкции $N + 1$, а время простоя $T_{п3}$ уровня Т вызвано тем, что действия на нем по инструкции N закончились раньше, чем действия на уровне Ч по инструкции $N + 1$.

7.2. Микропрограммное управление

Управляющие сигналы формируются в центральном устройстве управления в соответствии с принципами микропрограммного управления тогда, когда основные УС вырабатываются за счет последовательного считывания и расшифровки управляющих слов, располагающихся в ячейках отдельного постоянного ЗУ или постоянной памяти (ПП), которая рассматривается иногда как односторонняя память. Каждое такое слово называется *микрокомандой* и содержит информацию о тех действиях, тех микрооперациях, которые выполняются 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, установка некоторого триггера в единичное состояние, сброс регистра в исходное нулевое состояние, передача информации из одного регистра в другой и т. п.

Последовательность микрокоманд, обеспечивающих выполнение одной машинной команды, т. е. инструкции процессора, 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 с целью увеличения количества возможных наборов управляющих сигналов разделяется на ряд частей, или полей. Каждое поле преобразуется в управляющие сигналы автономно; собственно преобразование обеспечивается с помощью дешифраторов; дешифрирование отдельных полей может производиться с учетом информации, содержащейся в других частях микрокоманды.

Структура микрокоманды определяется возможностями используемой постоянной памяти, особенностями построения основных блоков и устройств машины, количеством выделяемых управляемых цепей и другими факторами, в том числе ее форматом. В любом случае одно или несколько полей обеспечивают формирование адреса следующей микрокоманды

с учетом возможных разветвлений в микропрограммах. Для машин ЕС-1020 и ЕС-1022 формат микрокоманд соответствует 64 двоичным разрядам, включая контрольные, для ЕС-1030 – 72 разрядам, а для ЕС-1033 – 132 разрядам.

Принципы построения и работы микропрограммного устройства управления рассматриваются ниже на примере машины ЕС-1022, в которой оно обеспечивает работу арифметическо-логического блока, каналов, оперативной и локальной памяти, выполнение пультовых и других процедур. Приводятся также особенности использования микрокоманд большого формата.

Структура машины ЕС-1022. Основные функциональные связи. Выделение полей в микрокомандах производится в соответствии с принятым делением машины на функциональные части, связями между ними, принципами выполнения действий по реализации инструкций процессора. Именно с этих позиций и рассматриваются общая структура ЕС-1022, а также функциональные связи между ее узлами и блоками, иллюстрируемые схемой, приведенной на рис. 7.5. В качестве основных функциональных частей выделены: центральное устройство управления ЦУУ, арифметическо-логический блок АЛБ, каналы, блок регистров БР, оперативная память ОП и локальная память ЛП со своими адресными и информационными регистрами.

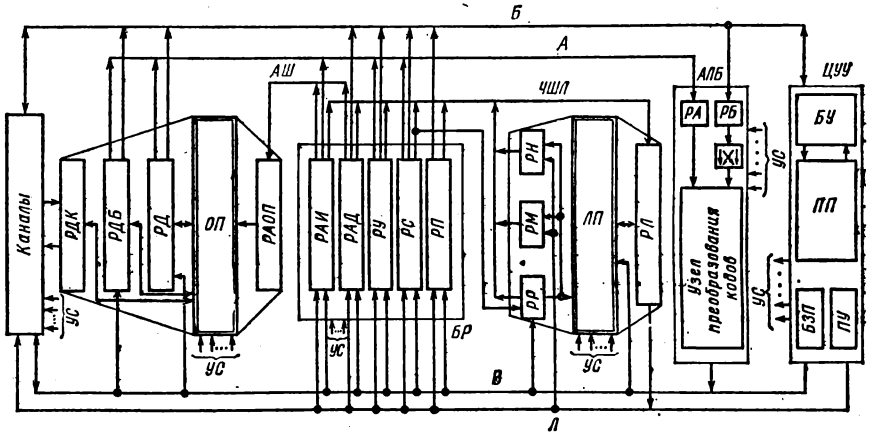


Рис. 7.5. Основные функциональные связи в машине с микропрограммным управлением. (ЕС-1022)

Центральное устройство управления содержит ряд составных частей, основные из которых показаны на рассматриваемой схеме; это блок управления БУ, постоянная память ПП микропрограмм, пульт управления ПУ и блок защиты памяти БЗП. Главной функцией ЦУУ является формирование управляющих сигналов УС в соответствии с характером выполняемых операций и процедур. Коды операций поступают в ЦУУ по общим информационным шинам Б. Посредством этих шин, а также общих информационных шин В и Л центральное устройство управления

имеет возможность обмениваться информацией с другими функциональными частями машины.

Арифметическо-логический блок может оперировать как с однобайтными, так и с двухбайтными операндами, что определяется соответствующими режимами работы АЛБ: «1 байт» и «2 байта». За время выполнения одной микрокоманды возможна реализация двух подтактов обработки операндов. В этом случае, если АЛБ работает в режиме «2 байта», при реализации первого подтакта выполняется заданная операция над младшими полусловами операндов, а при реализации второго подтакта — над старшими полусловами операндов. Входными для АЛБ являются регистры РА и РБ, связанные с информационными шинами А и Б соответственно. Тетрады из РБ в узлы преобразования кодов могут передаваться обычным порядком или со сменой мест, что на рис. 7.5 условно обозначено набором стрелок. Образующиеся на выходе АЛБ байты или полуслова результатов выполнения операций выдаются в информационные шины В.

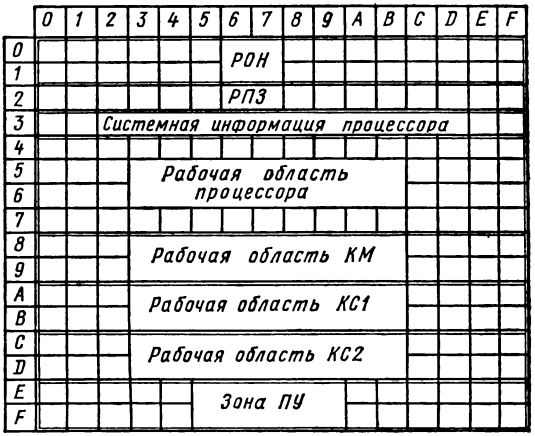
Оперативная память машины выполняет функции основной оперативной и мультиплексной памяти, так как определенная ее зона выделена для хранения информации, используемой мультиплексным каналом при организации работы с устройствами ввода и вывода данных. Общая емкость оперативной памяти может составлять 512К байт; тогда регистр адреса оперативной памяти РАОП имеет 19 двоичных разрядов, не считая контрольных; при обращении к мультиплексной памяти используются 11 разрядов РАОП. Ввод адреса в РАОП осуществляется по адресным шинам АШ. Основным информационным регистром, или регистром числа, является четырехбайтный регистр РД; используется также буферный информационный регистр РДБ; эти регистры связаны с информационными шинами А, Б и В. При обмене информацией с каналами используется четырехбайтный регистр РДК. Отметим, что при всех передачах данных каждый информационный байт сопровождается контрольным разрядом.

Локальная память представляет собой сверхоперативное запоминающее устройство общей емкостью 256 полуслов (512 байт). Часть ячеек выполняет функции регистров общего назначения РОН, служащих для размещения 16 слов с фиксированной запятой, и регистров, служащих для размещения четырех слов с плавающей запятой, или РПЗ. Выделяются также рабочие области процессора, каналов мультиплексного КМ, селекторных КС1 и КС2, а также ячейки для размещения системной информации процессора, как это показано на рис. 7.6. Цикл работы ЛП равен 275 нс при времени считывания информации, равном 125 нс.

Накопитель локальной памяти выполнен в виде триггерной матрицы, которую составляют интегральные микросхемы типа 155 РУ-1. Запись и считывание информации производятся полусловами — два байта с контрольными разрядами; доступ к ячейкам памяти — произвольный. При выполнении одной микрокоманды может быть организовано до двух обращений к локальной памяти; если осуществляется запись, то при первом обращении к ЛП записывается, как правило, младшее полуслово из заданного регистра, а при втором — старшее полуслово; такое же распределение полуслов обеспечивается и при организации

двух циклов считывания за время выполнения одной микрокоманды. Основным адресным регистром локальной памяти является однобайтный (девятиразрядный) регистр РР; кроме него в качестве адресных используются еще регистры РМ и РН (см. рис. 7.5), четыре младших разряда которых выполнены по схеме реверсивного счетчика, что позволяет увеличивать или уменьшать содержимое этих регистров. Информация

Рис. 7.6. Распределение поля локальной памяти



в адресные регистры может заноситься с общих информационных шин В и Л. Регистр РЛ выполняет функции информационного регистра локальной памяти. Коды в этот регистр вводятся по числовым шинам ЧШЛ; выдача кодов осуществляется на общие информационные шины Л. Отметим, что в отдельных случаях запись полуслов в накопитель ЛП может производиться непосредственно с общих информационных шин В.

Центральное устройство управления не содержит в своем составе узлов и блоков для преобразования управляющих слов типа счетчика команд, сумматора адреса и т.п. Преобразования информации осуществляются только в АЛБ. Для хранения и выдачи информации, необходимой в процессе функционирования машины, используются регистры процессора, составляющие вместе со своими входными и выходными схемами блок регистров. Сюда относятся адресные и другие регистры, ввод информации в которые осуществляется с общих информационных шин В и Л (см. рис. 7.5); выдача информации из регистров производится через общие информационные шины А и Б в другие блоки и устройства машины. Содержимое адресных регистров выдается в РАОП через адресные шины; кроме того, коды из любого регистра могут выдаваться через ЧШЛ в локальную память для записи.

Адресные регистры РАИ и РАД служат для хранения адресов команд (инструкций процессора) и операндов; в отдельных случаях они могут использоваться для хранения любой другой информации. Эти регистры являются трехбайтными, причем на каждый байт приходится один контрольный разряд. Основное назначение регистра РАИ состоит в хранении адреса команды, а регистра РАД — в хранении адресов

операндов. Занесение информации в адресные регистры производится по соответствующим полям микрокоманд, причем в течение одного такта они могут быть полностью заполнены информацией, поступающей с выхода АЛБ при работе последнего в режиме «2 байта».

Регистр общего назначения РУ предназначен для хранения и выдачи информации, участвующей в процессе выполнения операций. Он является четырехбайтным и по своим входам и выходам связан с входными регистрами и выходом арифметическо-логического блока, а также информационным регистром РЛ локальной памяти. Занесение информации в РУ может производиться как по одному, так и по двум байтам. Если запись осуществляется в каждом из двух возможных подтактов, например при работе АЛБ в режиме «2 байта», то первое полуслово фиксируется в двух младших байтах РУ, а второе полуслово — в двух старших.

Четырехбайтный регистр состояний РС служит для хранения информации, отражающей состояния различных блоков и узлов машины и состояние выполняемой программы. Эта информация на определенных этапах работы машины используется для принятия решений о последующих действиях. Первые три байта регистра, т. е. РС0, РС1 и РС2, представляют собой объединения отдельных триггеров, служащих для фиксации состояний блоков и узлов машины, а также выполняемой программы, и двухразрядных счетчиков байтов; четвертый байт регистра, т. е. РС3, служит для размещения части выполняемой инструкции процессора. Содержимое РС2 и РС3 может быть выдано на любой из двух входов АЛБ. Содержимое всего регистра РС за исключением отдельных разрядов РС1, может быть выдано для записи в локальную память; кроме того, содержимое РС3 используется для адресации ячеек ЛП, отвечающих регистрам общего назначения и РП3.

Нулевой разряд РС0 представляет собой триггер переполнения ТПЕР, который устанавливается в единичное состояние при формировании сигнала переполнения в АЛБ. Первый разряд РС0, т. е. РС0 [1], представляет собой триггер знака ТЗН, который хранит информацию о состоянии старшего разряда результата выполнения действия в АЛБ. Содержимое РС0 [2—3] отражает код условия, который формируется в соответствии с результатами выполнения действия в АЛБ. Разряды РС0 [4—7] используются как универсальные индикаторы, обеспечивающие организацию разветвлений в микропрограммах. В разрядах РС1 [0—1] хранится код длины выполняемой машиной команды, а содержимое РС1 [2] определяет место нахождения очередной команды — в регистре РДБ или в оперативной памяти. Разряд РС1 [3] отражает работу процессора или одного из каналов, а разряды РС1 [4—7] — результаты функционирования АЛБ.

Разряды РС2 распределены по трем двухразрядным счетчикам байт и указателям направления модификации содержимого этих счетчиков. Состояния этих счетчиков, выделяемые с помощью простых дешифраторов, обеспечивают передачу соответствующих байт информации между адресуемыми регистрами, входами и выходами арифметическо-логического блока. Разряды РС3 служат для хранения второго байта выполняемой команды; их содержимое может модифицироваться и выдаваться в ЛП.

Регистр прерываний РП служит для фиксации запросов на прерывания,

а также внешних и внутренних условий, определяющих действия, связанные с обеспечением прерываний выполняемых программ. В нем можно выделить четыре однобайтные части РП0, РП1, РП2 и РП3, причем РП3 имеет десять информационных разрядов. Разряды РП0 служат для хранения масок системы и контроля. При этом 0, 1, 2 и 7-й разряды хранят соответственно маски мультиплексного, первого селекторного, второго селекторного каналов и маску внешних прерываний; эти одноразрядные маски составляют маску системы. Единичное состояние триггера соответствующего разряда указывает на то, что прерывание разрешено, а нулевое — запрещено (замаскировано). Пятый разряд РП0 хранит маску контроля; единица в данном разряде указывает на то, что прерывание по сбою разрешено, а нуль — на то, что это состояние замаскировано.

Разряды РП1 служат для хранения ключа защиты памяти и ключа реализуемой программы; каждый из этих ключей представляет собой четырехразрядный двоичный код. В РП2 фиксируются запросы на внешние прерывания. Сигналы от таймера и кнопки прерывания на пульте управления поступают соответственно в РП2 [0] и РП2 [1]. Триггеры разрядов со 2-го по 7-й хранят запросы на прерывания, поступающие от внешних объектов по шести отдельным линиям. Наличие запроса на прерывание отражается единичным состоянием соответствующего триггера. Установка запроса на прерывание от таймера осуществляется микропрограммно путем пересылки в РП2 [0] единицы из АЛБ. Запросы на остальные внешние прерывания устанавливаются аппаратно, т. е. путем передачи сигналов непосредственно на входы соответствующих триггеров. В РП3 фиксируются сигналы сбоев отдельных узлов машины; разряд РП3 [0] служит для фиксации сигнала сбоя регистра блока защиты памяти, РП3 [1—2] — сбоев входных регистров АЛБ, РП3 [3—4] — сбоев регистров адреса и информации ОП, РП3 [5] — сбоя регистра микрокоманд, РП3 [6] — сбоя регистра адреса ОП, РП3 [7] — сбоя АЛБ, РП3 [8—9] — сбоев регистров адреса и данных ЛП.

Состав и работа блока управления. Блок управления является одной из основных частей микропрограммного ЦУУ, в состав которого еще обязательно 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 машинной команды, а начало микропрограмм выполнения тех или иных процедур — с соответствующими сигналами и условиями, формируемыми в процессе работы машины. Все микропрограммы хранятся в постоянной памяти, с которой в процессе функционирования непрерывно взаимодействует блок управления.

Структура блока показана на рис. 7.7, отражающем общую структуру микропрограммного устройства управления. На этом рисунке в составе постоянной памяти выделены только накопитель, регистр адреса РАПП и регистр числа, или дополнительный регистр микрокоманд РМКД.

Собственно блок управления включает в свой состав регистр микрокоманд РМК, дешифраторы полей микрокоманд ДШ, узел формирования адресов микрокоманд и их занесения в регистр адреса постоянной памяти

УЗАМК, узел управления формированием адреса микрокоманды, или узел управления регистром постоянной памяти УРАПП, регистр микропрограммной приостановки РМП.

Регистр микрокоманд служит для приема из ПП и хранения в течение одного такта очередной микрокоманды. Коды из него выдаются на дешифраторы полей; отдельные части микрокоманд, участвующие в формировании адресов последующих микрокоманд, выдаются в УЗАМК. Дешифраторы полей служат для непосредственного преобразования соответствующих частей микрокоманд в управляющие сигналы. В УЗАМК обеспечивается получение адреса очередной микрокоманды; этот узел по приему информации связан не только с РМК, но и с пультом управления, арифметическо-логическим блоком и регистром микропрограммной приостановки.

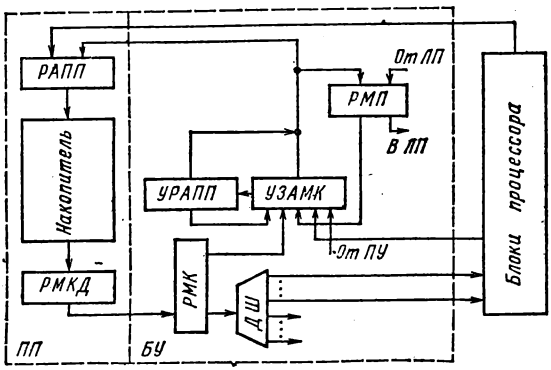


Рис. 7.7. Структура микропрограммного устройства управления

Реализуемые микропрограммы могут прерываться при возникновении соответствующих ситуаций. Для сохранения адреса микрокоманды, на которой производится прерывание, используется регистр микропрограммной приостановки. В случае необходимости содержимое РМП передается в локальную память; обратная пересылка информации обеспечивает восстановление содержимого РМП, т. е. возврат к прерванной микропрограмме в месте приостановки ее выполнения.

Работа блока управления протекает в следующей последовательности. Сформированный в УЗАМК адрес очередной микрокоманды передается в РАПП. Выбранная по этому адресу микрокоманда заносится сначала в РМКД, а затем в РМК. Поля микрокоманды, находящейся в РМК, дешифрируются с помощью ДШ блока управления; образуемый необходимый набор управляющих сигналов вызывает соответствующие действия в течение одного такта работы машины.

Параллельно и практически одновременно с дешифрированием полей текущей микрокоманды происходит формирование адреса следующей микрокоманды. В формировании адреса при различных условиях участвуют определенные поля текущей микрокоманды, содержимое РМП, код адреса микрокоманды, набранный на пульте управления, старшая тетрада информации из узла перекосов арифметическо-логического блока. Все основные действия определяются кодами полей выполняемых микрокоманд.

Структура микрокоманды. Микрооперации. Микрокоманды, используемые в машине ЕС-1022, представляют собой 64-разрядные слова, разделяемые на ряд полей. Каждое поле обычно определяет до 2^m микроопераций, если m — разрядность поля; за один машинный такт выполняется, как правило, только одна микрооперация для данного поля. Поля, по которым формируются управляющие сигналы, имеют не больше четырех разрядов, так что при сугубо автономном дешифрировании одно такое поле определяет не более 16 микроопераций. В случае использования содержимого других полей по одному четырехразрядному полю возможно формирование до 64 различных управляющих сигналов.

По своему функциональному назначению поля микрокоманд объединяются в группы, связанные с управлением работой арифметическо-логического блока, управлением локальной памятью, заданием режимов работы блоков и устройств машины, занесением информации в регистры и отдельные триггеры, организацией адреса очередной микрокоманды. Выделяются еще так называемое общее поле и контрольные разряды.

Выделенные группы и отдельные поля сведены в таблицу, представленную на рис. 7.8. Для всех возможных кодов некоторых полей в таблице даны сокращенные обозначения, отражающие суть соответсвующих микроопераций. Отметим, что при пользовании таблицей необходимо, если $m < 4$, в кодах полей, представленных в первом столбце, отбрасывать $(4-m)$ старших разрядов и использовать соответствующее число строк. Так, если $m = 3$, то в кодах полей читаются три младших разряда при использовании восьми строк таблицы, и т.д. Поля, занимающие разряды с 36-го по 63-й, подробно характеризуются при рассмотрении соответствующих действий.

Группу полей, обеспечивающих управление работой арифметическо-логического блока, составляют поля А, Б, В, Ф (ФУНКЦИЯ) и Д (ДЕФОРМАЦИЯ). Трехразрядное поле А определяет источник информации, вводимой во входной регистр РА арифметическо-логического блока. В качестве источников информации используются, как правило, регистры процессора, которые на рис. 7.8 отмечены своими сокращенными обозначениями. По коду 010 в РА вводится содержимое регистра прямого управления РПУ, используемого при получении информации из другого процессора по шинам прямого управления. По кодам 110 и 111 поля А в РА вводятся различные константы, т. е. К1 и К2. При коде 000 в РА заносятся нули. Отметим, что микрооперации, выполняемые по кодам поля А, могут иметь до четырех модификаций в соответствии со значениями одноразрядных полей БА и ЦБА выполняемой микрокоманды; это замечание относится также к микрооперациям, отвечающим кодам полей Б и В.

Четырехразрядное поле Б микрокоманды определяет источник информации, вводимой во входной регистр РБ арифметическо-логического блока. В качестве источников информации используются регистры процессора, а также каналы: мультиплексный КМ и селекторные КС1 и КС2. По коду 1000 в РБ вводится содержимое регистра ключей канала РКК, а по коду 1001 — байт состояния одного из селекторных каналов. При коде 0000 в РБ заносятся нули.

Четырехразрядное поле В, иногда относимое также к группе занесения

Код	КРТ	Поля																														
		АЛБ					ЛП			БА	ЦБА	ЦЛП	КБ	Общее поле	УСТ	УСЛ1	УСЛО	КСТ	КМЛ/АЛСТ	АПМЛ	КРЗ	КРЗ										
		А	Б	В	Ф	Д	ВЫХОД	ВХОД	АДРЕС																							
		0 1 3	4 7	8 11	12 13	14 15	16 19	20 23	24 27	28	29	30	31	32 35	36 39	40 43	44 47	48 51	52 55	56 61	62	63										
0000		РА:=0	РБ:=0	—	да КС нет	Прямо	—	РЛ:=0	РМ	2	2	2		—																		
0001		РАИ	РАД	РАИ	0-КФ 1-АлБ	Накрест	РАИ	РАИ	РН	1	1	1		РАИ→РАОП РД и РДБ																		
0010		РПУ	РУ	РД	2-А-Б 3-А·Б	Перекас	РАД	РАД	РМ(К)					РАД→РАОП РДК																		
0011		РУ	РС	РУ	0-АVБ 1-АЛБ	Дефор-мация	РУ	РУ	РН(К)					и т. д.																		
0100		РС	РД (РДБ)	РС	2-Б-А 3-А·Б		РС	РС	РСЗ																							
0101		РД (РДБ)	РП	РД (РДБ)			РД (РДБ)	РД (РДБ)	АЛБ																							
0110		К1	К2	РП			РМ, РН, РМП	РМП, РМ, РН	К																							
0111		К2	КМ (КС)	РКК			Зак. в ЛП	ЛП	РМ																							
1000			РКК	РБК			—	РВК	РМ					КФ																		
1001			БС	РФК			ТПР:=1	РДГ	РН																							
1010			КС1 КС2	РНК			РУ	РУ	РМ																							
1011			РС	РСК			РД (РДБ)	РД (РДБ)	РН																							
1100			РД (РДБ)	РДП			РМ, РН	РМ, РН	РМ																							
1101			РД (РДБ)	РАИ, ЛП			РП	РП	РН																							
1110			РУ	ЛП			РС	Уст.	РМ																							
1111			РД (РДБ)	РС			РИ	БАК	РН																							

Рис. 7.8. Структура микрокоманды

в регистры и триггеры, определяет приемник информации с выхода В арифметическо-логического блока, т. е. регистр, в который передается код из АЛБ. Кроме регистров процессора, адресуются регистры каналов: регистр буфера канала РБК, регистр флажков канала РФК, регистр номера канала РНК, регистр-счетчик канала РСК, а также регистр, предназначенный для хранения таймера, т. е. РДП. По коду 1110 информация с выхода АЛБ передается в локальную память, минуя ее информационный регистр. При коде 0000 информация с выхода АЛБ никуда не передается.

Двухразрядное поле Ф определяет базовые, или основные, микрооперации, выполняемые арифметическо-логическим блоком; остальные микрооперации задаются при использовании так называемого общего поля микрокоманды. В зависимости от того, работает селекторный канал или нет, задаются модификации базовых микроопераций, выполняемых над операндами А и Б, т. е. содержимым РА и РБ. При нулевом коде поля Ф, когда не работает селекторный канал, выполняется микрооперация по так называемой косвенной функции (считается, что в остальных случаях выполняется прямая функция); эта микрооперация заранее задается путем занесения необходимого кода в регистр косвенной функции РКФ арифметическо-логического блока. Для занесения кода в РКФ в поле УСТ или общем поле одной из предшествующих микрокоманд должен быть код, определяющий микрооперацию КФ, т. е. занесение косвенной функции. По этой микрооперации в РКФ записывается содержимое поля КМЛ микрокоманды с микрооперацией КФ. Таким образом, в рассматриваемом случае выполняется микрооперация, заданная, как правило, заранее, т. е. КОСВЕННАЯ ФУНКЦИЯ.

Двухразрядное поле Д определяет способы передачи информации со входного регистра РБ в узлы обработки данных, или узлы преобразования кодов арифметическо-логического блока; содержимое РБ при всех передачах информации не изменяется.

По коду 00 содержимое РБ передается без каких-либо изменений; по коду 01 первый байт содержимого РБ передается со взаимной сменой мест старшей и младшей тетрад; по коду 10 производится сдвиг тетрад первого байта содержимого РБ при использовании узла перекосов АЛБ. Если в поле Д задан код 11, то способ передачи информации определяется полем КСТ микрокоманды; при этом возможно задание способа передачи информации в узлы преобразования кодов и со входного регистра РА арифметическо-логического блока.

Группу полей, обеспечивающую управление локальной памятью, составляют три четырехразрядных поля, именуемых как ВЫХОД ЛП, ВХОД ЛП, АДРЕС ЛП.

Микрокоманды, определяемые кодами этих полей, могут иметь до двух модификаций, связанных с кодами одноразрядного поля ЦЛП. Поле ВХОД ЛП определяет источник информации для регистра числа локальной памяти, т. е. РЛ; эта информация записывается в соответствующие ячейки ЛП. Основными источниками информации являются регистры процессора. Кроме них используются регистр диагностики канала РДГ и регистр блока общего канала РВК; при коде 1110 осуществляется занесение единиц во все информационные разряды РЛ, а при коде 1111 — занесение байт набора адреса команды. По коду 0111 в РЛ записывается содержимое определенных ячеек ЛП; по коду 0000 РЛ устанавливается в исходное состояние (нулевое).

Поле ВЫХОД ЛП определяет приемник информации, считываемой из локальной памяти. В качестве основных приемников информации используются регистры процессора. При коде 1111 информация выдается в регистр индикации РИ пульта управления. По коду 0111 осуществляется запись считанной информации в определенные ячейки ЛП; по коду 1001 устанавливается в единичное состояние ТПР — триггер признака регистра, служащего для размещения слова с плавающей запятой, или РПЗ. При кодах 0000 и 1000 информация из ЛП никуда не выдается. Поле АДРЕС ЛП служит для указания регистра, в котором содержится адрес ячейки накопителя ЛП. При коде 0101 адрес поступает из АЛБ, а при коде 0110 задается константой К.

Группа одноразрядных полей используется в качестве указателей режимов работы функциональных частей процессора. Поле БА задает либо режим «2 байта», либо режим «1 байт» работы арифметическо-логического блока. Поля ЦБА и ЦЛП задают число циклов, два или один, работы арифметическо-логического блока и локальной памяти соответственно за время реализации микрокоманды. Поле КБ указывает, какой из информационных регистров памяти, РД или РДБ, используется при выполнении текущей микрокоманды.

Четырехразрядное общее поле микрокоманды, в частности, задает микрооперации управления оперативной памятью. Так, при коде 0001 источником адреса для ОП служит РАИ, а при коде 0010 — регистр адреса данных РАД. Ряд других кодов задает различные действия

по записи и считыванию информации, участвующей в выполнении заданной операции. По коду 0111 в регистр косвенной функции заносится четырехразрядный код поля КМЛ; это обеспечивает задание 16 различных микроопераций, реализуемых в арифметическо-логическом блоке.

Поле УСТ совместно с полем КСТ или без него определяет микрооперации установки различных служебных триггеров в состояния 0 и 1, а также некоторые специальные микрооперации. Это поле дешифрируется в зависимости от занятости поля КСТ формированием управляющих сигналов совместно с другими полями. Если поле КСТ используется совместно с полем УСТ, то его содержимое может заноситься в разряды РС или дешифрироваться с целью формирования сигналов на установку отдельных триггеров или на выполнение специальных действий. Если поле КСТ занято по микрооперациям других полей микрокоманды, то поле УСТ по соответствующей микрооперации дешифрируется аналогично полю КСТ.

Четырехразрядные поля УСЛ1 и УСЛ0 определяют состояния, в которые устанавливаются триггеры первого и нулевого разрядов соответственно регистра адреса постоянной памяти, т.е. РАПП [1] и РАПП [0]; отметим, что в РАПП, как и в некоторых других адресных регистрах, нумерация разрядов ведется справа налево. При коде 0000 в этих полях осуществляется безусловная установка РАПП [1] и РАПП [0] в нулевое состояние, при коде 0001 — в единичное состояние. При остальных кодах занесение в РАПП производится в зависимости от состояний триггеров (признаков), анализ которых производится при выполнении микрокоманд. Выбор триггеров (признаков) для анализа состояний осуществляется как по кодам рассматриваемых полей, так и по состоянию триггера проверки условий процессора и каналов ТУПК; этот триггер задает проверку триггеров либо процессора, либо каналов.

Поле КСТ используется, как правило, совместно с другими полями микрокоманды. Оно представляет собой старшую часть (старшую тетраду) 8-разрядной константы, передаваемой на вход РА. Это поле используется также для расширения возможностей полей Д и УСТ. Поле КМЛ/АПСТ используется либо как младшая часть 8-разрядной константы, подаваемой на входы РА, РБ, РС2, либо в качестве старшей части адреса перехода (адреса следующей микрокоманды); в последнем случае содержимое поля подается в РАПП [11—8]. Поле АПМЛ используется для формирования младшей части адреса постоянной памяти.

Контрольные разряды распределяются следующим образом. Нулевой разряд микрокоманды, т.е. КР1, представляет собой контрольный разряд для всей микрокоманды (для РМК), КР2 — контрольный разряд константы, КР3 — контрольный разряд РАПП.

Организация адреса микрокоманды. По способу организации адрес микрокоманды может быть нефиксированным, или естественным, и фиксированным. Естественные адреса — это те, которые образуются с помощью информации, содержащейся в полях УСЛ1, УСЛ0, КМЛ/АПСТ, АПМЛ выполняемых микрокоманд, при использовании в некоторых случаях содержимого РМП и кодов, поступающих из узла перекосов АЛБ. Различают два основных принципа формирования естественных

адресов, именуемых АПД – адрес перехода длинный и АПК – адрес перехода короткий; внутри каждого из них может быть реализован способ собственно АПД или АПК, а также способ ФПБ – функциональный переход по содержимому узла перекосов ПК1 [0–3] и способ ИРМ – переход по содержимому регистра мультиплексной приостановки РМП. Принципы формирования адресов микрокоманд отражены на диаграммах рис. 7.9. Обозначенный на этом рисунке сигнал СП обеспечивает установку в необходимое состояние РАПП [12]; формирование СП производится по кодам общего поля реализуемой микрокоманды. Отметим, что указанные принципы формирования адресов микрокоманд используются и при реализации команд условных и безусловных переходов.

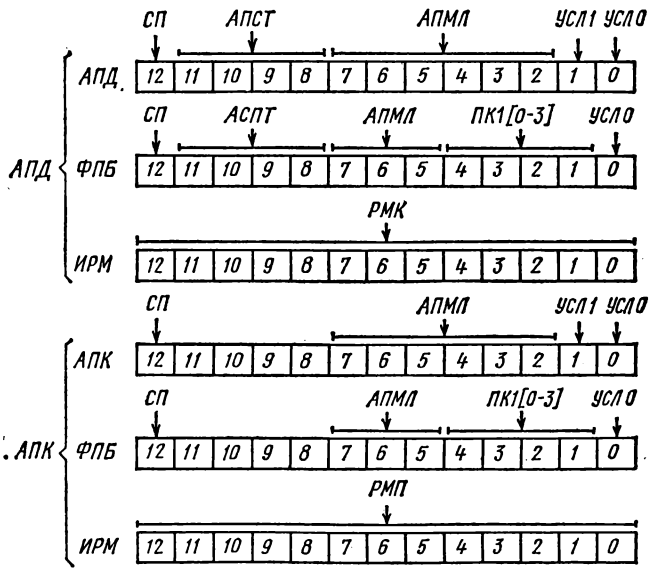


Рис. 7.9. Способы формирования адреса микрокоманды

Адрес перехода длинный формируется тогда, когда поле КМЛ/АПСТ не используется микрооперациями других полей. В этом случае оно рассматривается как старшая часть адреса перехода, т.е. как АПСТ; содержимое РМК [52–55] и РМК [56–61] передается в разряды РАПП [11–8] и РАПП [7–2] соответственно, как это показано на верхней диаграмме рис. 7.9. Поля УСЛ1 и УСЛ0 управляют установкой разрядов РАПП [1] и РАПП [0], что дает возможность организации ветвлений микропрограммы по четырем направлениям в предельном случае.

Функциональный переход при АПД используется тогда, когда возможно ветвление более чем по четырем направлениям. В этом случае конкретное значение адреса формируется при помощи содержимого ПК1 [0–3], которое подается в разряды РАПП [4–1]; это обеспечивает организацию ветвлений в пределе по 32 направлениям с учетом двух возможных значений РАПП [0]. Формирование значений разрядов РАПП [12–5] и РАПП [0] при функциональном переходе производится так же, как и при

собственно АПД. Переход по содержимому РМП обеспечивает возврат к прерванной микропрограмме; в этом случае РАПП полностью загружается содержимым регистра микропрограммной приостановки.

Адрес перехода короткий формируется тогда, когда поле КМЛ/АПСТ используется микрооперациями других полей. В этом случае содержимое разрядов РАПП [11—8] не изменяется; остальные разряды формируются аналогично случаю АПД. Функциональный переход при АПК аналогичен ФПБ при АПД; различие состоит только в том, что содержимое разрядов РАПП [11—8] не изменяется. Переход по содержимому РМП, т.е. ИРМ, при АПК полностью аналогичен такому же переходу при АПД.

Фиксированные адреса — это адреса, формирование которых производится без участия полей микрокоманд. Эти адреса заносятся в РАПП и дают начало микропрограммам первоначальной загрузки и гашения системы, диагностики, обслуживания селекторного и мультиплексного каналов, обработки прерываний по контролю и сбою машины и различных программных прерываний.

Особенности формирования и использования микрокоманд большого формата. Микрокоманды большого формата используются тогда, когда возможно совмещение во времени многих действий на уровне микроопераций. Такое совмещение обеспечивается, например, при построении процессора и каналов по принципам магистральных структур, как это сделано в машине ЕС-1033. Увеличение формата микрокоманд позволяет уменьшить общую длину микропрограмм, более рационально использовать возможности отдельных микрокоманд. Это приводит к уменьшению общей информационной емкости постоянной памяти, предназначенной для хранения микропрограмм. Так, если в машине ЕС-1022 общая информационная емкость постоянной памяти составляет 64 × 8192 бит, то в машине ЕС-1033 — 132 × 2048 бит; разрядность адреса при этом снижается с 13 до 11 двоичных разрядов.

Формат микрокоманд, принятых для ЕС-1033, составляют 132 двоичных разряда. Для хранения микрокоманд используются четыре унифицированных блока постоянной памяти с общей системой выборки информации; емкость каждого блока составляет 2048 слов по 33 двоичных разряда. Таким образом, при каждом обращении к постоянной памяти считываются четыре 33-разрядных слова, составляющих 132-разрядную микрокоманду, фиксируемую в регистре микрокоманд. Общая структура такой микрокоманды показана на рис. 7.10, где выделены ее четыре 33-разрядные части, именуемые нулевым, первым, вторым и третьим словами МК; последний разряд каждого слова является контрольным.

Нулевое слово МК содержит шесть пятиразрядных полей, первые разряды которых являются управляющими. При единичном значении управляющего разряда У остальные четыре разряда поля задают адрес операционного регистра, используемого при выполнении заданных операций. Обозначение М1ЧТ указывает на то, что содержимое адресуемого регистра выдается в магистраль М1, М1ЗП — в адресуемый регистр информация записывается с магистрали М1 и т.п. При нулевом значении У остальные четыре разряда поля используются для формирования

управляющих сигналов обычным методом дешифрирования кодов. Разряды 30 и 31 нулевого слова не используются.

Первое слово МК в разрядах [0—10] содержит адрес следующей микрокоманды; это, практически, базовый адрес, который при определенных условиях может модифицироваться. Четырехразрядное поле УЗАП обеспечивает управление модификацией базового адреса в соответствии с результатами анализа различных признаков и условий. Одноразрядное поле ЗП служит для позиционного кодирования запроса в оперативную

Нулевое слово МК

0	1-4	5	6-9	10	11-14	15	16-19	20	21-24	25	26-29	30-31	32
У	М1ЧТ	У	М1ЗП	У	М2ЧТ	У	М2ЗП	У	М3ЗП	У	М3ЧТ		КР

Первое слово МК

0-10				11-14	15	16-18	19-24	25	26-29	30	31	32
Базовый адрес				УЗАП	ЗП	А	Б	М	S	ОЧ	ПА	КР

Второе слово МК

0-3	4-6	7-9	10-12	13-15	16-17	18-19	20-21	22-23	24-25	26	27	28	29-31	32
В	Г	Д	Е	Ж	Л	Н	П	Р	С	УК	УТ	РД	–	КР

Третье слово МК

0	1-3	4-7	8-11	12-19	20-27	28-31	32
	Х	Ф					КР
	Адрес возврата			РМС	РКС	РРД	
0-й байт			1-й байт		2-й байт	3-й байт	

Рис. 7.10. Общая структура микрокоманды большого формата

память по выборке или записи одного байта. Поля А и Б обеспечивают формирование различных управляющих сигналов. Поля М и S определяют режим работы арифметическо-логического устройства и выполняемые им действия; при М = 0 выполняются арифметические операции, а при М = 1 — логические операции. Разряды [30] и [31] первого слова МК служат для позиционного кодирования отдельных управляющих сигналов.

Второе слово микрокоманды содержит четырехразрядное поле В, четыре трехразрядных поля Г, Д, Е, Ж и пять двухразрядных полей Л, Н, П, Р, С. Эти поля дешифрируются по общим правилам, обеспечивая формирование необходимых управляющих сигналов. Одноразрядные поля УК, УТ и РД являются указателями (признаками), определяющими способы использования трехразрядного поля Т второго слова МК и всего третьего слова микрокоманды. Значение УК = 1 свидетельствует о том, что в третьем слове МК содержится константа, используемая как адрес перехода, как код для сравнения с некоторым формируемым в процессоре кодом, для установки состояния определенного счетчика и т. п. При УК = 1 разряды [1—11] третьего слова микрокоманды всегда воспринимаются как адрес возврата. Если же УК = 0, то разряды [1—3] этого слова воспринимаются как трехразрядное поле Х, а разряды [4—7] того же слова — как четырехразрядное поле Ф; в этом случае

поля X и Ф используются обычным порядком для формирования управляющих сигналов.

По значению кода, содержащегося в однорязрядном поле УТ, устанавливается принцип использования всего третьего слова микрокоманды и отдельных разрядов второго слова или отдельных частей второго и третьего слов. Если УТ = 1, то третье слово микрокоманды используется как полноразрядная константа; в этом случае 28-й разряд второго слова воспринимается как дополнение до четности нулевого байта третьего слова МК, 29-й разряд второго слова — как дополнение до четности первого байта третьего слова МК, 30-й разряд второго слова — как дополнение до четности второго байта третьего слова МК и 31-й разряд второго слова — как дополнение до четности третьего, т. е. младшего, байта третьего слова МК; 32-й разряд второго слова используется для дополнения до нечетности основных разрядов [0—31] третьего слова микрокоманды. Если же УТ = 0, то единица одноразрядного поля РД воспринимается как сигнал занесения константы в регистр РД процессора из поля РРД третьего слова МК, а разряды [29—31] второго слова МК — как трехразрядное поле Т, дешифрируемое обычным порядком.

Поля третьего слова микрокоманды, если оно не используется в качестве полноразрядной константы, распределяются так. Разряды поля РМС, связанного с регистрами масок, служат для маскирования кодов, выдаваемых на магистрали. Разряды поля РКС используются при проведении анализов в качестве базовых кодов для сравнения. Из поля РКС могут выделяться также коды для установки в соответствующие состояния отдельных счетчиков процессора. В поле РРД располагается константа, которая записывается в регистр РД в обратном коде.

Формирование адреса очередной 132-разрядной микрокоманды осуществляется в отдельном блоке процессора. При этом соблюдается установленная приоритетность для возможных источников кода адреса. Высший приоритет имеет код, поступающий от пульта управления; низшим приоритетом обладает базовый адрес, содержащийся в коде исполняемой микрокоманды; возможная модификация младших четырех разрядов базового адреса имеет более низкий приоритет, чем код адреса возврата по прерванной микропрограмме и код начального адреса, формируемый в соответствии с заданной операцией по выполняемой инструкции процессора.

7.3. Схемное управление

Управление считается схемным тогда, когда формирование управляющих сигналов осуществляется специальными схемами и 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 и ЕС-1052.

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; блок адреса

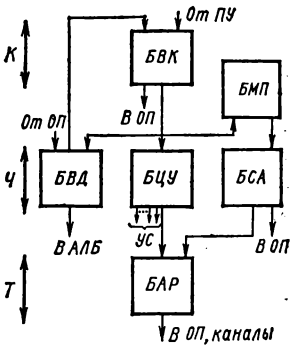


Рис. 7.11. Центральное ядро «схемного» устройства управления

результата БАР отвечает уровню Т. 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 необходимых действий при реализации инструкций процессора, причем с целью унификации используется понятие арифметическо-логического блока (АЛБ).

Блок выборки команд. Выходные коды машин ЕС-1050 (ЕС-1052) 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 основной 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 показаны их связи. К таким узлам относятся: счетчик тактов начального цикла СЧНЦ, счетчик тактов СЧТК уровня К, счетчик адресов команд, или счетчик адресов участков программ СЧАП, регистр адресов участков 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 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, приводящих в необходимые состояния 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.

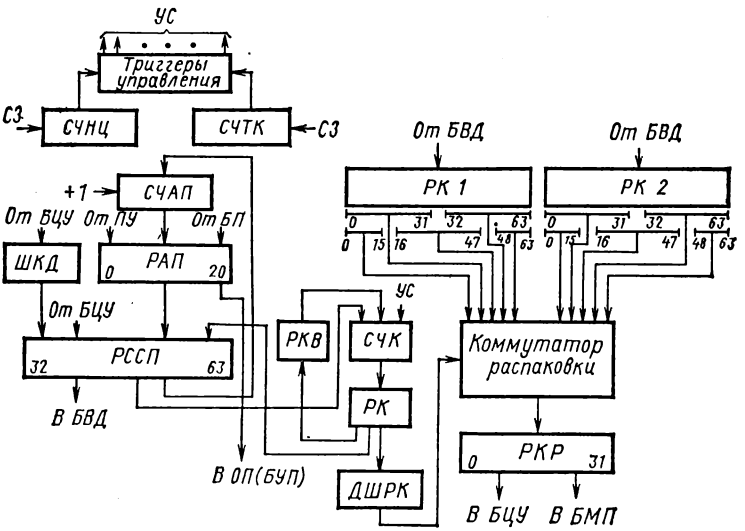


Рис. 7.12. Блок выборки команд

Двойное слово, выбираемое из оперативной памяти по содержимому РАП, через блок выборки данных вводится в один из регистров РК1 или РК2, что определяется состоянием 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 СЧТК, 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 никакого кода, 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 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 РП30, РП32, РП34, РП36. 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 РОН0—РОН15.

Ввод информации в регистры производится через группы входных вентилей, автономных 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.

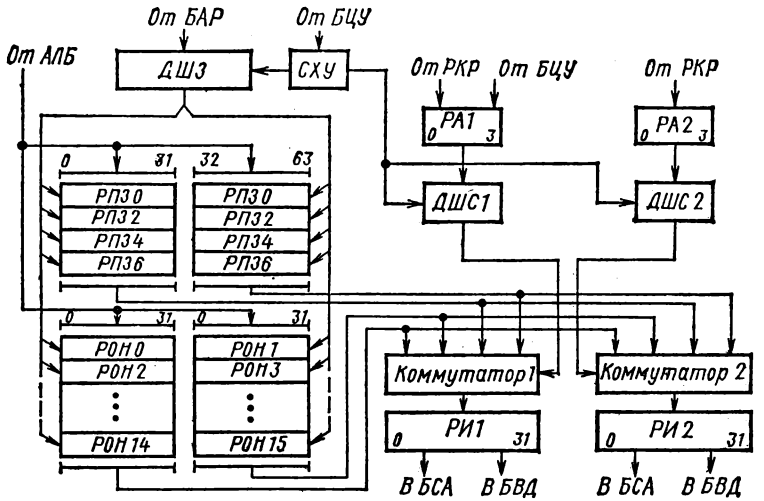


Рис. 7.13. Структура блока местной памяти

Вывод информации из регистрового накопителя производится через группы выходных вентилей, 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

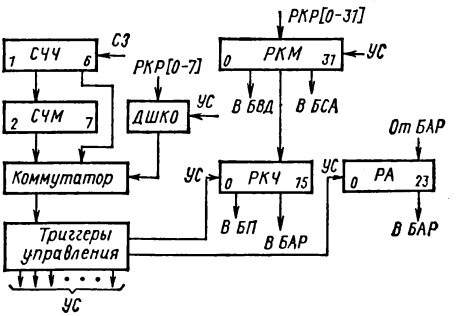


Рис. 7.14. Общая схема блока центрального управления

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 незанятости 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 начинает 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 связаны 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.

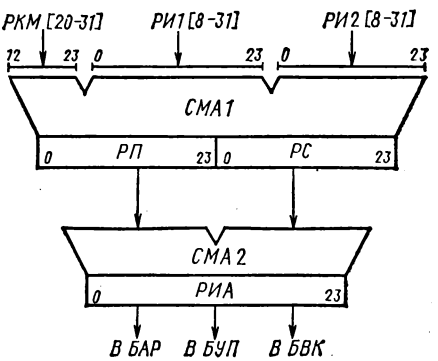


Рис. 7.15. Схема формирования исполнительного адреса

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 исполнитель-

ного адреса РИА, группы вентилей. 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 контрольных функций; причем достигается глубина контроля, соответствующая контролю 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. Отметим, что РБЦ, как и

все остальные регистры процессора, имеет контрольные разряды — по одному на каждый байт. Содержимое контрольных разрядов дополняет количество единиц информационных байт до нечетного числа и используется при контроле пересылок информации.

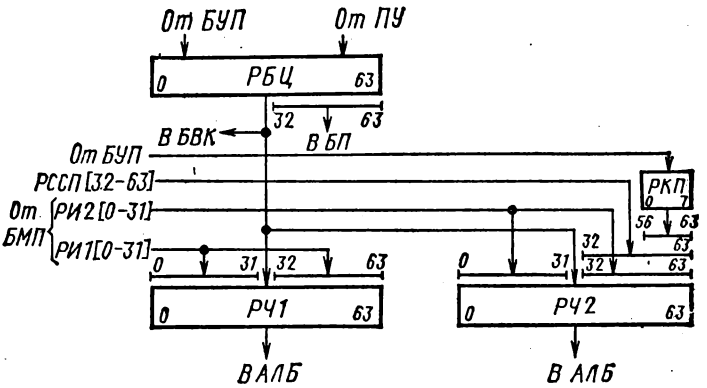


Рис. 7.16. Структура блока выборки данных

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, формиро-

вания 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

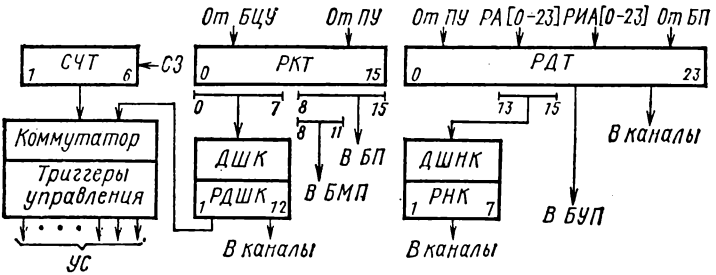


Рис. 7.17. Общая схема блока адреса результата

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 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 операции трансляции. Содержимое РКТ[8—15] может выдаваться в блок прерываний.

Регистр адреса результата уровня Т 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 инструкции 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, имеющий семь выходов. С каждым выходом дешифратора связан триггер регистра номера канала, 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 Т производится как бы в два приема. Сначала по первым трем тактовым импульсам счетчика тактов 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 Т, включает в свой состав схемы, формирующие запросы на прерывания по уровню Т. 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 прерываний и приоритетов

Во всех ЭВМ ЕС система прерываний и приоритетов включает в свой состав как аппаратные, так и программные средства. Основу программных средств составляют супервизорные программы, обеспечивающие анализ причин, вызвавших прерывание того или иного класса. Эти программы, называемые обработчиками прерываний, относятся к используемой операционной системе и могут считаться общими для различных машин Единой системы. Аппаратные средства системы прерываний и приоритетов различны. Так, 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-1022, запросы на прерывания, а также коды масок отдельных прерываний размещаются в триггерах регистра прерываний и отдельных триггерах (см. п. 7.2); запросы на некоторые программные прерывания не фиксируются и выявляются при реализации соответствующих микропрограмм. Анализ содержимого триггеров, хранящих запросы на прерывания и коды масок, производится по микрооперациям полей микрокоманд; при выявлении незамаскированного запроса на прерывание осуществляется передача управления первой команде программы, производящей обработку

запроса на прерывание. Состояние процессора на момент прерывания его работы запоминается, чтобы после завершения прерывающей программы, т. е. программы обработки запроса на прерывание, можно было бы возобновить работу по прерванной программе.

Вся основная информация, отражающая состояние процессора и необходимая для возобновления выполнения прерванной программы, содержится в слове состояния программы (ССП), имеющем формат двойного слова. Общее назначение ССП заключается в управлении порядком выборки и исполнения команд, а также в фиксации состояния всей машины (системы) по отношению к выполняемой программе. При этом выполнение любой программы происходит под управлением конкретного ССП, относящегося только к данной программе. Общая структура ССП приведена на рис. 7.18. Восьмиразрядная маска системы включает коды следующих масок: нулевой разряд — маска мультиплексного канала; 1, 2, 3, 4, 5, 6-й разряды — маски селекторных каналов с такими же номерами (в ЕС-1030 и ЕС-1033 не используются 4, 5 и 6-й разряды, в ЕС-1020 и ЕС-1022 — 3, 4, 5 и 6-й разряды); 7-й разряд — маска внешних прерываний. В разрядах 8—11 содержится ключ защиты, или ключ программы. В разрядах «Состояние» отражается следующее: 12-й разряд — режим работы с КОИ-8, 13-й — маска прерываний от схем контроля машины (прерываний по машинному сбою), 14-й — состояние ОЖИДАНИЕ, 15-й — состояние ЗАДАЧА. Если разряд маски содержит нуль, то соответствующее прерывание считается игнорируемым и не выполняется; маски каналов относятся к прерываниям от устройств ввода-вывода, подключенным к этим каналам.

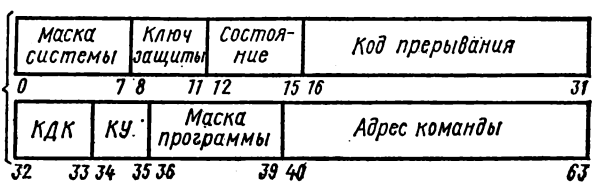


Рис. 7.18. Слово состояния программы

Код длины команды КДК отражает количество полуслов для выполняемой команды; в случае прерывания к этой команде обеспечивается возврат после обработки прерывания. Формирование КДК производится в ходе выборки команды; он устанавливается равным единице для команд формата RR, двум — для команд форматов RS, RX, SI и трем — для команд формата SS. Двухразрядный код условия КУ отражает результат большинства арифметических и логических операций, а также операций ввода-вывода. Установленный в процессе выполнения некоторой команды код условия не изменяется до тех пор, пока по одной из последующих команд не будет сформирован другой КУ. 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 пере-

полнения, 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; внутри каждой пары состояния взаимно исключают друг друга. Все возможные состояния составляют следующие пары:

1) ОСТАНОВ – РАБОТА;

2) СЧЕТ (ВЫПОЛНЕНИЕ ПРОГРАММЫ) – ОЖИДАНИЕ;

3) ПРЕРЫВАНИЕ ЗАМАСКИРОВАНО – ПРЕРЫВАНИЕ РАЗРЕШЕНО;

4) СУПЕРВИЗОР – ЗАДАЧА.

Последние три пары состояний отражаются в ССП. Первая пара состояний в ССП не отражается, так как нет никаких программных средств для перехода из состояния останова в рабочее состояние и наоборот. Функции всех состояний, их индикация и переключение не зависят друг от друга.

Переключение состояний не влияет на содержимое регистров арифметическо-логического устройства и выполнение операций ввода-вывода, но может повлиять на работу таймера.

Состояние ОСТАНОВ характеризуется тем, что команды не выполняются, а запросы на прерывания игнорируются. Перевод в это состояние, так же как и в состояние РАБОТА, может производиться только вручную при нажатии соответствующих кнопок (клавиш) на пульте управления. Процессор переводится в состояние ОСТАНОВ при нажатии кнопок ОСТАНОВ, ПИТАНИЕ ВКЛ, ГАШЕНИЕ, а также при реализации с пульта управления останова по адресу или работы в режиме «Команда» (после завершения выполнения одной команды). Перевод процессора в состояние РАБОТА обеспечивается только при нажатии кнопки ПУСК, а также при первоначальной загрузке программы. Отметим, что переход из состояния РАБОТА в состояние ОСТАНОВ происходит только после завершения выполняемой команды.

Пары состояний, отражаемые в ССП, могут задаваться только программным путем за счет изменения значений соответствующих разрядов ССП. Так, если 14-й разряд ССП содержит единицу, то процессор находится в состоянии ОЖИДАНИЕ, если же он содержит нуль, то это индицирует состояние СЧЕТ. Когда процессор находится в состоянии ОЖИДАНИЕ, команды не выполняются, но запросы на внешние и вводо-выводные прерывания воспринимаются и обрабатываются. В состоянии СЧЕТ происходит нормальная выборка и выполнение команд. Переключение этих состояний, т. е. изменение значений 14-го разряда ССП, производится только при занесении нового ССП в качестве текущего, т. е. реализуемого; в любом из них продвижение таймера, отвечающее счету времени, не прекращается.

Третья пара состояний как бы отражает отношение процессора к прерываниям. Процессор реализует прерывание только тогда, когда это прерывание не маскируется. Запрос на прерывание игнорируется, т. е. процессор находится в состоянии ПРЕРЫВАНИЕ ЗАМАСКИРОВА-

НО, если в соответствующем разряде ССП содержится код 0; при коде 1 прерывание разрешается. Замаскированные запросы на прерывания по вводу-выводу, внешние прерывания и прерывания по сбою машины сохраняются для обработки в дальнейшем; замаскированные запросы на программные прерывания не сохраняются. 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 может быть изменена только засылкой нового ССП. Значения всех масок могут быть изменены при выполнении команды ЗАГРУЗИТЬ ССП, при прерывании и при первоначальной загрузке программы. Смена масок не влияет на продвижение таймера.

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 принятой для ЕС ЭВМ системы команд. Состоянию СУПЕРВИЗОР отвечает код 0 в 15-м разряде ССП, а состоянию ЗАДАЧА – код 1 в этом же разряде. Переключение рассматриваемых состояний возможно при помощи команды ЗАГРУЗИТЬ ССП, а также при прерываниях и первоначальной загрузке программы; новое ССП задает любое из этих состояний независимо от предшествующего; смена таких состояний не влияет на продвижение таймера.

Прерывания. Все возможные прерывания и соответствующие им источники запросов на прерывания в машинах Единой системы разделены на пять групп, или пять классов:

прерывания от схем контроля (по сбою машины), или прерывания по контролю машины;

программные прерывания, или прерывания по ошибкам программы;

прерывания для обращения к супервизору (прерывания по вызову супервизора);

внешние прерывания, или прерывания по сигналам от внешних источников;

вводо-выводные прерывания, или прерывания от устройств ввода-вывода.

Все ситуации, вызывающие прерывание данного класса, анализируются (обрабатываются) отдельной программой, являющейся частью супервизора. Таким образом, пяти классам прерываний отвечают пять супервизорных программ. Общая процедура прерывания состоит в передаче управления от выполняемой программы к супервизорной программе обработки прерываний соответствующего класса.

Любая программа выполняется в машине с учетом информации, содержащейся в ССП. Слово состояния программы, отвечающее выполняемой в данный момент времени программе, называется текущим. При прерывании это ССП должно запоминаться в оперативной памяти, а на его место должно вводиться ССП, отвечающее прерывающей программе. Поэтому каждому классу прерываний отвечают два ССП, называемых соответственно старым и новым; они размещаются в определенных

(фиксированных) ячейках оперативной памяти. Для большинства машин Единой системы принято, что в случае внешних прерываний старое ССП размещается в ячейке ОП с десятичным номером 24, а новое ССП – в ячейке 88; для прерываний по обращению к супервизору старое и новое ССП размещаются соответственно в ячейках ОП с номерами 32 и 96, для программных прерываний – в ячейках 40 и 104, для прерываний от схем контроля – в ячейках 48 и 112 и для вводо-выводных прерываний – в ячейках 56 и 120.

В самом общем плане 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, обеспечивает управление выполнением супервизорной программы обработки запроса на прерывание. После завершения супервизорной программы обычно производится возврат к прерванной программе. Для этого используется концевая команда

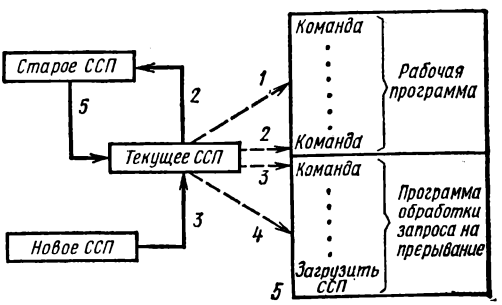


Рис. 7.19. Общая процедура прерывания: 1 – выполнение рабочей (проблемной) программы; 2 – начальный этап прерывания, связанный с выявлением запроса на прерывание определенного класса и отсылкой текущего ССП в ячейку старого ССП; 3 – этап прерывания, связанный с началом обработки запроса на прерывание; 4 – обработка запроса; 5 – восстановление прерванного состояния.

ЗАГРУЗИТЬ ССП, по которой текущее ССП заменяется на старое, т. е. система возвращается в состояние, предшествующее прерыванию. Общая процедура прерывания схематично отображена на рис. 7.19.

В большинстве случаев прерывания производятся после того, как заканчивается выполнение текущей команды и до начала выполнения следующей команды. Результаты выполнения завершенной команды заносятся в память, а код условия устанавливается в обычном порядке. При программных прерываниях и прерываниях по сбою машины выполнение команды обычно прекращается с занесением в память результата или его части; занесение в память может и не производиться. Отдельным случаям прерывания отвечает подавление выполняемой команды, что равносильно ее пропуску; результат в память не заносится, код условия остается без изменений.

Для обеспечения возможности обработки запроса на прерывание в старом ССП фиксируется причина прерывания (код прерывания), а также информация, позволяющая установить расположение в памяти последней выполнявшейся до прерывания команды (адрес команды и код ее длины). При вводо-выводных прерываниях в память заносится дополнительная информация, представляющая собой содержимое слова

Таблица 7.1

Прерывания

Источник прерывания	Код прерывания	Разряд 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 bbbbbbbb	—	1	Завершается
4. Внешние прерывания				
1) Программа ВРЕМЯ, или таймер	00000000 1ccccccc	7	—	Завершается
2) Клавиша ПРЕРЫВАНИЕ на ПУ	00000000 c1cccccc	7	—	»
3) Внешний сигнал 2	00000000 cc1ccccc	7	—	»
4) Внешний сигнал 3	00000000 ccc1cccc	7	—	»
5) Внешний сигнал 4	00000000 cccc1ccc	7	—	»
6) Внешний сигнал 5	00000000 ccccc1cc	7	—	»
7) Внешний сигнал 6	00000000 cccccc1c	7	—	»
8) Внешний сигнал 7	00000000 ccccccc1	7	—	»
5. Прерывания ввода-вывода				
1) Канал 0 (МК)	00000000 aaaaaaaa	0	—	Завершается
2) Канал 1 (СК-1)	00000001 aaaaaaaa	1	—	»
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	—	»

состояния канала. Основная информация по всем классам и возможным источникам прерываний отражена 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. Отметим, что код прерывания в текущем ССП не имеет смысла, так как он формируется во время прерывания. Фиксация кода прерывания в слове состояния программы производится в соответствии с данными табл. 7.1 только при его запоминании в ячейке (ячейках) оперативной памяти машины.

Текущее ССП обеспечивает управление выполнением любой программы. В машинах ЕС-1033, ЕС-1050, ЕС-1052 оно хранится в специальном регистре слова состояния программы (РССП). 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 машинах ЕС-1020, ЕС-1022 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 памяти. Для машины ЕС-1022 места размещения отдельных частей ССП отражены данными табл. 7.2, в которой номера ячеек ЛП представлены в шестнадцатиричной системе счисления.

Таблица 7.2

Размещение частей ССП

Части текущего ССП	Место расположения		
	Локальная память		Регистры процессора
	Номер ячейки	Разряды	
Маска системы	30	[0—7]	РП0 [0—2] и РП0 [7]
Ключ защиты	30	[8—11]	РП1 [0—3]
Режим работы с КОИ-8	30	[12]	—
Маска прерываний от схем контроля машины	30	[13]	РП0 [5]
Состояние ОЖИДАНИЕ	30	[14]	РП0 [6]
Состояние ЗАДАЧА	30	[15]	—
Код прерывания	—	—	—
Код длины команды	—	—	РС1 [0—1]
Код условия	—	—	РС0 [2—3]
Маска программы	31	[4—7]	—
Адрес команды	32 и 33	—	РАИ [0—23]

Приоритеты прерываний. В процессе работы машины могут быть одновременно сформированы запросы на прерывания различных классов. Для однозначного удовлетворения таких запросов введены приоритеты прерываний по классам. Программные прерывания и прерывания для обращения к супервизору взаимно исключают друг друга, так как запросы на них не могут формироваться одновременно. Поэтому прерывания этих классов имеют одинаковый приоритет. Глубина прерываний отвечает четырем принятым приоритетам для различных классов прерываний.

Самый высокий приоритет (приоритет 1) имеет прерывание по сбою машины. При его реализации действия по текущей команде прекращаются. Программное прерывание или прерывание для обращения к супервизору, которое могло произойти в результате выполнения текущей команды, игнорируется. В то же время сбои машины обычно не влияют на вводо-выводные и внешние прерывания, продвижение таймера и передачу данных при вводе-выводе информации по селекторному каналу.

Для остальных классов прерываний рассматриваются не только приоритеты по реализации, но и приоритеты по восприятию, что объясняется принятым порядком загрузки и реализации ССП. После прерывания по сбою машины в порядке общей приоритетности, или приоритетности по реализации, следуют вводо-выводные, внешние и программные прерывания, включая прерывания для обращения к супервизору. Восприятие запросов на указанные прерывания, если они формируются одновременно, производится в обратном порядке; поэтому приоритеты вводо-выводных, внешних и программных прерываний по восприятию нумеруются, начиная с четвертого, т. е. 4, 3, 2. Если выполняется некоторое прерывание, то процессор маскируется для других прерываний этого же класса.

Если одновременно имеются запросы на программные (вызов супервизора), внешние и вводо-выводные прерывания, то происходит следующее. Текущее ССП, т. е. СПП для прерываемой рабочей программы, запоминается в ячейке 40 оперативной памяти, а новое ССП загружается на место текущего из ячейки 104. Так как есть запросы на другие прерывания, то это ССП не реализуется, т. е. программа обработки не выполняется. Текущее ССП, ставшее новым ССП программных прерываний, запоминается в ячейке 24, а вместо него загружается новое ССП из ячейки 88 – ССП внешних прерываний. После загрузки этого ССП выявляется запрос на вводо-выводное прерывание, поэтому программа обработки внешних прерываний также не выполняется. Снова текущее ССП, которым теперь является новое ССП внешних прерываний, запоминается, на этот раз в ячейке 56; в качестве текущего становится выбираемое из ячейки 120 новое ССП – ССП вводо-выводных прерываний. Отметим, что при реализации прерывания для обращения к супервизору используют не ячейки 40 и 104, а 32 и 96 соответственно.

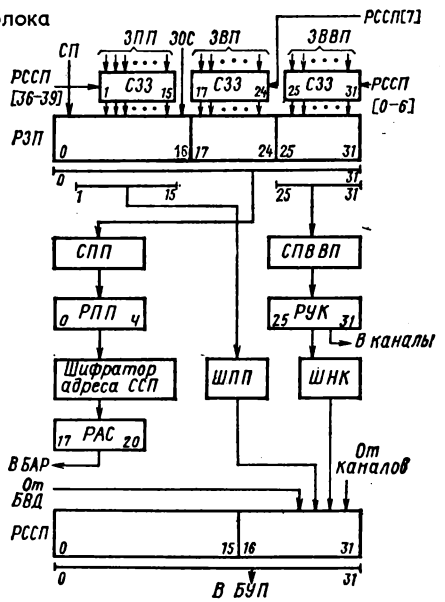
Реализация учтенных запросов на прерывания производится в порядке истинной приоритетности прерываний. Так как после всех загрузок текущим стало ССП вводо-выводных прерываний, то в рассматриваемом случае оно реализуется первым. В конце программы обработки прерывания от ввода-вывода осуществляется загрузка старого ССП для этого прерывания из ячейки 56. Теперь текущим становится ССП внешних прерываний, и реализуется соответствующая программа обработки. В конце этой программы выполняется загрузка ССП из ячейки 24, как ячейки старого ССП для внешних прерываний. Текущим становится ССП программных прерываний; выполняется отвечающая им программа обработки, в конце которой загружается ССП из ячейки 40. Тем самым осуществляется переход к прерванной рабочей программе, которая выполнялась до реализации запросов на прерывания.

При одновременном поступлении запросов на вводо-выводные прерыва-

ния от нескольких источников их удовлетворяют в соответствии с приоритетной шкалой. Высшим приоритетом обладают запросы мультиплексного канала; для селекторных каналов приоритеты устанавливаются 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:

Рис. 7.20. Структура блока прерываний



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 вводно-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.

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 вентилей, 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 пять 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:

РПП[0]: = 1 при РЗП[0] = 1,

РПП[1]: = 1 при $\overline{\text{РЗП}[0]} \wedge \text{РЗП}[1-15] = 1$,

РПП[2]: = 1 при $\overline{\text{РЗП}[0-15]} \wedge \text{РЗП}[16] = 1$,

РПП[3]: = 1 при $\overline{\text{РЗП}[0-16]} \wedge \text{РЗП}[17-24] = 1$,

РПП[4]: = 1 при $\overline{\text{РЗП}[0-24]} \wedge \text{РЗП}[25-31] = 1$;

обозначение вида РЗП[1 — 15] указывает на то, 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; 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; оттуда они направляются в блок управления памятью для организации обращения к ОП.

Схема приоритета вводо-выводных прерываний и регистр управления каналами работают аналогично паре СПП — РПП. Но начинают работать они 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.

Схемы блока прерываний по обработке одного запроса на прерывание работают в два этапа. На первом этапе производятся все действия, связанные с записью текущего ССП на место старого, а на втором этапе — 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 вводо-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 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 большой 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.

Управляющие сигналы и информация от блоков памяти в БУП передаются по 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

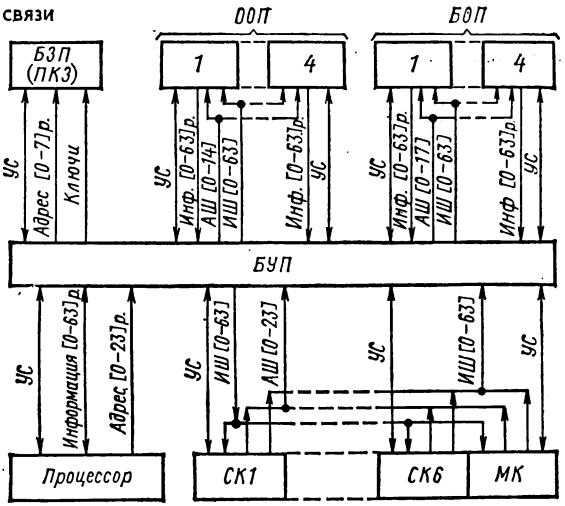


Рис. 7.21. Основные внешние связи блока управления памятью

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 ООП, 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.

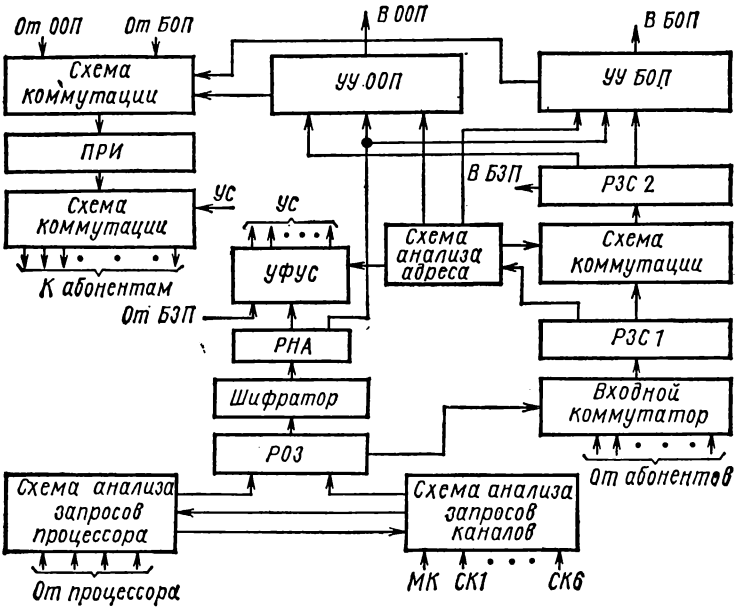


Рис. 7.22. Общая структура блока управления памятью

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 обеими 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, занисываемыми 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 имеющихся 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 производятся 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 запроса 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 нечетность, 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 ООП 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, понимают 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.

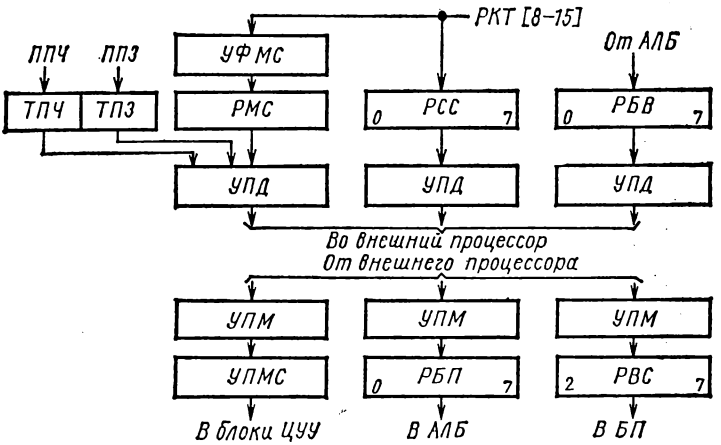


Рис. 7.23. Упрощенная схема блока внешних связей

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), РБВ — 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.

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 таймера 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, где 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.

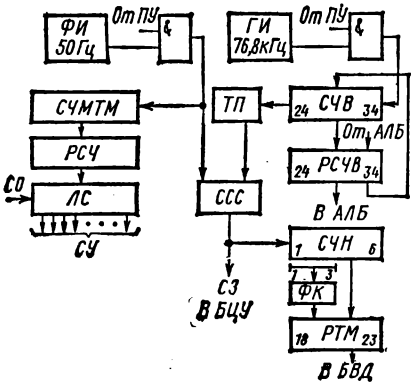


Рис. 7.24. Структура блока таймеров

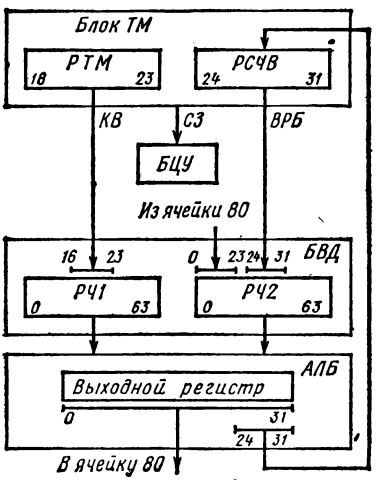


Рис. 7.25. Внешние связи блока таймеров

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 работе непрерывно взаимодействует с блоками процессора, используя свои внешние связи, показанные на рис. 7.25. Дополнительные обозначения для этого рисунка: КВ — код

вычитания, сформированный 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 большей, 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 каждые 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 – 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, памяти основных видов, каналов и устройства питания. Принципы построения ПУ, во многом общие для различных моделей, рассмотрим на примере пульта управления машины ЕС-1022.

В составе пульта управления функционально выделены: блок выдачи информации, блок сравнения адресов, блок управления, блок набора информации, регистр индикации, блок причин ручных операций и счетчика времени, или регистр ДПО, блок выработки управляющих сигналов и блок индикации (рис. 7.26). Блок выдачи информации обеспечивает передачу информации из регистров пульта управления в локальную

память ЛП. Блок сравнения адресов служит для сравнения адреса, установленного на переключателях пульта управления, с содержимым регистра адреса постоянной памяти ПП или регистра адреса оперативной памяти ОП. Блок управления преобразует сигналы, поступающие с кнопок и переключателей пульта, а также из блока сравнения адресов, в сигналы, необходимые для управления работой центрального процессора ЦП.

Блок причин ручных операций и счетчика времени в качестве своей основы имеет восьмиразрядный триггерный регистр, обозначаемый шифром

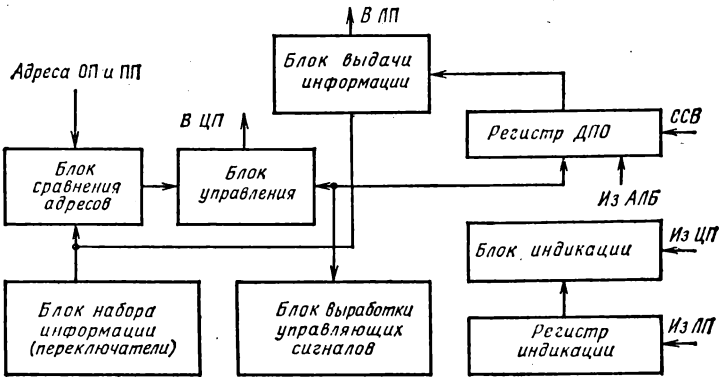


Рис. 7.26. Функциональный состав пульта управления

ДПО. Разряды [0—3] этого регистра 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 тока) поступают сигналы счета времени (ССВ) стандартной формы. Максимальное содержимое счетчика отвечает временному интервалу, равному 300 мс. В процессе работы машины содержимое ДПО [4—7] используется для изменения значения таймера — числа, размещенного в стандартных ячейках (50—53) ОП и фиксирующего временные затраты на выполнение различных работ. При нажатии соответствующей кнопки на пульте управления в ДПО[0—3] устанавливается признак выполняемой операции, который сбрасывается микропрограммно только по окончании этой операции.

Регистр индикации представляет собой 32-разрядный триггерный регистр, в который может приниматься информация из локальной и оперативной памяти, регистров процессора и каналов, не имеющих постоянной индикации.

Блок набора информации содержит переключатели для установки адреса памяти, типа памяти и адреса команды. Блок выработки управляющих сигналов содержит кнопки и переключатели; при нажатии той или иной кнопки управления в ДПО[0—3] устанавливается признак операции, а через блок управления выдается сигнал для запуска процессора на выполнение соответствующей операции; при помощи переключателей управления задается режим работы процессора. Блок индикации

обеспечивает индикацию состояния регистров и отдельных управляющих триггеров, характеризующих работу процессора. Элементы управления лампочками индикации размещаются в соответствующих функциональных частях процессора.

Регистры и наборы переключателей информации, установленные на пульте управления и рассматриваемые как регистры, подключаются к процессору через блок выдачи информации в ЛП. Обмен информацией осуществляется через регистр данных локальной памяти. Управление передачами информации, как и всей работой пульта управления, обеспечивается двумя микропрограммами: обслуживания пультовых операций (ОБСП) и первоначальной загрузки программ (ПЗП).

Конструктивно пульт управления состоит из двух панелей: панели управления и панели индикации.

Панель управления. На панели управления (рис. 7.27) расположены следующие элементы блока набора информации и блока управляющих сигналов:

переключатели набора информации АДРЕС ПАМЯТИ (А – Д) и АДРЕС КОМАНДЫ (Е – К);

переключатели управления РЕЖИМ РАБОТЫ, КОНТРОЛЬ, СРАВНЕНИЕ АДРЕСОВ, ТИП ПАМЯТИ, ДИАГНОСТИКА;

кнопки управления.

Набор переключателей АДРЕС ПАМЯТИ состоит из пяти переключателей А, Б, В, Г, Д на 16 положений каждый 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 центрального процессора при передаче их содержимого в регистр индикации на переключателе Д;

г) номера байта слова состояния программы ССП при считывании или записи на переключателе Д;

д) адреса ячейки (регистра) локальной памяти (на переключателях Г и Д);

е) адреса ячейки памяти ключей защиты.

Набор переключателей АДРЕС КОМАНДЫ состоит из пяти переключателей (Е, Ж, З, И, К) на 16 положений каждый и предназначен для задания:

а) адреса текущей команды на переключателях Е – К;

б) адреса микрокоманды, или адреса постоянной памяти, на переключателях Ж – К;

в) адреса (номера) внешнего устройства при вводе начальной программы на переключателях З – К;

г) данных (на переключателях И – К).

Переключатель ТИП ПАМЯТИ в зависимости от своего положения указывает, к какой памяти производится обращение с пульта управления:

ОП – к основной оперативной 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 1-го селекторного канала;
КС2 – к регистрам 2-го селекторного канала;
ССП – к слову состояния программы.

Переключатель СРАВНЕНИЕ АДРЕСОВ указывает, с адресом какой памяти (оперативной или постоянной) должно производиться сравнение адреса, установленного на наборе переключателей АДРЕС ПАМЯТИ, и определяет реакцию процессора при сравнении этих адресов; в зависимости от положения переключателя эта реакция выражается как:

останов по команде или микрокоманде;

переход по адресу, заданному на переключателях АДРЕС КОМАНДЫ;

останов при записи (регенерации) в оперативную память по адресу, установленному на наборе переключателей АДРЕС ПАМЯТИ;

выдача синхроимпульса при считывании заданной микрокоманды;

в положении АВТОМАТ сравнение адресов не производится.

Переключатель РЕЖИМ РАБОТЫ позволяет задавать различные режимы работы центрального процессора:

автоматический режим («Автомат») – основной режим работы центрального процессора при выполнении различных 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 оперативной памяти.

Переключатель КОНТРОЛЬ определяет своим положением реакцию процессора при обнаружении сбоя:

ОСТАНОВ – при возникновении сбоя производится останов центрального процессора; причина сбоя индицируется в регистре ошибок;

БЛОКИРОВКА – осуществляется блокировка всех сигналов сбоев;

АВТОМАТ – производится обработка данных, связанных с сигналом сбоя, по специальной программе.

Переключатель ДИАГНОСТИКА своим положением задает выполнение диагностических микропрограмм:

КМ – мультиплексного канала;
КС1 – первого селекторного канала;
КС2 – второго селекторного канала;
ЦП – центрального процессора;
ОП – оперативной памяти;
ПП – постоянной памяти;
ЛП – локальной памяти;

в положении АВТОМАТ диагностические микропрограммы не выполняются; их запуск осуществляется кнопкой ПУСК.

На панели управления расположено 14 кнопок управления, функционально относящихся к блоку выработки управляющих сигналов ПУ; двенадцать из них позволяют задавать различные режимы работы и вызывать определенные действия в основных устройствах машины (системы), а две служат для включения и выключения электропитания.

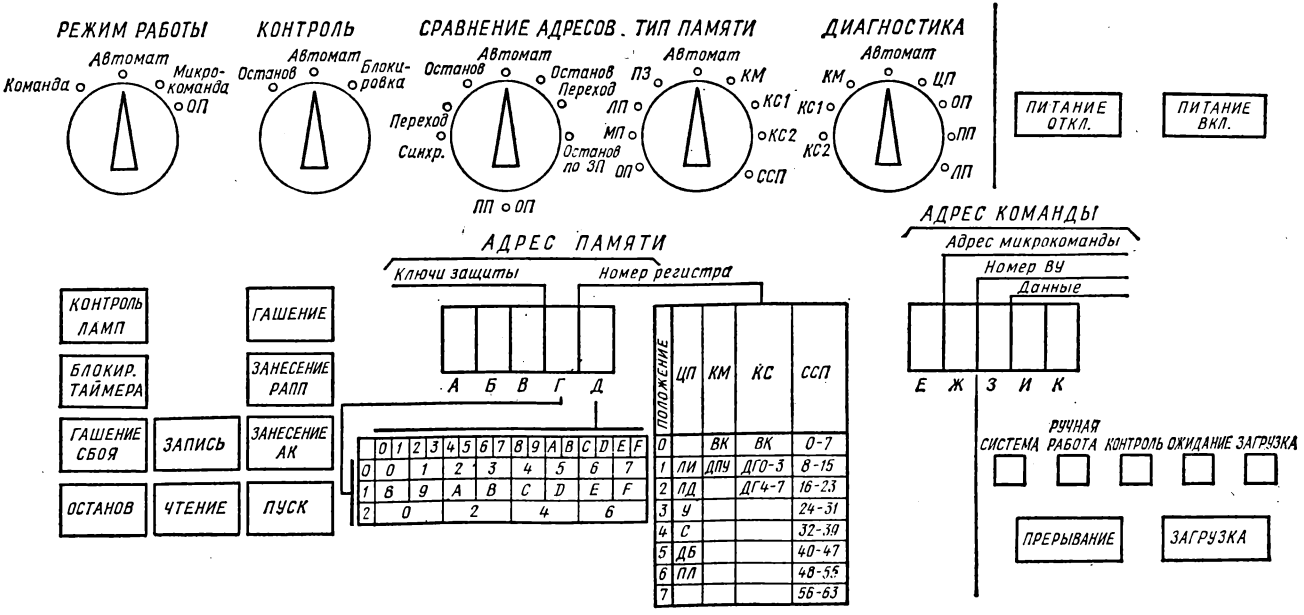


Рис. 7.27. Панель управления

Кнопка ГАШЕНИЕ предназначена для перевода машины в исходное состояние независимо от того, в каком состоянии она до этого находилась (если питание включено). В исходном состоянии подсвечиваются кнопка ОСТАНОВ и табло РУЧНАЯ РАБОТА.

Кнопка ЗАГРУЗКА предназначена для запуска микропрограммы первоначальной загрузки программы ПЗП, когда содержимое памяти или слова состояния программы непригодно для дальнейшей обработки информации. Во время выполнения первоначальной загрузки программы подсвечивается табло (лампочка) ЗАГРУЗКА.

При нажатии кнопки ОСТАНОВ процессор переводится в состояние останова после выполнения текущей команды и обслуживания всех запросов на прерывания. В состоянии останова можно выполнять операции чтения и записи, а также производить занесение адреса команды и микрокоманды. Если процессор находится в состоянии останова, то подсвечивается кнопка ОСТАНОВ.

Кнопка ПУСК предназначена для инициирования выполнения операции в том режиме, который задан положением переключателя РЕЖИМ РАБОТЫ или переключателя ДИАГНОСТИКА.

Кнопка ЗАНЕСЕНИЕ АК предназначена для занесения адреса текущей команды, установленного на наборе переключателей АДРЕС КОМАНДЫ, в счетчик адресов команд (регистр РАИ процессора).

Кнопка ЗАНЕСЕНИЕ РАПП предназначена для занесения адреса микрокоманды, установленного на соответствующих переключателях, в регистр адреса постоянной памяти.

Кнопка ЗАПИСЬ предназначена для записи информации, набранной на переключателях И, К, в основную оперативную, мультиплексную, локальную память, ячейки хранения ССП, память ключей защиты по адресу, установленному на наборе АДРЕС ПАМЯТИ. Запись информации производится по соответствующим микрокомандам; во время выполнения операции кнопка ЗАПИСЬ подсвечивается.

Кнопка ЧТЕНИЕ служит для вывода на индикацию содержимого ячеек памяти различных видов, регистров процессора и каналов, а также байт ССП. По отношению к ячейкам памяти и байтам ССП вывод на индикацию осуществляется микропрограммно, а по отношению к регистрам — аппаратно. Если нужно прочитать содержимое ячейки памяти или байта ССП, то при помощи переключателя ТИП ПАМЯТИ задается признак памяти, а на наборе АДРЕС ПАМЯТИ устанавливается адрес требуемой ячейки или номер части ССП. Считанная из ячейки информация индицируется в регистре индикации. Для чтения содержимого требуемого регистра переключателем ТИП ПАМЯТИ задается один из каналов или ЦП, переключатель РЕЖИМ РАБОТЫ устанавливается в положение МИКРОКОМАНДА, а на переключателе Д набора АДРЕС ПАМЯТИ указывается номер регистра; считанная из выбранного регистра информация принимается в регистр индикации. Во время выполнения рассматриваемой операции кнопка ЧТЕНИЕ подсвечивается.

Кнопка ПРЕРЫВАНИЕ предназначена для формирования запроса на внешнее прерывание, если оно не маскировано. При ее нажатии в 25-м разряде текущего ССП устанавливается единица; это указывает

на то, что источником внешнего прерывания является кнопка ПРЕРЫВАНИЕ.

Кнопка ГАШЕНИЕ СБОЯ предназначена для сброса в исходное состояние всех индикаторов сбоев процессора.

Кнопка КОНТРОЛЬ ЛАМП служит для проверки элементов и ламп системы индикации.

Кнопка БЛОКИР. ТАЙМЕРА служит для управления работой счетчика временных интервалов. Если эта кнопка подсвечивается, счет временных интервалов не производится.

Кнопка ПИТАНИЕ ВКЛ. нажимается для инициирования последовательности действий по включению питания. При включенном питании эта кнопка подсвечивается.

Кнопка ПИТАНИЕ ОТКЛ. предназначена для инициирования последовательности действий по отключению питания. Кнопка подсвечивается, если питание отключено.

На панели управления расположены также пять лампочек (табло), индицирующих различные состояния процессора:

СИСТЕМА – процессор выполняет микропрограмму (любую);

РУЧНАЯ РАБОТА – процессор находится в состоянии останова или выполняется операция с пульта управления;

КОНТРОЛЬ – один из переключателей СРАВНЕНИЕ АДРЕСОВ, КОНТРОЛЬ, РЕЖИМ РАБОТЫ или ДИАГНОСТИКА не находится в положении АВТОМАТ;

ОЖИДАНИЕ – процессор находится в состоянии ожидания работ;

ЗАГРУЗКА – выполняется программа первоначальной загрузки.

Панель индикации. На панели индикации размещены элементы, обеспечивающие индицирование содержимого регистров постоянной памяти, содержимого некоторых регистров и состояния отдельных управляющих триггеров процессора, мультиплексного и селекторных каналов.

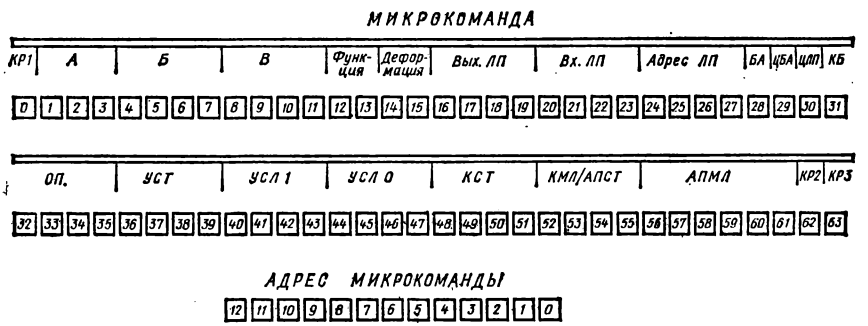


Рис. 7.28. Индикация регистров постоянной памяти

Индикация регистров постоянной памяти. На панели индикации всегда представляется содержимое регистра микрокоманды и регистра адреса микрокоманды (рис. 7.28). При работе в режиме «Микрокоманда» или при останове по адресу постоянной памяти в результате сравнения адресов индицируется содержимое и адрес той микрокоманды, которая

должна быть выполнена при следующем нажатии кнопки ПУСК. При других режимах работы указанные регистры находятся в состоянии динамического изменения содержащейся в них информации, и считывание данных об их содержимом по элементам индикации невозможно.

Индикация мультиплексного канала. На рассматриваемой панели пульта управления индицируется содержимое регистра выходной информации (ШИН-К), состояния признаков контроля и признаков интерфейса мультиплексного канала (рис. 7. 29). Через информационные шины канала ШИН-К устройствам ввода-вывода передаются адреса устройств, коды команд и байты данных. Находящаяся на ШИН-К информация всегда идентифицируется признаками интерфейса, которые индицируются одновременно с ней. Признаки интерфейса мультиплексного канала имеют то же значение, что и признаки интерфейса любого из селекторных каналов, поэтому при их рассмотрении используется понятие внешнего устройства, обобщающее понятия устройств ввода-вывода и внешних запоминающих устройств.

На панели индикации подсвечиваются следующие признаки интерфейса:

РАБ-А (работа внешнего устройства — абонента) — внешнее устройство выбрано и подключено к каналу;

АДР-А (адрес от абонента) — внешнее устройство установило на выходные информационные шины (ШИН-А) свой адрес;

УПР-А (управление от абонента) — внешнее устройство установило на ШИН-А байт состояния;

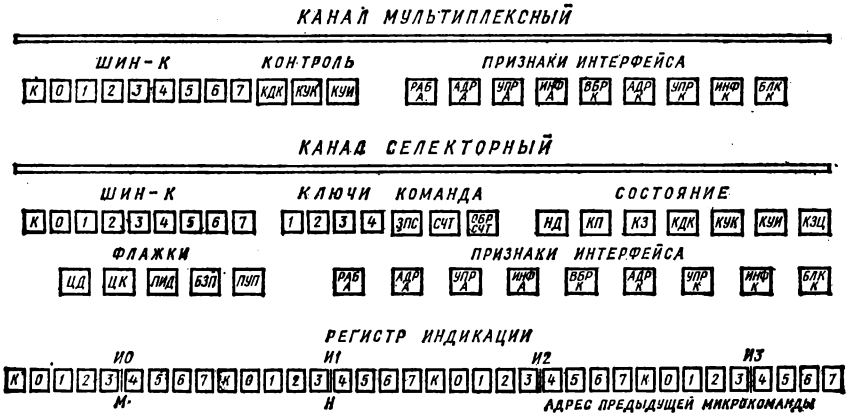


Рис. 7.29. Индикация каналов

ИНФ-А (информация от абонента) — выбранное внешнее устройство готово к передаче или приему байта информации;

ВБР-К (выборка от канала) — канал опрашивает внешние устройства для определения того из них, 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 ему «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, что сигнал ИНФ-А или УПР-А, поступивший от внешнего устройства, воспринят;

БЛК-К (блокировка от канала) — канал совместно с другими сигналами указывает о блокировке данных, блокировке состояния, цепочке команд канала (зацеплении по команде), селективном сбросе.

Отражаемые на панели индикации признаки контроля мультиплексного канала означают:

КУК (контроль управления канала) — на линиях интерфейса одновременно имеют место несовместимые признаки идентификации сигналов канала (УПР-К и ИНФ-К, АДР-К и ИНФ-К и т. д.); обнаружен сбой в оборудовании канала при передаче информации;

КДК (контроль данных канала) — принятый через интерфейс байт данных содержит неверный контрольный разряд;

КУИ (контроль управления интерфейса): а) на линиях интерфейса одновременно действуют несовместимые сигналы идентификации абонента, например ИНФ-А и УПР-А, УПР-А и АДР-А и т. д.; б) байт адреса или байт состояния ВУ, принятый через интерфейс каналом, содержит неверный контрольный разряд.

Индикация селекторных каналов. На панели пульта управления размещаются элементы индикации выходных информационных шин селекторного канала ШИН-К, регистра ключей защиты памяти, регистра команд, регистра состояния, регистра флажков и признаков интерфейса (см. рис. 7.29); выбор индицируемого селекторного канала осуществляется с помощью простого переключателя на два положения, расположенного в правой части панели индикации.

С помощью элементов индикации для ШИН-К индицируются передаваемые от канала к ВУ адреса устройств, коды команд, байты данных, а также принимаемые от устройств байты состояния и байты данных. Находящаяся на ШИН-А информация идентифицируется соответствующими признаками интерфейса, указанными при рассмотрении индикации мультиплексного канала.

Регистр ключей защиты содержит ключ защиты памяти, установленный в канале при выполнении инструкции НАЧАТЬ ВВОД-ВЫВОД.

Регистр команд содержит следующие признаки:

ЗПС (запись) — канал передает информацию из ОП во внешнее устройство;

СЧТ (считывание) — канал принимает информацию из внешнего устройства 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 байт, затребованных или передаваемых ВУ;

КП (контроль программы) — канал выявил программную ошибку;

КЗ (контроль защиты) — канал «пытался» записать (или прочитать) данные в защищенной области памяти, а ключ защиты данных 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 действуют несовместимые сигналы идентификации канала, например УПР-К и ИНФ-К; ИНФ-К и АДР-К и т. д.; в оборудовании канала обнаружен сбой при передаче информации;

КУИ (контроль управления интерфейса) — на линиях интерфейса одновременно действуют несовместимые сигналы идентификации абонента, например ИНФ-А и УПР-А, ИНФ-А и АДР-А и т. д.; байт адреса или байт состояния, принятый каналом от внешнего устройства, 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 регистра флажков отвечает признакам:

ЦД (цепочка данных) — в УСК задано зацепление по данным;

ЦК (цепочка команд) — в УСК задано зацепление по команде;

ПИД (подавление индикации неверной длины) — указание о неверной длине при выполнении операции ввода-вывода блокируется, если нет ЦД;

БЗП (блокировка записи) — операция передачи данных будет выполняться без записи информации в оперативную память;

ПУП (программно-управляемое прерывание) — в УСК был задан признак ПУП, и канал будет формировать запрос на прерывание после ввода в него этого УСК.

Регистр индикации обеспечивает индикацию содержимого регистров процессора и каналов, не имеющих постоянной индикации, а также слова состояния программы и содержимого ячеек памяти; он является четырехбайтным с выделением контрольных разрядов на каждый байт (рис. 7.30). Если выводимая на индикацию информация содержит меньше 4 байт, то старшие разряды, не заполненные данными, гасятся. В случае индикации ССП при положениях 0—3 переключателя Д в регистр индикации выводятся разряды ССП [0 — 31], а при положениях 4 — 7 — разряды [32—63]. В правой части панели индикации имеется переключатель РИ — АДРЕСА; положение РИ этого переключателя отвечает индикации 4 байт регистра индикации, а положение АДРЕСА — индикации адреса предыдущей микрокоманды и адресных регистров РМ и РН локальной памяти.

Индикация процессора. Элементы индикации состояний некоторых управляющих триггеров и регистров процессора, а также основной оперативной памяти показаны на рис. 7.30. Постоянно индицируются:

адресный регистр основной оперативной памяти;

регистр данных (Д0, Д1, Д2, Д3) основной оперативной памяти;

входные регистры (А0, А1 и Б0, Б1) и выход (В0, В1) арифметико-логического блока, обозначенного на рис. 7.30 как блок арифметический;

Рис. 7.30. Индикация основной оперативной памяти и процессора

регистр ошибок;

регистр косвенной функции РКФ.

В регистре ошибок индицируются следующие признаки сбоев:

БЗ – сбой в регистре защиты;

А – сбой во входном информационном регистре РА арифметико-логического блока;

Б – сбой во входном информационном регистре РБ арифметико-логического блока;

В – сбой на выходных шинах АЛБ;

РАОП – сбой в адресном регистре оперативной памяти;

РДОП – сбой в регистре данных оперативной памяти;

РАПП – сбой в адресном регистре постоянной памяти;

РМК – сбой в регистре микрокоманд;

РАЛП – сбой в адресном регистре локальной памяти;

РДЛП – сбой в регистре данных локальной памяти.

Индицируются состояния управляющих триггеров, отвечающих следующим состояниям процессора:

КМ – выполняется микропрограмма обслуживания мультиплексного канала;

КС1 – выполняется микропрограмма обслуживания первого селекторного канала;

КС2 – выполняется микропрограмма обслуживания второго селекторного канала;

ОСТ АДР – адрес, установленный на наборе АДРЕС ПАМЯТИ, равен адресу постоянной или оперативной памяти (в зависимости от положения переключателя СРАВНЕНИЕ АДРЕСОВ);

ТО – процессор находится в состоянии тяжелого останова.

Пульты управления других машин Единой системы строятся и используются по тем же принципам, что и рассмотренный ПУ машины ЕС-1022. Но они всегда отражают особенности соответствующей машины, ее комплектацию и режимы работы.

Глава 8

ВЗАИМОДЕЙСТВИЕ ОСНОВНЫХ УСТРОЙСТВ ЭВМ

Все операции в машинах Единой системы выполняются при тесном взаимодействии основных функциональных частей процессора и оперативной памяти, составляющих так называемое центральное ядро машины или системы. При выполнении операций ввода-вывода кроме устройств центрального ядра в работу включаются каналы и внешние устройства, 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, входящего в состав процессора обычной или магистральной структуры, а также реализации пультовых процедур, когда инициирование действий производится вручную с помощью соответствующих кнопок и переключателей пульта управления. Вопросы контроля выполнения операций подробно рассматриваются в гл. 16.

8.1. Выполнение пультовых процедур

Блоки, узлы и элементы пультов управления всех машин Единой системы позволяют выполнять различные действия, связанные как с проведением профилактических работ, так и с непосредственной реализацией рабочих программ. Их использование допускается в состоянии останова, когда в процессоре не выполняются никакие команды и не обслуживаются прерывания. Исключение составляют элементы, позволяющие переводить машину (систему) в состояние останова.

Посредством блоков, элементов и схем пульта управления можно инициировать и в конечном итоге выполнять следующие действия:

осуществлять гашение системы, т. е. приведение ее в состояние останова;

производить первоначальную загрузку программы (загрузку начальной программы);

заносить адрес команды в текущее слово состояния программы (для ЕС-1022 — в регистр РАИ);

производить запись информации в регистры процессора, ячейки памяти различных типов и ССП;

выводить на индикацию содержимое регистров центрального процессора, ячеек памяти и ССП;

заносить заданную на пульте информацию в регистр адреса постоянной памяти (если она используется в машине);

задавать реакцию процессора при сравнении адресов;

задавать реакцию процессора на сбои;

устанавливать режим работы процессора;

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, ЕС-1022, ЕС-1030 и ЕС-1033, гашение системы, первоначальная (начальная) загрузка программы, занесение адреса текущей команды или всего ССП, запись и считывание информации (кроме считывания информации из регистров процессора и каналов), а также обслуживание таймера, выполняются микропрограммно, а остальные функции реализуются схемным путем, т. е. аппаратно. Выполнение обслуживающих микропрограмм производится после нажатия соответствующих кнопок или клавиш на пульте управления. В ЕС-1050, ЕС-1052 все действия, связанные с выполнением пультовых процедур, реализуются схемным путем. При работе на пульте управления любой машины блокируются программные прерывания, машинные ошибки, сбои по защите и адресации.

Гашение. Процедура, переводящая систему (машину) в состояние, исходное для начала обработки информации. Под исходным состоянием системы понимается такое состояние, при котором может быть начата обработка информации без устранения неисправности и без предварительных проверок машины. Гашение системы выполняется 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 исходное нулевое состояние все регистры процессора, «гасятся» каналы и внешние устройства, аннулируются все прерывания, ждущие обслуживания. Затем в зависимости от используемого в машине принципа управления аппаратно или по специальной микропрограмме гашения производится проверка регистров процессора, корректируются контрольные разряды ячеек памяти и осуществляется перевод системы в состояние останова.

Первоначальная (начальная) загрузка программы. Процедура выполняется с целью ввода и запуска программ, когда содержимое памяти или слова состояния программы непригодны для дальнейшего использования; практически при первоначальной загрузке осуществляется смена ССП и содержимого памяти. Процесс первоначальной загрузки программы рассматривается ниже на примере ЕС-1022 при использовании схемы, приведенной на рис. 7.27.

Адрес внешнего устройства, с которого вводится первоначальная программа, задается на переключателях З, И, К. При нажатии кнопки ЗАГРУЗКА выполняется гашение системы, а затем — микропрограмма ПЗП, которая формирует управляющее слово канала (УСК); это УСК отсылается в канал, к которому подключено выбранное внешнее устройство. По принятому УСК канал передает в оперативную память машины

(системы) первый блок информации, состоящий из 24 байт. Эти начальные байты записываются в ячейках 0—23 основной оперативной памяти. Первые 8 байт составляют слово состояния программы ПЗП, а последующие — первое и второе УСК для ПЗП. Дальнейший ввод информации, т. е. выполнение операции ввода, производится под управлением первого УСК ПЗП, расположенного в ячейках 8—15 оперативной памяти. Второе УСК ПЗП (ячейки 16—23) используется в случае реализации цепочки данных или цепочки команд. По окончании первоначальной загрузки программы управление передается новому ССП, 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; это табло гасится после успешного завершения всех операций по первоначальной загрузке программы.

Основные действия по первоначальной (начальной) загрузке программы во всех машинах Единой системы выполняются в одной и той же последовательности с записью первых 24 байт в ячейки 0—23 оперативной памяти. Но в ЕС-1050, например, все сигналы, управляющие действиями по начальной загрузке программы, формируются схемным путем. Кроме того, адрес внешнего устройства, с которого производится прием информации в ОП, набирается на отдельной группе переключателей АДРЕС УСТРОЙСТВА; затем нажимается кнопка НЗП.

После нажатия кнопки НЗП с помощью триггеров управления формируется общий сигнал сброса, направляемый во все устройства машины. 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. Затем информация, набранная на переключателях пульта управления, вводится в РАТ[13—23] блока адреса результата (см. рис. 7.17). По РАТ[13—15] выбирается канал, в который направляется сигнал начала выполнения операции по вводу информации; содержимое РАТ[16—23] непосредственно передается в выбранный канал, где используется для формирования управляющего слова. Далее канал через блок управления памятью вводит начальные байты в ячейки 0—23 ОП; формирование адресов в канале производится по исходному управляющему слову. Затем из ячеек 8—15 канал выбирает УСК, по которому завершает выполнение операции ввода, если нет цепочки данных или цепочки команд.

По окончании работы канала блок выборки команд посылает запрос в блок управления памятью для выборки ССП. Поскольку при нажатии кнопки НЗП был произведен общий сброс и все адресные регистры находятся в исходном нулевом состоянии, то ССП выбирается из нулевой общей ячейки, куда оно было записано при начальной загрузке программы. Новое слово состояния программы вводится в РССП; машина готова к выполнению очередной программы.

Первоначальная (начальная) загрузка программы, как следует из изложенного выше, не имеет принципиальных различий для машин Единой системы, отличающихся построением центральных устройств управления.

Это относится и к другим пультовым процедурам, что позволяет их рассмотреть на примере машины ЕС-1022, пульт управления которой был описан в п. 7.7.

Занесение адреса команды в текущее ССП. Эта операция производится в машине ЕС-1022 микропрограммно. Заносимый адрес набирается на переключателях Е – К, после чего нажимается кнопка ЗАНЕСЕНИЕ АК. При нажатии этой кнопки начинается реализация микропрограммы обслуживания пульта управления (ОБСП), что приводит к побайтной передаче адреса, установленного на переключателях Е–К набора АДРЕС КОМАНДЫ, в поле счетчика адреса команды текущего ССП (регистр РАИ) и в регистр индикации. По состоянию регистра индикации осуществляется визуальный контроль правильности ввода. После ввода последнего байта адреса команды микропрограмма ОБСП обеспечивает перевод машины в состояние останова.

Запись информации. Операция записи информации с пульта управления позволяет вручную изменять содержимое ячеек памяти и текущее ССП. Для ее выполнения процессор переводится в состояние останова, и осуществляется запуск микропрограммы ОБСП. Адрес требуемой ячейки памяти или номер байта ССП устанавливается на переключателях АДРЕС ПАМЯТИ, тип памяти – на переключателе ТИП ПАМЯТИ. Записываемая информация устанавливается на переключателях И, К, после чего нажимается кнопка ЗАПИСЬ. При этом выполняется микропрограмма ОБСП, которая осуществляет запись информации в указанную ячейку памяти или байт ССП и останов процессора.

Для записи информации в ряд последовательно расположенных ячеек памяти, т. е. в ячейки с последовательными номерами, начальный адрес передается в регистр РАИ (фактически выполняется занесение адреса команды в текущее ССП); конечный адрес задается на переключателях А–Д. Информация, которая будет записываться во все выбираемые ячейки, устанавливается на переключателях И, К. 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 заданной информации в выбираемые ячейки памяти, начиная с адреса, переданного в регистр РАИ с набора АДРЕС КОМАНДЫ. При записи информации в ячейку с адресом (номером), установленным на переключателях А – Д, вырабатывается сигнал сравнения адресов, по которому производится останов процессора. Кнопка ЗАПИСЬ всегда подсвечивается при выполнении операции записи любого вида.

Считывание информации. Операция считывания (чтения) информации позволяет выводить в регистр индикации содержимое любой ячейки памяти, ССП, регистров центрального процессора и каналов, не имеющих постоянной индикации. При помощи переключателя ТИП ПАМЯТИ указывается один из каналов (КМ, КС1 или КС2), память (ОП, МП, ЛП или ПЗ), регистры центрального процессора или ССП. Если в регистр индикации необходимо передать содержимое регистра ЦП, регистра канала или содержимое байта ССП, то на переключателе Д указывается номер регистра или номер байта ССП. Если же в регистр индикации необхо-

димо передать содержимое ячейки памяти, то на переключателях А—Д устанавливается адрес ячейки памяти.

В случае считывания информации из регистров нажатие кнопки ЧТЕНИЕ вызывает передачу содержимого требуемого регистра на вход локальной памяти и далее в регистр индикации, который подключен к выходу локальной памяти. Для подключения регистра, номер которого задан на переключателе Д, к входу локальной памяти при соответствующем положении переключателя ТИП ПАМЯТИ формируется код, аналогичный коду управления входом ЛП. Считывание информации из ячеек памяти производится посредством микропрограммы ОБСП, причем из ОП и ЛП одновременно выдаются четыре байта. Считывание содержимого ряда последовательных ячеек памяти осуществляется аналогично случаю записи в ряд последовательных ячеек, но при нажатии кнопки ЧТЕНИЕ. При выполнении операции считывания любого вида кнопка ЧТЕНИЕ всегда подсвечивается.

Модификация таймера. Таймер, или счетчик временных интервалов, представляет собой одно четырехбайтное слово, которое располагается в ячейках 50—53 оперативной памяти и периодически изменяется, т. е. модифицируется, в соответствии с содержимым некоторого физического счетчика времени. В машине ЕС-1022 используется только интервальный таймер, который позволяет регистрировать время выполнения программы и вырабатывать сигналы для управления ходом выполнения программы по истечении заданного промежутка времени; схемы таймера непосредственно связаны с пультом управления. Модификация, или продвижение, таймера осуществляется микропрограммно; его полный цикл составляет 15,5 ч.

Для подсчета временных интервалов служит четырехразрядный двоичный счетчик, выполненный как ДПО[4—7]; на счетчик подаются сигналы счета времени с частотой напряжения сети переменного тока, т. е. 50 Гц. Если содержимое счетчика не равно нулю, то формируется запрос на обновление таймера, рассматриваемый как запрос на прерывание. В конце выполнения каждой команды производится обслуживание запросов на внешние прерывания, прерывания по вводу-выводу и на обновление таймера. Так как таймер имеет самый низкий приоритет, то запрос на обновление таймера не всегда может быть удовлетворен в течение 20 мс, составляющих период изменения содержимого ДПО[4—7]. Поэтому счетчик имеет возможность накапливать до 15 временных интервалов по 20 мс, т. е. $T_{сч} = 300$ мс. Если содержимое счетчика временных интервалов не будет учтено в течение 300 мс, то для интервального таймера оно будет потеряно.

Собственно обновление, или модификация, таймера состоит в считывании содержимого ячеек 50—53 оперативной памяти и вычитании из него содержимого ДПО[4—7], умноженного на 6 при частоте сети 50 Гц; эти действия производятся в АЛБ. Результат записывается в ячейки таймера, после чего процессор переходит к выполнению прерванной программы. Если при вычитании значение таймера становится отрицательным, то формируется запрос на внешнее прерывание по таймеру; отметим, что этот запрос не равнозначен запросу 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. Обновление таймера всегда производится при переходе процессора в состояния ожидания или останова.

Сравнение адресов. Для выполнения пультовой процедуры по сравнению адреса, установленного на наборе АДРЕС ПАМЯТИ, с адресом ячейки оперативной или постоянной памяти используется переключатель СРАВНЕНИЕ АДРЕСОВ. В зависимости от положения этого переключателя определяется реакция процессора при совпадении сравниваемых адресов.

В случае совпадения (равенства) адреса, установленного на переключателях набора АДРЕС ПАМЯТИ, с адресом ячейки оперативной памяти возможны режимы: останова, останова по записи или регенерации, перехода. При совпадении адреса ячейки оперативной памяти с адресом, установленным на переключателях А—Д, формируется сигнал равенства адресов, по которому триггер останова перебрасывается в единичное состояние, что приводит к переходу процессора в состояние останова после выполнения текущей команды. При останове подсвечиваются табло РУЧНАЯ РАБОТА и кнопка ОСТАНОВ, горит лампочка ОСТ. АДР; продолжение автоматической работы процессора вызывается нажатием кнопки ПУСК. В случае совпадения адресов при выполнении микрокоманды записи в ОП происходит останов процессора по записи (регенерации); в информационном регистре памяти индицируется записываемая информация, в адресном регистре находится ее адрес, горит лампочка ОСТ. АДР; продолжение автоматической работы процессора осуществляется при нажатии кнопки ПУСК. В режиме перехода при совпадении сравниваемых адресов сначала производится останов процессора, если он работал в автоматическом режиме, а затем выполняется занесение адреса команды в текущее ССП, т. е. адрес, установленный на переключателях Е—К, передается в регистр РАИ; процессор продолжает автоматическую работу, начиная с команды, адрес которой передан в РАИ. Если переключатель РЕЖИМ РАБОТЫ установлен в положение КОМАНДА или ОП, то происходит зацикливание операции, выполняемой с пульта управления.

В случае совпадения адреса, установленного на переключателях набора АДРЕС ПАМЯТИ, с адресом ячейки постоянной памяти возможны режимы останова, перехода и синхросигнала. При совпадении содержимого адресного регистра постоянной памяти с адресом, установленным на переключателях Б—Д, формируется сигнал, обеспечивающий останов процессора; при этом микрокоманда, которая должна выполняться при продолжении работы ЦП в любом режиме, и ее адрес индицируются на пульте управления, горит лампочка ОСТ. АДР; продолжение работы ЦП осуществляется при нажатии кнопки ПУСК. В режиме перехода при совпадении адресов информация, установленная на переключателях Ж — К, передается в адресный регистр постоянной памяти; продолжение работы осуществляется с микрокоманды, адрес которой был принят с переключателей. При формировании синхросигнала по совпадению сравниваемых адресов изменений в работе процессора не происходит; этот режим используется только при наладке и проверке работы процессора.

Занесение адреса микрокоманды. Для занесения этого адреса в регистр

адреса постоянной памяти (РАПП) переключатель РЕЖИМ РАБОТЫ устанавливается в положение МИКРОКОМАНДА, адрес микрокоманды задается на переключателях Ж – К и нажимается кнопка ЗАНЕСЕНИЕ РАПП. При этом формируется управляющий сигнал, по которому производится передача информации с выходов шифраторов переключателей Ж–К в РАПП; очевидно, что эта пультовая процедура выполняется схемным путем.

Контроль работы процессора. На пульте управления имеется переключатель КОНТРОЛЬ, положение которого задает реакцию процессора на сбои. Если этот переключатель находится в положении АВТОМАТ, то при возникновении сбоя производится переход к выполнению соответствующей обслуживающей программы. Если переключатель находится в положении ОСТАНОВ, то при возникновении сбоя производится останов процессора с индикацией ошибки в регистре ошибок процессора или на индикаторах сбоя каналов. Если же переключатель находится в положении БЛОКИРОВКА, то сбои игнорируются; при их возникновении процессор продолжает работу в заданном режиме.

Пуск и останов процессора. Состояния процессора в основном отражают состояния машины в целом. Поэтому можно рассматривать пуск и останов процессора как пуск и останов всей машины.

Для инициирования начала выполнения операций в процессоре в соответствии с режимом, заданным положением переключателя РЕЖИМ РАБОТЫ, нажимается кнопка ПУСК. По формируемому при этом сигналу триггер останова сбрасывается в нулевое состояние, и процессор переходит либо к автоматическому выполнению последовательности команд (режим «Автомат»), либо к выполнению одной инструкции (режим «Команда»), либо к выполнению одной микрокоманды, адрес которой находится в адресном регистре постоянной памяти (режим «Микрокоманда»).

Переход процессора в состояние останова осуществляется после включения питания, гашения системы, останова по адресу, по окончании выполнения команды (инструкции процессора), 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 прерывания. Процессор останавливается также тогда, когда он не способен закончить выполнение текущей команды или обслуживающей микропрограммы из-за машинного сбоя. Останов производится следующим образом. По окончании выполнения каждой команды анализируется состояние триггера останова; этот триггер устанавливается в единичное состояние сигналом, формируемым при работе процессора в режиме «Команда», при сравнении заданного адреса с адресом ячейки оперативной памяти и при нажатии кнопки ПУСК; при включении питания или выполнении микропрограммы гашения системы триггер останова устанавливается в единичное состояние микропрограммно; такое состояние триггера прерывает процесс формирования начальных управляющих сигналов, или базовых синхроимпульсов. При нажатии кнопки ПУСК после останова триггер останова перебрасывается в нулевое состояние, и процессор переходит к выполнению команды, адрес которой находится в регистре адресов команд. 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, приведенной на 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, понимают объединение узлов и блоков, непосредственно выполняющих арифметические и логические операции.

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. РКМ. При этом используются различные группы выходных вентилей РКР, отвечающие частям кода инструкции, который располагается в старших

разрядах РКР, т. е. РКР[0—15]. Действия первого этапа описываются соотношениями

$$РКМ[0-15] := РКР[0-15],$$
$$РКР[0-7] \rightarrow ДШКО,$$
$$РА1 := РКР[8-11],\ РА2 := РКР[12-15].$$

Дешифратор кода операций (ДШКО) блока центрального управления расшифровывает операционную часть инструкции и обеспечивает формирование необходимых сигналов управления; одновременно устанавливается

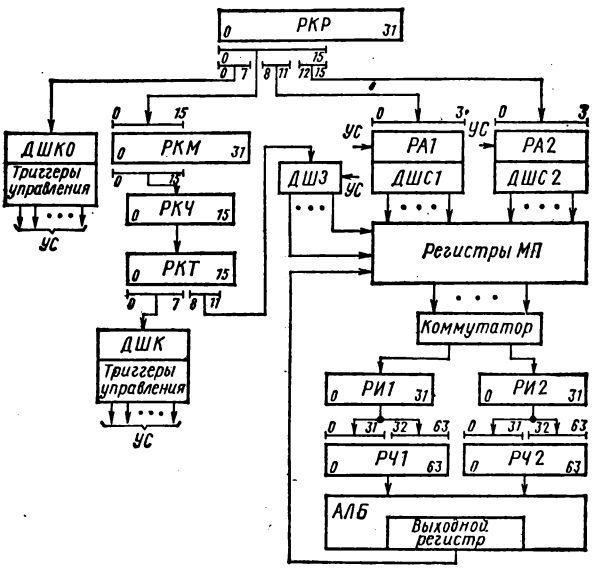


Рис. 8.1. Связи узлов и блоков процессора при выполнении инструкций формата RR

в состояние 1 триггер признака формата RR. Передача кода инструкции в РКМ необходимо 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 обозначить содержимое ячейки памяти, выбираемое по адресу, находящемуся, например, в РА1, символами ((РА1)), то действия второго этапа можно описать следующими соотношениями:

$$РКЧ := РКМ[0-15],$$
$$РИ1 := ((РА1)),\ РИ2 := ((РА2)).$$

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 учитывается 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:

$$РА1 := РА1 + 1,\ РА2 := РА2 + 1,$$
$$РИ1 := ((РА1)),\ РИ2 := ((РА2)),$$
$$РЧ1[32-63] := РИ1,\ РЧ2[32-63] := РИ2;$$

вместо первых двух соотношений, учитывая изложенное выше, можно записать

$$РА1[3] := 1,\ РА2[3] := 1.$$

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 вентилей. 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 выборки 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:

$$\begin{gathered}
РЧ1 \to АЛБ,\ РЧ2 \to АЛБ, \\
РКТ := РКЧ, \\
РКТ[0-7] \to ДШК, \\
РОП \to РВ,\ РКТ[8-11] \to ДШЗ, \\
((РКТ[8-11])) := РВ[0-31],
\end{gathered}$$

где РВ – выходной регистр арифметическо-логического блока; РОП – результат выполняемой операции.

Необходимо отметить, что после передачи информации из РЧ1 и РЧ2 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

$$((РКТ[8-11])) := РВ[0-31].$$

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 вентилей для его младших разрядов (см. рис. 7.13). Далее содержимое РВ[32–63] передается для хранения в 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

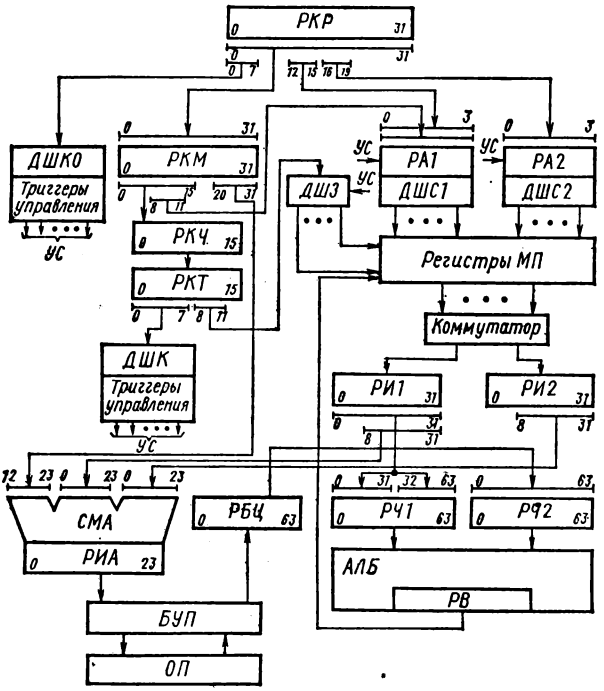


Рис. 8.2. Связи блоков и устройств при выполнении инструкций формата RX

распакованной команды. Для этого рисунка приняты обозначения, использовавшиеся в п. 7.3; 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 инструкция формата RX реализуется поэтапно с выполнением на каждом этапе вполне определенного комплекса действий. Некоторые действия подобны действиям, 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 вентилей для РКР. Инструкция формата RX является 32-разрядной, поэтому ее код во входной регистр блока центрального управления выдается из РКР через полноразрядную группу вентилей. 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

$$\text{РКМ} := \text{РКР},$$
$$\text{РКР}[0-7] \rightarrow \text{ДШКО},$$
$$\text{РА1} := \text{РКР}[12-15],\ \text{РА2} := \text{РКР}[16-19].$$

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 вентилей. 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:

$$\text{РИА} \rightarrow \text{БУП},$$
$$\text{РА1} := \text{РКМ}[8-11],$$
$$\text{РИ1} := ((\text{РА1})),$$
$$\text{РЧ1}[0-31] := \text{РИ1},\ \text{РА1} := \text{РА1} + 1,$$
$$\text{РИ1} := ((\text{РА1})),$$
$$\text{РЧ1}[32-63] := \text{РИ1},\ \text{РКЧ} := \text{РКМ}[0-15],$$
$$\text{РБЦ} := ((\text{РИА})),\ \text{РЧ2} := \text{РБЦ}.$$

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 РЧ, а содержимое РА1 увеличивается на единицу, что дает адрес младшего 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 RX аналогичны действиям четвертого этапа выполнения инструкции формата RR, поэтому они выражаются следующими соотношениями:

$$\begin{gathered}
РЧ1 \rightarrow АЛБ,\ РЧ2 \rightarrow АЛБ, \\
РКТ := РКЧ, \\
РКТ[0-7] \rightarrow ДШК, \\
РОП \rightarrow РВ,\ РКТ[8-11] \rightarrow ДШЗ, \\
((РКТ[8-11])) := РВ[0-31], \\
((РКТ[8-11]+1)) := РВ[32-63],
\end{gathered}$$

где РОП – результат выполняемой по инструкции формата RX 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 RX.

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 предварительно 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 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 РА1 местной памяти — содержимое

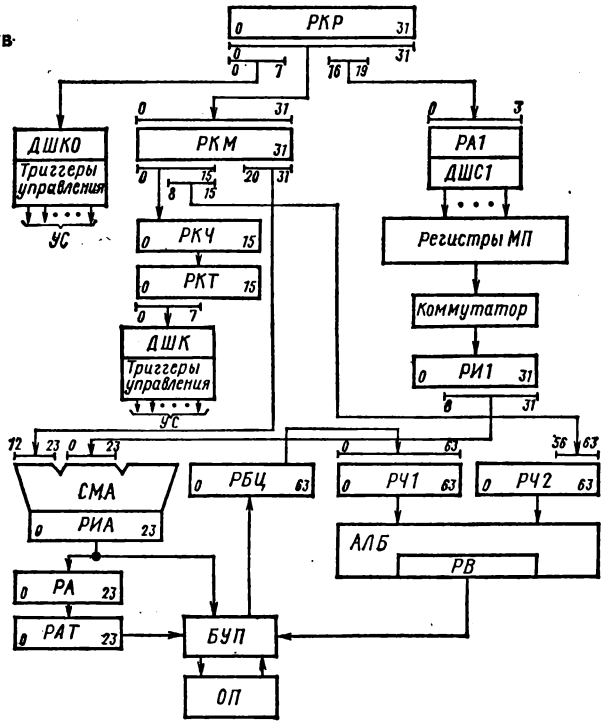


Рис. 8.3. Связи блоков и устройств при выполнении инструкций формата SI

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 адрес первого операнда, устанавливается в выходном регистре блока, т. е. в РИА. Действия второго этапа можно описать так:

$$\text{РИ1} := ((\text{РА1})),$$
$$\text{РИА} := \{\text{РИ1}\,[8-31] + \text{РКМ}\,[20-31]\}.$$

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:

$$\text{РИА} \rightarrow \text{БУП},$$
$$\text{РКЧ} := \text{РКМ}[0-15],\ \text{РА} := \text{РИА},$$
$$\text{РЧ2}[56-63] := \text{РКМ}[8-15],$$
$$\text{РБЦ} := ((\text{РИА})),$$
$$\text{РЧ1} := \text{РБЦ}.$$

Выборка первого операнда обеспечивается действиями, начинающимися 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. Так как 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. в 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:

$$\text{РЧ1} \rightarrow \text{АЛБ},\ \text{РЧ2} \rightarrow \text{АЛБ},$$
$$\text{РКТ} := \text{РКЧ},\ \text{РАТ} := \text{РА},$$
$$\text{РКТ}\,[0-7] \rightarrow \text{ДШК},$$
$$\text{РОП} \rightarrow \text{РВ},$$
$$\text{РАТ} \rightarrow \text{БУП},\ \text{РВ} \rightarrow \text{БУП},$$
$$((\text{РАТ})) := \text{РВ},$$

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 9C, 9D, 9E, 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 адрес канала не выдается в каналы для опознания, так как для выборки каналов используются индивидуальные канальные радиальные линии; в этой модели 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.

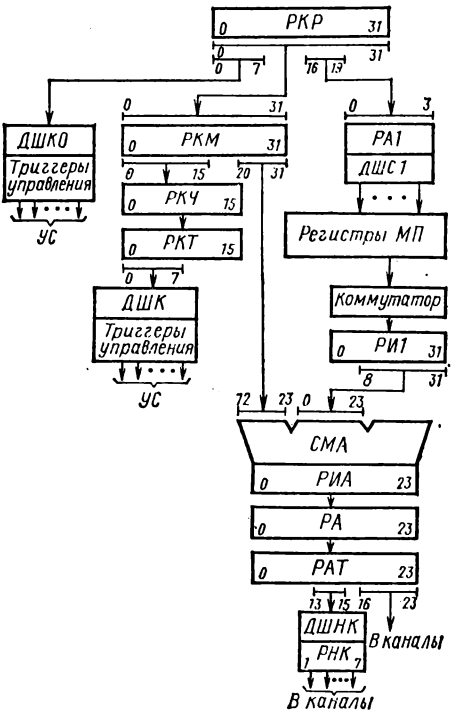


Рис. 8.4. Связи блоков и устройств при реализации инструкций ввода-вывода

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 различий, 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 (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 РЧ1 через арифметическо-логический блок код, выбранный по первому адресу. Записью кода в ОП завершается выполнение любой из рассматриваемых инструкций.

8.4. Взаимодействие устройств при микропрограммном управлении и магистральной структуре процессора

Реализация принципов микропрограммного управления обеспечивает возможность достаточно универсального формирования управляющих сигналов даже при переходе к новым алгоритмам выполнения операций

и расширение их состава. В то же время она может привести к снижению быстродействия при использовании процессоров обычной структуры, имеющих ограниченное число связей между своими функциональными частями, из-за сугубо последовательного выполнения отдельных микрокоманд. Одним из способов повышения быстродействия при микропрограммном управлении является построение процессоров по схемам, отвечающим магистральным структурам.

Сущность организации магистральных структур заключается в следующем. Все функциональные части процессора (или канала), т. е. регистры, комбинационные схемы преобразования кодов, отдельные триггеры и т. п., рассматриваются как единое поле памяти; при этом каждая функциональная часть считается адресуемой ячейкой. Через такое поле памяти проходят магистрали, к которым подключаются своими входами и выходами все адресуемые ячейки; выдача информации из ячейки в магистраль и прием информации из магистрали в ячейку производятся при действии соответствующих управляющих импульсов, формируемых, как правило, по исполняемым микрокомандам. Поскольку управление различными входами и выходами осуществляется по независимым друг от друга цепям, то всегда имеется возможность организовать связь между несколькими функциональными частями процессора по одной магистрали, причем одна из них является источником информации, а остальные — приемниками. В это же время по другой магистрали можно организовать связь между функциональными частями второго набора, причем отдельные из них могут входить в первый набор, и т. д.

При выполнении потактных действий, задаваемых исполняемыми микрокомандами, в большинстве случаев полный набор составляют три числа: два операнда и результат выполнения действия (операции). Поэтому для формирования структуры процессора считается достаточным использование трех информационных магистралей, как правило, полноразрядных. В процессе выполнения операций, задаваемых машинными командами, всегда формируются различные признаки, в том числе и о состоянии отдельных функциональных частей процессора. Для передачи соответствующих кодов, учитываемых при организации ветвлений в микропрограммах, вводится так называемая магистраль анализов.

Упрощенная структура процессора магистральной структуры при микропрограммном управлении и его основные связи с другими устройствами машины показаны на рис. 8.5; практически по такой схеме выполнен процессор машины ЕС-1033. При формировании схемы, приведенной на рис. 8.5, произведено объединение отдельных функциональных частей процессора; такие функциональные части, как блок синхронизации, блок прямого управления, блок контроля и блок диагностики, которые не играют существенной роли в рассматриваемых процессах взаимодействия устройств, на этой схеме не показаны. В качестве основных выделены: блок управления БУ с постоянной памятью ПП, блок местной памяти БМП, арифметическо-логическое устройство АЛУ, блок обработки прерываний БОПР, блок регистров управляющих БРУ, блок управления оперативной памятью БУОП, информационные магистрали М1, М2, М3, магистраль анализов МАН и блок магистрального

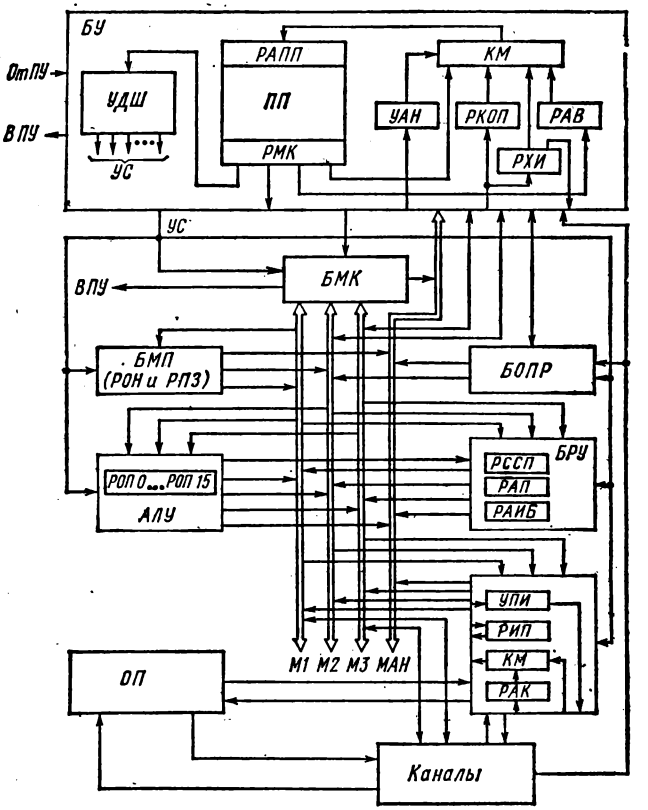


Рис. 8.5. Общая схема связей процессора магистральной структуры при микропрограммном управлении

коммутатора БМК; показаны также основные связи с оперативной памятью ОП и каналами.

Блок магистрального коммутатора. Этот блок предназначен для приема информации из других блоков процессора, хранения информации в течение заданного времени (как правило, в течение одного такта, длительность которого равна 400 нс), контроля принятой информации, выделения части кодов и выдачи информации в узлы и блоки процессора. Он образует три полноразрядные (32 + 4) информационные магистрали М1, М2 и М3. Каждый разряд любой магистрали представляет собой объединение проводным монтажом элементов соответствующего разряда нескольких регистров и комбинационных схем; это объединение подключается к разрядному элементу магистрального коммутатора.

Основу разрядной схемы магистрального коммутатора составляют триггер и ряд логических элементов *И—НЕ*. Триггер обеспечивает кратковременное хранение информации, выдаваемой на магистраль, а логические элементы позволяют, в случае необходимости, производить маскирование кодов, т. е. осуществлять запрет их выдачи. Маскирование производится в соответствии с кодами полноразрядных масок, содержащихся

в отдельных регистрах БМК; коды масок задаются частями третьего и второго слов исполняемых микрокоманд. Схемы магистрального коммутатора позволяют также производить обнуление любого количества байт по второй и третьей магистралям, что необходимо при выполнении некоторых логических операций и операций десятичной арифметики.

Блок управления. Этот блок представляет собой совокупность функциональных частей процессора, обеспечивающих реализацию принципов микропрограммного управления. Все микропрограммы хранятся в постоянной памяти, имеющей регистр адреса РАПП и выходной регистр числа, или регистр микрокоманд (РМК). Базовый адрес очередной микрокоманды задается во втором слове предыдущей микрокоманды (см. п. 7.2); для обеспечения возможных ветвлений микропрограмм четыре младших разряда базового адреса могут модифицироваться в соответствии с результатами анализов складывающихся ситуаций, проводимых в узле анализов УАН. Основная информация в УАН поступает по магистрали анализов, имеющей четыре основных разряда и формируемой аналогично информационным магистралям. Узел анализов обеспечивает, кроме того, управление работой коммутатора КМ, через который на РАПП может передаваться не только основной базовый или модифицированный адрес микрокоманды, но и информация, хранящаяся в регистрах блока управления.

Основными регистрами блока управления, участвующими в формировании адреса очередной микрокоманды, являются регистр кода операций РКОП, регистр хранения инструкции РХИ и регистр адреса возврата РАВ. Регистр кода операций служит для размещения операционной части очередной машинной команды, т. е. инструкции процессора, которая может вводиться в него с магистрали М3. При начале выполнения машинной операции код операции используется как адрес первой микрокоманды соответствующей микропрограммы. В двухбайтном регистре хранения инструкции размещается либо код команды формата RR, либо часть команды другого формата, принятого для машин Единой системы. Код операции из РХИ может выдаваться в РАПП для выборки первой микрокоманды, начинающей микропрограмму выполнения заданной операции. 11-разрядный регистр РАВ служит для хранения адреса микрокоманды, на которой произошло прерывание микропрограммы; это обеспечивает возврат, после реализации прерывания, к выполнению прерванной микропрограммы. Этот же регистр может использоваться для фиксации адреса, поступающего из РМК и используемого для организации ветвлений.

Дешифрирование основных полей микрокоманд обеспечивается схемами узла дешифраторов УДШ, включающего в свой состав и формирователи управляющих сигналов УС. Используются восемь дешифраторов на четыре входа, шесть дешифраторов на три входа и пять дешифраторов на два входа — в соответствии с принятой структурой микрокоманд. В рассматриваемом узле формируются также шесть отдельных управляющих сигналов, отвечающих одноразрядным полям микрокоманд.

Управляющие сигналы, формируемые блоком управления, рассылаются по соответствующим цепям во все функциональные части процессора;

они используются также для управления работой схем самого БУ. Исходная информация поступает в блок управления от пульта управления ПУ, из каналов, а также из других блоков процессора по магистральным цепям. Основная часть информации, используемой в БУ при функционировании процессора по автоматическому выполнению команд, поступает из магистрали анализов, к которой подключены схемы формирования различных признаков и схемы сравнения других блоков.

Блок обработки прерываний. Этот блок является одной из постоянно взаимодействующих с БУ функциональных частей процессора и предназначен для приема сигналов прерываний и формирования номера класса прерывания в виде двухразрядного двоичного кода. Блок имеет в своем составе четырехразрядный триггерный регистр, в котором фиксируются запросы на программные прерывания, прерывания по обращению к супервизору, внешние и вводо-выводные прерывания; маскирование запросов на отдельные прерывания производится микропрограммно, вне схем блока. Для выявления наличия запросов на прерывания используется отдельная схема анализа (схема анализа на нуль содержимого регистра). К выходам триггеров регистра подключена также схема приоритетов, функционирующая в соответствии с принятыми для машин Единой системы приоритетами. Выявленный запрос на прерывание, имеющий в данной ситуации наивысший приоритет, вызывает формирование в схеме кодирования соответствующего двоичного кода, выдаваемого на магистраль МАН. Возможна также выдача информации, содержащейся в основном регистре блока, на магистраль М2.

Блок местной памяти. Блок содержит 16 четырехбайтных регистров общего назначения РОН и 4 восьмибайтных регистра для размещения чисел с плавающей запятой РПЗ; кроме того, имеются схема дешифрирования адреса и схема управления обращением к РОН или РПЗ. Информация в регистры блока всегда принимается из магистрали М1; выдача информации из регистров может производиться как в магистраль М2, так и в магистраль М3.

Арифметическо-логическое устройство. Это устройство включает в свой состав как блоки для преобразования и сдвигов кодов, так и блок отдельно выделяемых операционных регистров (РОП0—РОП15). Отметим, что преобразование двоично-десятичных кодов в двоичные и двоичных в двоично-десятичные производится в АЛУ аппаратно при использовании схем блока конвертирования. Собственно операционные регистры служат для хранения данных, адресов промежуточных и окончательных результатов выполнения операций, т. е. той информации, которая используется и формируется при выполнении микрокоманд. Все шестнадцать операционных регистров являются четырехбайтными и связаны по своим входам и выходам с информационными магистралями М1, М2 и М3.

Блок регистров управляющих. Этот блок предназначен для хранения и модификации слова состояния программы, адреса команды инструкции и адреса ячейки оперативной памяти при обращении к ней со стороны процессора. Его основу составляют три регистра: РССП — регистр слова состояния программы, РАП — регистр адреса процессора и РАИБ — регистр адреса инструкции буферный. Адресная часть восьмибайтного РССП, т. е. РССП[40—63], выполнена по схеме счетчика, что позволяет

увеличивать ее содержимое на два и на четыре для формирования адреса очередной команды. По такой же схеме выполнен и трехбайтный РАП; все это позволяет модифицировать адреса одновременно с выполнением действий в АЛУ. Модификация адресной части РССП производится в соответствии с форматом выбранной из ОП очередной команды. Анализ формата команды производится при ее размещении в одном из регистров блока управления оперативной памятью. При формате RR содержимое РССП[40—63] увеличивается на два, при форматах RX, RS и SI — на четыре, при формате SS — на шесть (последовательно на два и на четыре). Все управляющие регистры связаны между собой, а также с информационными магистралями процессора.

Блок управления оперативной памятью. Этот блок представляет собой своеобразное связующее звено, обеспечивающее взаимодействие процессора и каналов с основной оперативной памятью машины. Формируя управляющие сигналы, этот блок инициирует работу ОП по записи или считыванию; из него же в ОП поступают адреса, а также числа при записи; при считывании он принимает информацию из оперативной памяти и передает ее по назначению в другие устройства. Рассматриваемый блок оперирует с запросами на обращение к ОП только двух типов: со стороны процессора и со стороны каналов, точнее общего канала; в котором имеется отдельная приоритетная схема обслуживания мультиплексного и селекторных каналов (см. ниже, п. 13.5). При одновременном поступлении запросов от процессора и канала более высокий приоритет имеет запрос канала; в остальных случаях действует правило: «первым пришел — первым и обслуживается»; если в момент поступления запроса оперативная память занята, данный запрос запоминается и ждет своей очереди на обслуживание.

В составе блока управления оперативной памятью имеется ряд функциональных узлов, обеспечивающих реализацию всех действий по обмену информацией между процессором, каналами и оперативной памятью; часть из них выделена на схеме рис. 8.5. Узел приема информации УПИ служит для первичной обработки принимаемой из ОП информации и передачи ее в регистр РИП или в каналы; первичная обработка данных заключается в усилении принимаемых кодовых сигналов и обращении кодов, т. е. преобразовании прямых кодов в обратные. Регистр информации процессора РИП предназначен для временного хранения информации, выдаваемой из ЦП в оперативную память при записи, а также информации, считываемой из ОП; он связан со всеми информационными магистралями процессора как по записи, так и по считыванию. Коммутатор КМ служит для выдачи адресов и данных в оперативную память, причем адрес может поступать как из регистра адреса канала РАК, так и из регистра адреса процессора, находящегося в блоке управляющих регистров, а данные поступают либо непосредственно из общего канала, либо из регистра информации процессора.

Важной составной частью блока управления оперативной памятью является память ключей защиты, взаимодействующая с узлом выявления ошибок УВО при обращении к ОП; практически память ключей защиты

совместно со схемами этого узла может рассматриваться как блок защиты основной оперативной памяти. Кроме ошибок, связанных с защитой памяти, в УВО выявляются случаи нарушения спецификации и неправильной адресации.

Характерной особенностью процессора магистральной структуры при микропрограммном управлении является постоянство основных связей между его функциональными частями при выполнении различных операций. При этом выполнение инструкций (команд) по преобразованию информации производится вне зависимости от используемых форматов в следующей последовательности основных действий:

1) считывание (чтение) команды из оперативной памяти;
2) подготовка адресов операндов;
3) выборка операндов;
4) обработка операндов, т. е. реализация заданной операции;
5) запись результата.

Эта последовательность действий может нарушаться при выявлении запросов на прерывания. Кроме того, при выполнении команд переходов, если не выполняются условия перехода к новой последовательности команд, после первого из указанных выше этапов производится выборка очередной команды. Запрос в оперативную память для выборки очередной команды при выполнении действий по преобразованию информации производится обычно после завершения выборки операндов. Командная информация выбирается из ОП отдельными словами; если текущая команда заканчивается на полуслове, то неиспользованная правая половина слова засылается в регистр РХИ блока управления и в последующем может использоваться как очередная команда.

Считывание команды из оперативной памяти и подготовка адресов операндов, т. е. выполнение действий первых двух из пяти этапов реализации инструкции процессора, всегда производится отдельной микропрограммой ВЫБОРКА КОМАНД. Эта микропрограмма рассчитана на реализацию действий при всех возможных ситуациях, которые могут создаваться в процессе функционирования машины. Вследствие этого предусмотрено семь входов в микропрограмму ВЫБОРКА КОМАНД, отвечающих конкретным возможным ситуациям, устанавливаемым по результатам анализа состояний отдельных функциональных частей процессора. Каждому входу отвечает определенная последовательность микрокоманд, многие из которых являются общими для различных случаев организации выборки команд и подготовки адресов операндов. Это позволяет рассмотреть взаимодействие блоков и устройств при выполнении типовой ветви указанной микропрограммы.

Организация выборки команд. Рассмотрим последовательность действий, выполняемых при реализации микропрограммы ВЫБОРКА КОМАНД, для случая, когда отсутствуют запросы на прерывания, в предыдущем цикле был произведен запрос в ОП по выборке очередной команды, которая начинается с левой половины слова, т. е. на его границе. Так как запрос о выборке очередной команды уже произведен, то выполнение микрокоманды ветви микропрограммы начинается только после удовлетворения этого запроса, т. е. после приема из ОП очередной команды. Принятая команда размещается в соответствующих ре-

гистрах процессора, после чего проводится анализ условий для выявления одного из возможных направлений выполнения последующих действий: перехода к микропрограмме ОБРАБОТКА ПРЕРЫВАНИЙ, реализации команд передач управления (переходов), обработки информации в соответствии с форматом принятой команды, предусматривающей выполнение арифметических или логических операций. Завершающие микрокоманды рассматриваемой микропрограммы всегда обеспечивают формирование адресов операндов.

В составе блока управления оперативной памятью имеется триггер очередности, нулевое состояние которого свидетельствует о том, что еще не обслужен предыдущий запрос к ОП. Все микропрограммы сформированы так, что выработке очередного запроса к ОП предшествует анализ состояния триггера очередности (ТОЧ). Если триггер очередности находится в нулевом состоянии, т. е. [ТОЧ] = 0, то выполнение находящейся в РМК микрокоманды блокируется до тех пор, пока ТОЧ не будет переброшен в единичное состояние. Содержимое РМК остается при [ТОЧ] = 0 неизменным, что обеспечивает непрерывный анализ состояния этого триггера. После установления ТОЧ в единичное состояние разрешается последовательная смена микрокоманд в РМК, изменение содержимого РАП, т. е. адреса, по которому производится обращение к ОП, и формирование очередного запроса по обращению к оперативной памяти.

При завершении обслуживания запроса по выборке очередной команды код этой команды, принимаемый из ОП, помещается в регистр информации процессора блока управления оперативной памятью, а ТОЧ устанавливается в единичное состояние. После этого начинается выполнение начальной микрокоманды рассматриваемой ветви микропрограммы ВЫБОРКА КОМАНД. По этой микрокоманде производятся следующие действия:

$$\text{РОП-1} := \text{РИП},$$
$$\text{РХИ} := \text{РИП}[0-15],$$
$$\text{РКОП} := \text{РИП}[0-7],$$
$$\text{АР БМП} := \text{РИП}[8-15],$$
$$\text{РАП} := \text{РАП} + 4$$

и снимается блокировка запросов по обращению к ОП. Таким образом командная информация подготовлена к ее непосредственной реализации, так как код операции размещен в РКОП, а составляющие адресной части команды — в адресных регистрах блока местной памяти (АР БМП), или счетчиках регистров общего назначения; в основном адресном регистре, т. е. в РАП, сформирован адрес следующей команды.

Очередная микрокоманда обеспечивает изменение содержимого адресной части регистра слова состояния программы в соответствии с результатом анализа формата принятой из ОП команды и осуществляет анализ различных условий. В зависимости от результатов анализов производится переход к первой микрокоманде соответствующей ветви микропрограммы в данной точке разветвления; для получения адреса необходимой микрокоманды по результатам анализа формируется четырехразрядный код, который замещает младшие разряды адресной части микрокоманды, при выполнении которой производится анализ.

Так, код 1111 формируется при выявлении некорректности кода операции принятой команды; при этом очередная микрокоманда устанавливает признак программного прерывания, что обеспечивает переход к микропрограмме ОБРАБОТКА ПРЕРЫВАНИЙ, и записывается код прерывания в РОП-10. Аналогичные действия выполняются при выявлении неправильной спецификации (адрес регистра с плавающей запятой задан отличным от 0, 2, 4 или 6), когда формируется код 0111; отличается только код прерывания, записываемый в РОП-10.

Коды 0101, 0100 и 0110 формируются в случае выполнения команд переходов. Они обеспечивают реализацию микрокоманд, которые во всех случаях предусматривают формирование запроса по обращению к ОП за очередной командой. Это полностью отвечает общей логике выполнения команд перехода как команд управления.

Коды 0000, 0001 и 0010, например, формируются при задании команд форматами RR, RX и RS соответственно. Если задана для исполнения команда формата RR, то общая микропрограмма ВЫБОРКА КОМАНД заканчивается микрокомандой, которая завершает подготовку к непосредственному выполнению заданной операции:

$$\text{РАПП} := \text{РКОП},$$
$$\text{РХИ} := \text{РОП-1}[16-31],$$

т. е. код операции текущей команды заносится в регистр адреса постоянной памяти как адрес первой микрокоманды по непосредственной реализации заданных действий, а в регистр хранения инструкции — неиспользованная часть командного слова, которая может быть очередной командой формата RR.

При выявлении команды формата RX сначала осуществляется увеличение содержимого РССП[40—63] на два (считается, что предыдущее изменение содержимого этой части регистра слова состояния программы также производилось на два), затем передача адреса базы в адресный регистр БМП, т. е. выполняются действия

$$\text{РССП}[40-63] := \text{РССП}[40-63] + 2,$$
$$\text{АР БМП} := \text{РОП-1}[16-19],$$

а также анализ на нуль адресов индекса и базы.

Если адреса индекса и базы равны нулю, т. е. $X2 = 0$ и $B2 = 0$, то адрес операнда практически равен смещению D; выполняется микрокоманда, по которой производятся действия

$$\text{РАПП} := \text{РКОП},$$
$$\text{РАП} := \text{РОП-1}[20-31]$$

и формируется запрос в оперативную память по выборке операнда, адрес которого передан в РАП.

Если $X2 \neq 0$, $B2 = 0$ или $X2 = 0$, $B2 \neq 0$, то выполняются микрокоманды, отличающиеся от рассмотренных выше только проведением действий по

сложению со смещением индекса или базы. Так, при $X2 \neq 0$ и $B2 = 0$ производятся следующие действия:

$$\text{РАПП} := \text{РКОП},$$
$$\text{А2} = \text{РОН}(X2) + \text{РОП-1}[20-31],$$
$$\text{А2} \rightarrow \text{РАП},$$
$$\text{А2} \rightarrow \text{РОП-15};$$

адрес второго операнда, т. е. А2, формируется в АЛУ путем сложения смещения с индексом, который располагается в одном из регистров общего назначения с номером, задаваемым $X2$. Если $X2 \neq 0$ и $B2 \neq 0$, то последовательно выполняются две микрокоманды, первая из которых обеспечивает сложение базы со смещением и засылку кода операции в РАПП, а по второй производится окончательное установление А2 путем сложения полученной суммы с индексом, засылка А2 в РАП и РОП-15, а также формирование запроса в оперативную память по выборке операнда.

Действия при выявлении команды формата RS аналогичны приведенным выше, но разветвление возможно только по двум направлениям ($B2 = 0$ и $B2 \neq 0$), так как адрес второго операнда определяется значениями смещения и базы. Близкими оказываются действия и при выявлении команд других форматов. Общей для любых возможных случаев реализации микропрограммы ВЫБОРКА КОМАНД является передача управления концевыми микрокомандами соответствующим микропрограммам выполнения заданных операций.

Выполнение арифметических и логических операций. Арифметические операции над числами с фиксированной запятой и плавающей запятой, а также многие логические операции выполняются при реализации соответствующих команд форматов RR и RX. Если командами этих форматов задаются одинаковые операции, то выполнение действий по этим операциям в основном отличается начальными шагами. Так, при выполнении операций сложения и вычитания над полноразрядными числами с фиксированной запятой микропрограммы содержат либо одну микрокоманду (формат RR), либо две микрокоманды (формат RX).

Пусть выполняется операция сложения двух полноразрядных чисел с фиксированной запятой по команде формата RR. Так как в результате реализации микропрограммы ВЫБОРКА КОМАНД адреса операндов уже занесены в счетчики РОН, то микропрограмма выполнения рассматриваемой операции состоит из одной микрокоманды, по которой производятся выборка на магистрали операндов, их сложение в арифметическо-логическом устройстве, запись результата в РОН по первому адресу исполняемой команды, формирование признака результата, формирование запроса за следующей командой и передача управления микропрограмме ВЫБОРКА КОМАНД. В символической форме эти действия записываются так:

$$\text{М2} := \text{РОН}(\text{R1}),\ \text{М3} := \text{РОН}(R2),$$
$$F = \text{А} + \text{В},\ F \rightarrow \text{М1},$$
$$\text{РОН}(\text{R1}) := \text{М1},\ \text{РССП}[34-35] := \text{ПРА},$$

где ПРА – признак результата выполнения операции, который формируется в арифметическо-логическом устройстве.

При выполнении такой же операции по команде формата RX предварительно реализуется микрокоманда, которая обеспечивает выборку из ОП второго операнда по запросу, сформированному одной из микрокоманд микропрограммы ВЫБОРКА КОМАНД, и выдачу его в магистраль М3. После передачи адреса второго операнда из РАП в оперативную память адрес очередной команды из РССП[40–63] заносится в РАП; этим обеспечивается выдача адреса очередной команды в оперативную память при реализации соответствующего запроса, формируемого заключительной микрокомандой по выполнению рассматриваемой операции.

Алгоритмы выполнения операций умножения и деления над числами с фиксированной запятой, а также алгоритмы выполнения арифметических операций над числами с плавающей запятой предусматривают проверку различных условий, связанных с размещением операндов, особенностями операндов и т. п. Так, при выполнении операций умножения и деления над числами с фиксированной запятой поле R1 должно задавать регистр общего назначения с четным номером; при выполнении операций над числами с плавающей запятой поля R1 и R2 должны задавать регистры с номерами 0, 2, 4 или 6. В связи с этим в микропрограммах выполнения операций начальные микрокоманды кроме выборки операндов обеспечивают еще проверку соответствующих условий с возможной передачей управления микропрограмме ОБРАБОТКА ПРЕРЫВАНИЙ.

Реализация команд (инструкций) переключения состояния. Посредством команд этой группы обеспечивается изменение состояний процессора за счет загрузки нового ССП, установления новой маски программы или новой маски системы, установления нового ключа памяти и некоторых других действий. Все эти команды имеют форматы SI или RR и выполняются по близким микропрограммам, что позволяет рассматривать принципы их реализации на примерах команд ЗАГРУЗИТЬ ССП и УСТАНОВИТЬ МАСКУ ПРОГРАММЫ.

Команда ЗАГРУЗИТЬ ССП имеет формат SI; в коде команды разряды, отведенные для непосредственного операнда, не используются. База и смещение задают адрес первого из двух слов, составляющих новое ССП; при загрузке ССП сохраняются значения 32-го и 33-го разрядов ранее использовавшегося ССП. После завершения микропрограммы ВЫБОРКА КОМАНД адрес старшего слова загружаемого ССП находится в РАП, а соответствующий код операции – в РАПП. Первая микрокоманда из микропрограммы рассматриваемой операции производит анализ состояния процессора, так как загрузка ССП может производиться только в состоянии СУПЕРВИЗОР; если процессор находится в состоянии ЗАДАЧА, то происходит прерывание программы.

В том случае, когда процессор находится в состоянии СУПЕРВИЗОР, т. е. РССП[15] = 0, начинаются действия по загрузке ССП. По ранее сформированному запросу в РИП из оперативной памяти принимается старшая половина ССП, выбираемая по адресу A1; содержимое РИП пересылается в регистр слова состояния программы; формируются

запрос в ОП и адрес младшей половины ССП. Действия выполняются в следующей последовательности:

$$\text{РИП} := ((\text{A1})),\ \text{M3} := \text{РИП},$$
$$\text{РОП-2} := \text{M3},\ \text{M1} := \text{РОП-2},$$
$$\text{РССП}[0-31] := \text{M1},$$
$$\text{РАП} := \text{РАП} + 4.$$

При загрузке младшей половины ССП сохраняются разряды 32 и 33 предыдущего ССП. Эти разряды выделяются с помощью маски, имеющей в двух старших разрядах двоичного кода единицы; в шестнадцатиричной системе счисления маска записывается как код C000 0000. В принимаемой из оперативной памяти младшей половине загружаемого ССП должны игнорироваться два старших разряда; для этого используется маска 3FFF FFFF. Последовательность выполняемых действий представляется так:

$$\text{РИП} := ((\text{A1} + 4)),$$
$$\text{M3} := \text{C000 0000};\ \text{M3} := \text{РССП}[32-63],$$
$$\text{M2} := \text{РИП};\ \text{M2} := \text{3FFF FFFF},$$
$$F = A \vee B,$$
$$\text{РССП}[32-63] := \text{M1}.$$

Отметим, что коды масок до выдачи в магистрали размещаются в регистрах масок блока магистрального коммутатора, а окончательное формирование младшей половины ССП производится в АЛУ за счет выполнения поразрядного логического сложения с выдачей результата в магистраль M1.

Команда УСТАНОВИТЬ МАСКУ ПРОГРАММЫ имеет формат RR; по этой команде содержимое РССП[34—39], т. е. признак результата (код условия) и маска программы в текущем ССП, заменяются содержимым 2—7-го разрядов регистра общего назначения, задаваемого полем R1 команды; поле R2 не используется. Собственно операция смены кодов выполняется при реализации двух микрокоманд. По первой из них осуществляется занесение новых признаков результата и маски программы в РОП-1, формирование маски для выделения 2—7-го разрядов слова, принятого из РОН (R1) в РОП-1, т. е. кода 3F00 0000, а также формирование запроса за следующей командой. Реализация второй микрокоманды обеспечивает выделение сохраняемых разрядов младшей половины текущего ССП посредством маски C0FF FFFF, выделение 2—7-го разрядов из слова, находящегося в РОП-1, формирование нового состава младшей половины текущего ССП с занесением его в РССП[32—63]. Действия выполняются в следующей последовательности:

$$\text{M2} := \text{РОН(R1)},\ \text{РОП-1} := \text{M2},$$
$$\text{M3} := \text{РССП}[32-63],\ \text{M3} := \text{C0FF FFFF},$$
$$\text{M2} := \text{РОП-1},\ \text{M2} := \text{3F00 0000};$$
$$F = A \vee B,$$
$$\text{РСПП}[32-63] := \text{M1}.$$

После этого управление передается микропрограмме выборки очередной команды.

Реализация команд (инструкций) ввода-вывода. Все команды (инструкции) процессора, относящиеся к вводу-выводу, являются привилегированными и имеют формат SI. Их реализация сводится к проверке некоторых условий и отсылке в каналы (общий канал) информации, необходимой для непосредственной реализации соответствующих действий. Базовой для них является микропрограмма выполнения команды НАЧАТЬ ВВОД-ВЫВОД, имеющей код операции 9С. Во всех командах ввода-вывода непосредственный операнд не используется, а номер канала и номер внешнего устройства, участвующих в выполнении заданной операции, выделяются из слова, образуемого при сложении смещения, т. е. D1, с содержимым регистра общего назначения, номер которого задается полем B1, т. е. с содержимым РОН (B1).

Как отмечалось выше, микропрограмма ВЫБОРКА КОМАНД предусматривает формирование адресов операндов. Поэтому к началу реализации собственно операции по команде НАЧАТЬ ВВОД-ВЫВОД сумма D1 и содержимого РОН(B1) уже получена и размещена в РОП-15, а код операции размещен в РКОП. В каналы необходимо отослать номер канала и номер внешнего устройства, т. е. содержимое РОП-15[22—23] и РОП-15[24—31], а также код выполняемой операции.

Первая микрокоманда из микропрограммы рассматриваемой операции производит анализ состояния процессора; если процессор находится в состоянии ЗАДАЧА, то происходит прерывание программы. Если же процессор находится в состоянии СУПЕРВИЗОР, то выполнение микропрограммы продолжается. Сначала осуществляется проверка номера канала, который в машине ЕС-1033 не может быть больше трех. Для этого из содержимого РОП-15 выделяется 21-й разряд (константа 0000 0400); если РОП-15[21] $\neq 0$, то заданный номер канала больше трех, и в РССП заносится код условия «3». Для этого выполняются следующие действия:

$$\text{М3} := 3000\,0000,\ \text{РОП-1} := \text{М3},$$
$$\text{М3} := \text{РССП}[32-63],\ \text{М3} := \text{CFFF FFFF},$$
$$\text{М2} := \text{РОП-1},\ F = A \vee B,$$
$$\text{РССП}[32-63] := \text{М1}.$$

После их выполнения управление передается микропрограмме ВЫБОРКА КОМАНД.

Если же РОП-15[21] $= 0$, то заданный номер канала является допустимым, и из содержимого РКОП и РОП-15 формируется код для передачи в каналы (в регистр буферной связи с процессором — РБСП) через магистраль М3 (см. рис. 8.5). Последовательность действий представляется так:

$$\text{М3} := \text{РКОП},\ \text{М3} := \text{FF00 0000},$$
$$\text{М2} := \text{РОП-15},\ F = A \vee B,\ \text{РОП-1} := \text{М1},$$
$$\text{М3} := \text{РОП-1},\ \text{РБСП} := \text{М3}.$$

Последующие микрокоманды организуют цикл ожидания завершения выполнения заданной операции в канале и прием признака результата из канала в РОП-1. Затем полученный из канала признак результата пересылается в регистр слова состояния программы.

Глава 9

РАСПРЕДЕЛЕНИЕ И ЗАЩИТА ПАМЯТИ

9.1. Принципы распределения памяти

Для машин Единой системы предусматривается использование принципов динамического распределения памяти, при котором выделение ячеек памяти для размещения массивов информации производится в процессе решения задач с учетом фактических длин массивов, а также порядка и времени их активизации; под активизацией понимается переход к непосредственному использованию данного массива. Распределение памяти в процессе решения задач осуществляется управляющей программой, учитывающей характеристики рабочих программ по длинам их участков и порядку активизации участков и занятость групп ячеек запоминающих устройств.

В случае динамического распределения памяти заранее неизвестно, какие ячейки ЗУ будут выделены для размещения массивов информации. Поэтому программирование может осуществляться только в условных, или логических, адресах. Переход к действительным адресам в процессе реализации программ обеспечивается специальными схемами, функционирующими в тесном взаимодействии с соответствующими программными средствами. Отметим, что динамическое распределение базируется на страничной организации памяти, при которой в оперативных и внешних запоминающих устройствах выделяются участки или зоны одинаковой емкости, называемые физическими страницами и имеющие емкость, равную целой степени числа два; емкость страницы называют ее размером.

При страничной организации вся память машины может быть сведена как бы к одному уровню, называемому математической, или виртуальной, памятью; иначе говоря, под математической памятью понимается совокупность всех ячеек оперативной и определенной части 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. Поэтому адреса ячеек в пределах одной математической страницы полностью совпадают с адресами ячеек соответствующей физической страницы или участка оперативной памяти. Такое соответствие облегчает преобразование математических адресов в физические, т. е. в номера действительных ячеек опе-

ративной 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, 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.

Таблица страниц содержит также сведения о распределении физических страниц между оперативной и внешней памятью. Если сведения указывают, 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.

Для случая страничной организации памяти преобразование математического адреса в физический сводится 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 таблица страниц обычно размещается в специальной регистровой памяти малой емкости. С целью сокращения общего аппаратурного состава в регистровой памяти, как правило, хранится только часть таблицы страниц, отвечающая наиболее часто используемым на данном временном интервале участкам оперативной памяти.

В машинах Единой системы предусматривается также сегментация математической памяти, заключающаяся в разделении всех математических страниц на группы – сегменты с независимой адресацией слов внутри каждой группы. В границах одного сегмента математические страницы могут относиться как к оперативной, так и к внешней памяти. Адрес любого слова теперь составляется из трех частей: номера сегмента, номера страницы и адреса слова внутри страницы; а для преобразования математического адреса в физический необходимо еще иметь таблицу сегментов, определяющую соответствие между номерами сегментов и таблицами страниц.

Рассмотрим принципы распределения памяти при ее страничной организации, когда каждая страница содержит 2048 байт. Полный 24-разрядный адрес используется для адресации ячеек математической памяти так: 4 разряда определяют номер сегмента, 9 разрядов – номер страницы и 11 разрядов – номер ячейки внутри страницы. Таким образом совокупная математическая память разбита на 16 сегментов, а каждый сегмент содержит 512 страниц. При емкости оперативной памяти, равной 512К байт, выделяется 256 физических страниц, располагаемых в ОП; очевидно, что

номер физической страницы ОП является восьмиразрядным; разрядность частей адреса учитывается при построении соответствующих схем.

При решении любой задачи используется таблица сегментов и для каждого сегмента – таблица страниц; это необходимо для преобразования математического адреса в физический. Если оказывается, что необходимая страница находится во внешней памяти, то решение данной задачи прерывается, а управляющая программа организует вызов нужной страницы в ОП. После этого задача ставится в очередь на обслуживание процессором. Таблица сегментов и большие части таблиц страниц размещаются в оперативной памяти, что связано с ограниченной емкостью регистровой памяти, используемой при преобразовании математических адресов в физические. Для хранения строк таблиц страниц, отвечающих наиболее часто используемым физическим страницам, обычно используется ассоциативная память [28], содержащая ряд основных регистров и соответствующее количество схем сравнения.

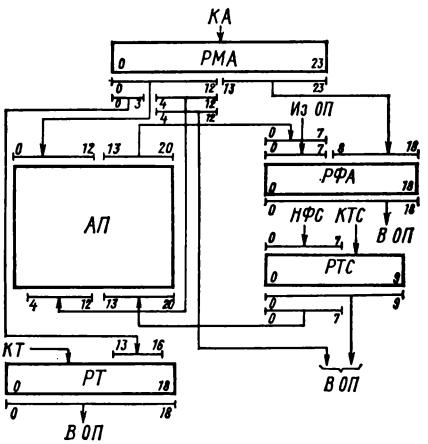


Рис. 9.1. Схема преобразования адресов

Схема преобразования адресов приведена на рис. 9.1, где выделены ассоциативная память (АП), показанная в виде одного блока, и некоторые регистры с их основными связями. Для этой схемы приняты следующие обозначения: РМА – регистр математического адреса, КА – код адреса, РФА – регистр физического адреса, РТ – табличный регистр сегментов, РТС – табличный регистр страниц, КТ – код таблицы сегментов, КТС – код таблицы страниц, НФС – номер физической страницы.

Каждый основной регистр АП хранит одну строку таблицы страниц; разряды [0 – 12] содержат номер сегмента и номер математической страницы, разряды [13 – 20] – номер физической страницы. Разряд [21] каждого регистра служит указателем использования соответствующей строки таблицы страниц; при обращении к данной строке, т. е. некоторой физической странице, в разряд [21] заносится единица. Разряд [22] является указателем вида физической памяти; если в нем 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 [13 – 23] непосредственно передается в РФА [8 – 18]. Номер же физической страницы в РФА [0 – 7] 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—12] с содержимым старших 13 разрядов всех основных регистров АП; сравнение производится параллельно с помощью имеющихся в составе АП схем сравнения. Если содержимое 22-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 [13—20] соответствующего регистра АП. Выявленный номер физической страницы передается в РФА [0—7], а в 21-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 — 12] не совпадает с содержимым старших разрядов ни одного из регистров АП, то это указывает на необходимость поиска нужной строки таблицы страниц и ее размещения в АП. Для этого используются таблица сегментов и таблица страниц, размещенные в оперативной памяти. По таблице сегментов определяется адрес начальной строки нужной таблицы страниц, а затем выявляется требуемая строка этой таблицы.

С началом реализации программы решения некоторой задачи, т. е. при ее активизации, управляющая программа вводит в табличный регистр РТ код таблицы сегментов, 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.

Сформированный в РТ адрес направляется в оперативную память для выборки соответствующей строки таблицы сегментов, представляющей собой полное слово. Младшие разряды этого слова дают код таблицы страниц, определяющий адрес первой строки соответствующей таблицы страниц; 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—12]. Полученный 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; на место размещения указывает значение соответствующего разряда. Если искомая страница находится в ОП, то ее номер вводится в РТС [0—7] и в РФА [0—7]. 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 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 21-м разряде. Считается, что физическая страница, определяемая строкой таблицы, находящейся в этом регистре, используется реже других (отметим, что при равенстве единице содержимого 21-го разряда всех регистров АП производится сброс этого разряда регистров в нулевое состояние). Собственно запись новой строки таблицы страниц в АП производится за счет передачи в разряды [13 – 20] выбранного регистра содержимого РТС [0 – 7], а в разряды [4 – 12] этого регистра – содержимого РМА [4 – 12]; если необходимо, то в разряды [0 – 3] вводится код номера сегмента.

Если выбранная при втором обращении к оперативной памяти строка таблицы страниц свидетельствует о том, что нужная страница находится во внешней памяти, то управляющая программа производит перераспределение за счет обмена страницами между ОП и внешней памятью. При этом корректируются таблицы страниц, 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

Любой блок защиты предназначается для предотвращения искажения информации, хранящейся в основной оперативной памяти, вследствие возможных ошибочных обращений к ОП по записи и ее неправильного использования вследствие возможных ошибочных обращений к ОП по считыванию. Во всех моделях ЕС ЭВМ блоки защиты строятся в соответствии с принципами защиты оперативной памяти по ключам. Каждой рабочей программе и отводимым для нее массивам ячеек (страницам) основной оперативной памяти придается специальный код, именуемый ключом. Обращение к памяти разрешается только в том случае, когда ключ программы, содержащийся в ССП, совпадает с ключом памяти (ключом защиты памяти), т. е. с ключом, приданным массиву ячеек (странице), к которому осуществляется обращение по записи или считыванию. При несовпадении ключей обычно формируется сигнал о нарушении защиты, рассматриваемый как сигнал запроса на программное прерывание. Условно считается, что нулевой ключ программы сравним со всеми ключами памяти. Отметим, что при обращении к ОП со стороны канала вместо ключа программы используется ключ, содержащийся в адресном слове канала.

Код ключа памяти кроме разрядов, содержащих собственно ключ памяти, имеет несколько дополнительных разрядов. В простейшем случае разряды [0 – 3] представляют собственно ключ памяти, разряд [4] ука-

зывает на режим защиты; используется еще и контрольный разряд. Если в разряде 4 содержится единица, то защита осуществляется при всех обращениях к памяти, т. е. как по записи, так и по считыванию. Если же в этом разряде содержится нуль, то защита осуществляется при обращениях к ОП по записи; иначе говоря, при нулевом значении рассматриваемого разряда обращение к памяти по считыванию разрешается и в случае несовпадения ключа программы с ключом памяти. В то же время принципы защиты памяти не зависят от того, находится ли процессор в состоянии ЗАДАЧА или в состоянии СУПЕРВИЗОР.

Общая структура блоков защиты. Для всех ЭВМ Единой системы структура блоков защиты одинакова, хотя функциональные части блоков могут относиться к различным частям процессора или оперативной памяти. Общность принципов защиты памяти позволяет выявить сущность построения соответствующих схем на примере блока защиты, принятого для машины ЕС-1022. Этот блок в соответствии с общими принципами защиты памяти обеспечивает защиту основной оперативной памяти от недопустимых обращений не только со стороны рабочих или обслуживающих программ, т. е. со стороны процессора, но и со стороны каналов; каждому каналу всегда придается так называемый ключ канала.

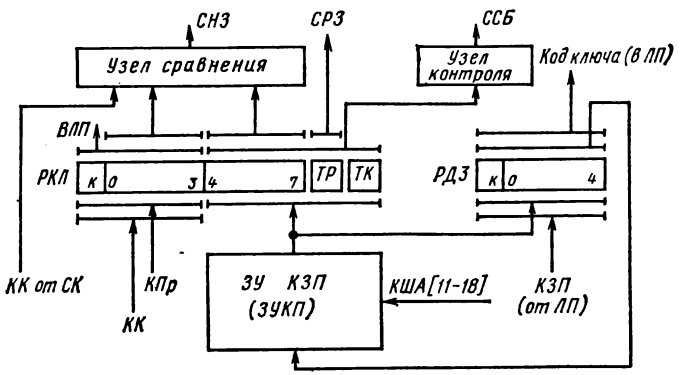


Рис. 9.2. Структура блока защиты

Состав блока защиты и основные связи между его функциональными частями показаны на рис. 9.2. В блок защиты БЗ входят: запоминающее устройство ключей защиты памяти ЗУ КЗП, или запоминающее устройство просто ключей памяти ЗУКП, узел сравнения, узел контроля, регистр ключей РКЛ, с которым по группам входных и выходных вентилей объединены триггер режимов ТР и триггер контроля ТК; используется также третий байт регистра данных оперативной памяти, т. е. РДЗ. Управление всеми частями блока защиты — микропрограммное. Основной информацией, вводимой в блок защиты, являются коды ключей. Собственно ключ имеет четыре разряда, кроме того используется контрольный разряд К; в коде ключа памяти имеется также разряд, определяющий режим защиты.

Шестиразрядные ключи защиты памяти, хранимые в ЗУ КЗП, отвечают всем страницам оперативной памяти, каждая из которых содержит 2048 байт. Загрузка ключей в ЗУ КЗП производится после каждого распределения или перераспределения памяти в интересах рабочих программ. Емкость ЗУ КЗП может составлять 64, 128 и 256 слов, что определяется емкостью используемой в составе машины оперативной памяти. Номер страницы ОП определяют старшие разряды адреса; они же используются для выборки соответствующих ключей из ЗУ КЗП, поступая в него по кодовым шинам адреса КША [11 – 18]. В качестве информационного регистра, или регистра числа, для запоминающего устройства ключей защиты памяти используется РД3. В этот регистр из локальной памяти (ЛП) могут вводиться ключи защиты памяти (КЗП) для последующей записи в ячейки ЗУ КЗП по адресам, определяемым кодами на КША [11 – 18]. Считанные из ЗУ КЗП коды ключей через РД3 могут выдаваться в локальную память.

Регистр ключей РКЛ. Этот регистр представляет собой первый байт регистра прерываний (см. п. 7.2); его функциональным дополнением и расширением служат триггеры ТР и ТК. Этот запоминающий узел предназначен для приема, хранения и выдачи кодов ключей. В РКЛ [4 – 7], ТР и ТК хранится код ключа защиты памяти, принимаемый из ЗУ КЗП. Весь этот код может выдаваться в узел контроля; по состоянию триггера ТР формируется сигнал режима защиты (СРЗ), используемый в устройстве управления машины. Собственно ключ памяти, размещаемый в РКЛ [4 – 7], выдается в узел сравнения.

В РКЛ [К, 0 – 3] обычно хранится ключ программы КПр, реализуемой на данном временном интервале. Ввод ключа программы в РКЛ [К, 0 – 3] производится всегда при смене слова состояния программы по команде ЗАГРУЗИТЬ ССП. Собственно ключ программы из РКЛ [0 – 3] выдается в узел сравнения для установления его идентичности ключу памяти (ключу защиты памяти). При микропрограммной приостановке реализации рабочей программы вследствие перехода к обслуживанию мультиплексного канала код ключа программы передается из РКЛ [К, 0 – 3] для временного хранения в локальную память. На его место заносится ключ канала КК, содержащийся в управляющем слове устройства, которое через мультиплексный канал будет обмениваться информацией с оперативной памятью. После возврата к реализации рабочей программы в РКЛ [К, 0 – 3] вновь вводится ключ программы.

При микропрограммной приостановке реализации рабочей программы вследствие перехода к обслуживанию селекторного канала смена кодов в РКЛ [К, 0 – 3] не производится. Ключ канала подается в узел сравнения непосредственно из регистра ключа канала РКК, расположенного в приступающем к обмену информацией с ОП селекторном канале СК.

Узел контроля. Этот узел служит для контроля правильности кода, принятого из запоминающего устройства ключей защиты памяти в РКЛ [4 – 7], ТР и ТК. Его основу составляют схемы сложения по mod 2. В том случае, когда принятый из ЗУ КЗП код ключа содержит четное число единиц, формируется сигнал сбоя (ССБ), который может вызвать соответствующее прерывание работы машины.

Узел сравнения. Узел предназначен для формирования сигнала нарушения защиты по результатам анализа ключей защиты памяти и ключей программ или ключей каналов. Код ключа программы и код ключа устройства, взаимодействующего с ОП через мультиплексный канал, передаются в узел сравнения из РКЛ [0 – 3], код ключа защиты памяти подается из РКЛ [4 – 7], а код ключа селекторного канала – из регистра ключа канала РКК. Сравнение ключа защиты памяти с ключом селекторного канала производится при действии управляющего импульса $УИ_к$; в противном случае ключ защиты памяти сравнивается с содержимым РКЛ [0 – 3].

Сигнал нарушения защиты вырабатывается в виде сигнала высокого уровня, если не выполняется хотя бы одно из четырех равенств для двух случаев сравнения ключей:

1) РКК[0] = РКЛ[4],
   РКК[1] = РКЛ[5],
   РКК[2] = РКЛ[6],
   РКК[3] = РКЛ[7],

2) РКЛ[0] = РКЛ[4],
   РКЛ[1] = РКЛ[5],
   РКЛ[2] = РКЛ[6],
   РКЛ[3] = РКЛ[7],

где РКК[i] – значение i-го разряда кода ключа, содержащегося в регистре ключа канала ($i = 0, 1, 2, 3$); РКЛ[j] – значение соответствующего разряда ключа защиты памяти, содержащегося в РКЛ [4 – 7].

В соответствии с принципами формирования сигнала нарушения защиты можно записать следующее логическое соотношение:

$$СНЗ = (x_0 \oplus y_0) \vee (x_1 \oplus y_1) \vee (x_2 \oplus y_2) \vee (x_3 \oplus y_3), \tag{9.1}$$

где

$$x_i = x_i^1 УИ_к \vee x_i^2 \overline{УИ_к}; \tag{9.2}$$

$$x_i^1 = РКК[i],\ i = 0, 1, 2, 3;$$

$$x_i^2 = РКЛ[i],\ i = 0, 1, 2, 3;$$

$$y_i = РКЛ[j],\ i = 0, 1, 2, 3;\ j = 4, 5, 6, 7.$$

Если используются элементы *И – НЕ* и *И – ИЛИ – НЕ*, то соотношения (9.1) и (9.2) преобразуются путем использования выражения для представления суммы по mod2 через элементарные логические операции, а также приема двойного инверсирования. Тогда

$$СНЗ = \overline{z_0 z_1 z_2 z_3}, \tag{9.3}$$

где

$$z_i = \overline{x_i \bar{y}_i \vee \bar{x}_i y_i},\ i = 0, 1, 2, 3.$$

Очевидно, что для реализации соотношений (9.3) необходимо использовать четыре 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):

$$x_i = \overline{\bar{x}_i^1 УИ_к \vee \bar{x}_i^2 \overline{УИ_к}}. \tag{9.4}$$

Инверсируя обе части (9.2), получим

$$\bar{x}_i = \overline{x_i^1 \text{УИ}_\text{к} \vee x_i^2 \overline{\text{УИ}_\text{к}}}. \qquad (9.5)$$

В реальной схеме необходимо реализовать четыре соотношения (9.4) и четыре соотношения (9.5), поэтому используются восемь элементов *И — ИЛИ — НЕ* с двумя двухвходовыми элементами *И* каждый. Для получения парафазных кодов по кодам, поступающим из РКК, используются четыре инвертора; два инвертора необходимы для инверсирования сигнала $\text{УИ}_\text{к}$.

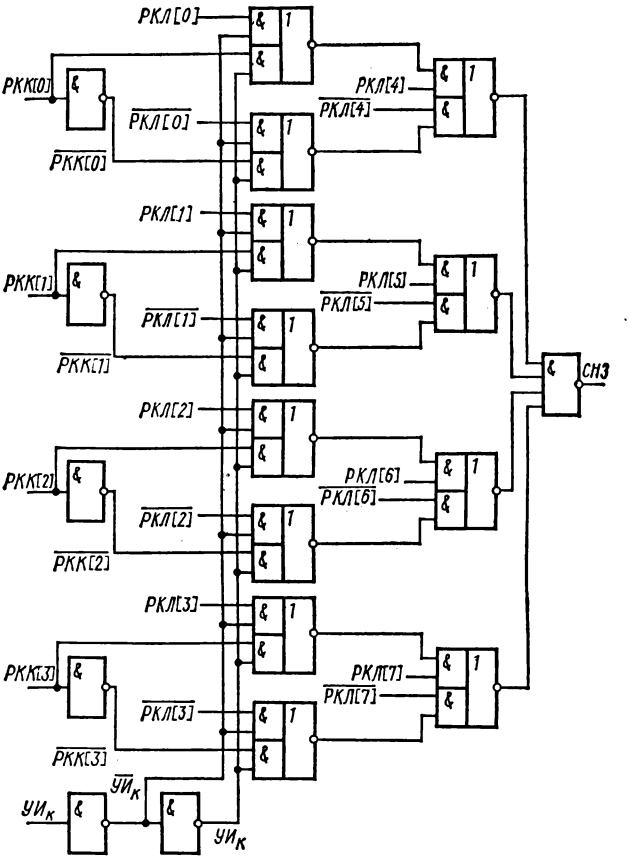


Рис. 9.3. Схема сравнения ключей

Схема сравнения ключей (рис. 9.3) полностью отвечает приведенным выше соотношениям, обеспечивая на своем выходе формирование сигнала нарушения защиты в виде сигнала высокого уровня. Если нарушения защиты памяти нет, то на выходе схемы сравнения имеет место сигнал низкого уровня. Если хотя бы один из ключей имеет нулевое значение, 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.

Работа блока защиты. Рассмотренный блок защиты может работать в трех режимах:

автоматическом, который включается при каждом обращении к основной оперативной памяти;

режиме считывания информации из ЗУ КЗП;

режиме записи информации в ЗУ КЗП.

При работе блока защиты в его ЗУ КЗП всегда передаются старшие разряды 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, т. е. кода ключа, из ЗУ КЗП и размещение его в РКЛ [4 – 7], а также триггерах ТР и ТК;

3) формирование сигнала режима защиты и контроль кода ключа в узле контроля;

4) сравнение кода ключа защиты памяти, считанного из накопителя ЗУ КЗП с ключом программы, предварительно засланным в РКЛ [0 – 3], или ключом канала; при совпадении ключей на выходной шине узла сравнения (шине СНЗ) сохраняется низкий уровень сигнала; при равенстве нулю ключа программы или ключа канала формируется сигнал блокировки;

5) возвращение в исходное состояние схем узлов блока защиты.

В режиме считывания информации из ЗУ КЗП блок защиты работает при выполнении команды ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ. Все действия производятся, как и в автоматическом режиме, но считанный из ЗУ КЗП ключ памяти передается в РД3 и далее в локальную память; 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 тем, что производится только запись информации, находящейся в РД3, по адресу, вводимому в ЗУ КЗП по кодовым шинам адреса. Этот режим используется при выполнении команды УСТАНОВИТЬ КЛЮЧ ПАМЯТИ.

9.3. Память ключей защиты

Включение в состав процессоров блоков защиты, работающих в каждом цикле обращения к основной оперативной памяти, не должно практически 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 запоминающих устройств с малым циклом обращения, работающих, к тому же, параллельно основ-

ной ОП; время обращения к таким ЗУ должно быть в несколько раз меньше времени обращения к основной оперативной памяти. В связи с этим в качестве памяти ключей защиты используются запоминающие устройства с накопителями из триггерных матриц в интегральном исполнении (ЕС-1022, ЕС-1033) и на тонких магнитных пленках (ЕС-1030, ЕС-1050, ЕС-1052).

ЗУ ключей защиты с накопителями из триггерных матриц. Запоминающие устройства ключей защиты памяти, используемые в машинах ЕС-1022 и ЕС-1033, близки по основным принципам построения, хотя и отличаются возможностями применяемых микросхем. Поэтому в качестве базового рассматривается ЗУ КЗП применительно к машине ЕС-1022,

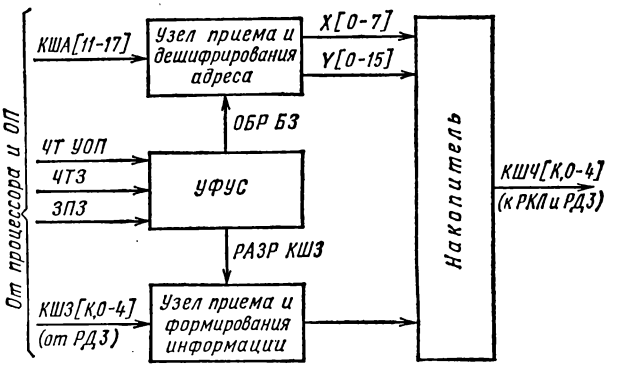


Рис. 9.4. Схема запоминающего устройства ключей защиты памяти с накопителем из триггерных матриц

имеющей оперативную память емкостью 256К байт; при этом емкость запоминающего устройства ключей защиты составляет 128 шестиразрядных слов, что отвечает выделению в ОП страниц, содержащих по 2048 байт информации. Регистры, служащие для размещения кодов, записываемых в ЗУ КЗП или считываемых из него, отнесены к процессору (см. рис. 9.2); это учтено при формировании схемы рассматриваемого запоминающего устройства, приведенной на рис. 9.4. Основу устройства составляют накопитель, узел приема и дешифрирования адреса, поступающего по КША [11 – 17], узел приема и формирования информации, поступающей для записи в ЗУ КЗП по КШЗ [К, 0 – 4], а также узел формирования управляющих сигналов (УФУС) обращения к блоку защиты (ОБР БЗ) и разрешения записи (РАЗР КШЗ) по сигналам, поступающим от процессора и ОП. Сигнал ЧТ УОП поступает из оперативной памяти при обращении к ней по записи или считыванию; сигналы ЧТЗ и ЗПЗ, поступающие из процессора, отвечают соответственно случаям считывания и записи информации. Время считывания информации из ЗУ КЗП данного типа составляет 125 нс при полном цикле работы, равном 275 нс.

Накопитель выполнен на микросхемах 155РУ1, которые функционально представляют собой триггерные матрицы с усилителями записи и считывания. Каждая микросхема, выполненная в стандартном 14-контактном корпусе, содержит 16 триггеров, усилители записи нуля и единицы (УЗП«0»

и УЗП«1») и усилители считывания нуля и единицы (УЧТ«0» и УЧТ«1»). Общий состав микросхемы 155РУ1 показан на рис. 9.5, а принципы записи и считывания информации поясняются с помощью схемы рис. 9.6. Очевидно, что каждая микросхема представляет собой запоминающее устройство емкостью 16 однобитных (одноразрядных) слов. Выборка ячеек этого ЗУ — двухкоординатная; за один цикл обращения всегда выбирается только одна одноразрядная ячейка, т. е. один триггер.

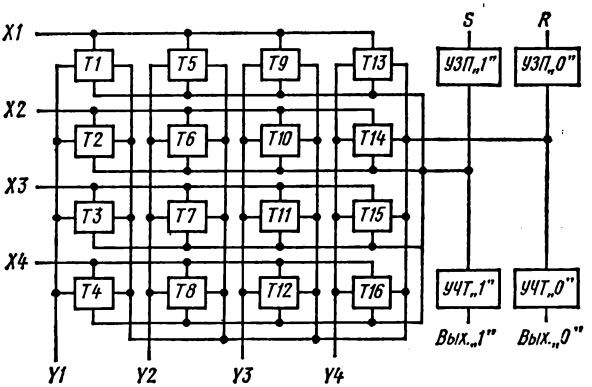


Рис. 9.5. Общая схема триггерной матрицы

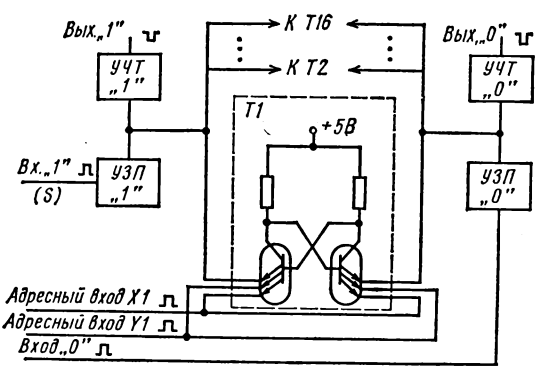


Рис. 9.6. Организация записи и считывания в ячейке Т1

Для записи информации в выбранную ячейку по соответствующим адресным шинам X и Y подаются сигналы высокого уровня; одновременно подается сигнал высокого уровня на вход УЗП«0» (при записи нуля) или на вход УЗП«1» (при записи единицы); эти входы обозначают буквами R и S соответственно. В режиме хранения информации в адресные шины и входы усилителей записи постоянно подается сигнал низкого уровня.

Считывание записанной в ячейку памяти информации происходит при подаче в адресные шины сигналов высокого уровня с одновременной подачей сигналов низкого уровня на входы усилителей записи; считывание информации всегда производится без ее разрушения. Так как запись и считывание информации осуществляются при использовании одной и

той же пары разрядных шин, то циклы записи и считывания в рассматриваемом ЗУ совместить нельзя. Отметим, что выходной транзистор усилителя считывания выполнен с открытым коллектором, что обеспечивает объединение по схеме *ИЛИ* при формировании общей разрядной шины.

Весь накопитель функционально можно разделить на одноразрядные запоминающие массивы, включающие в свой состав по восемь микросхем 155РУ1, если общая емкость ЗУ КЗП равна 128 словам. Принципы построения такого массива поясняются схемой, приведенной на рис. 9.7.

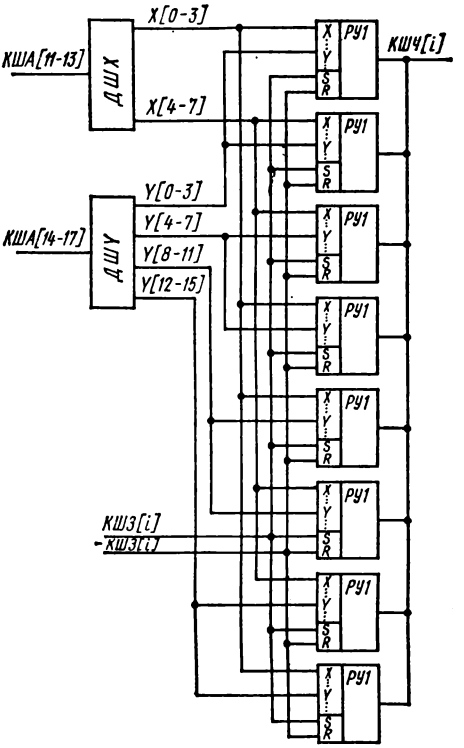


Рис. 9.7. Организация разрядного массива накопителя

Входы *X* и *Y* микросхем параллельно подключены к выходам дешифраторов ДШ*X* и ДШ*Y*; при этом каждый выход ДШ*X* подключен к входам четырех микросхем, а каждый выход ДШ*Y* — к входам двух микросхем, что обеспечивает при любом коде адреса выборку только одной ячейки из 128, составляющих данный массив.

Конструктивно накопитель выполнен из так называемых модулей памяти. Каждый модуль обеспечивает хранение 64 четырехразрядных слов и включает в свой состав 16 микросхем 155РУ1; кроме того, на плате соответствующего типового элемента замены размещаются микросхемы, выполняющие функции усиления и инверсирования выходных сигналов дешифраторов. При рассматриваемой емкости ЗУ КЗП его накопитель состоит из четырех модулей памяти, причем возможности двух из них при 6-разрядных ключах используются не полностью, так как четыре модуля памяти способны хранить значения 128 восьмиразрядных слов.

Узел приема и дешифрирования адреса служит для приема кода адреса, поступающего по КША[11—17], преобразования его в парафазный код и дешифрирования для возбуждения одной из шин системы *X* и одной из шин системы *Y*. Он содержит входные схемы и два дешифратора. Дешифратор ДШ*X* имеет три входа и восемь выходов; дешифратор ДШ*Y* — четыре входа и 16 выходов. Сигналы выборки, формируемые на выходах дешифраторов, имеют низкий уровень, поэтому и предусмотре-

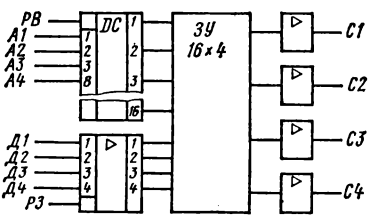
но их инверсирование в модулях памяти до подачи на адресные шины микросхем 155РУ1.

Узел приема и формирования информации служит для приема кодов из КШЗ[К, 0—4] и преобразования их в парафазные. Он содержит схемы, аналогичные входным схемам узла приема и дешифрирования адреса. Парафазный код с выходов узла подается на информационные входы запоминающих элементов; одноразрядному парафазному коду отвечают шины КШЗ[*i*] и — КШЗ[*i*] на рис. 9.7.

Узел формирования управляющих сигналов обеспечивает управление работой ЗУ КЗП в трех возможных режимах: автоматическом, записи и считывания, которым отвечают соответственно внешние сигналы ЧТ УОП, ЗПЗ и ЧТЗ. В автоматическом режиме считывание информации из ЗУ КЗП производится при каждом обращении к оперативной памяти; выборка информации производится по адресу, получаемому из КША[11—17]; сигнал разрешения обращения ОБР БЗ формируется в УФУС по внешнему сигналу ЧТ УОП. В режиме считывания производятся те же действия, но сигнал ОБР БЗ формируется по внешнему сигналу ЧТЗ. При работе в режиме записи сигнал ОБР БЗ формируется по внешнему сигналу ЗПЗ; одновременно формируется сигнал РАЗР КШЗ, что обеспечивает запись в выбранную ячейку накопителя ЗУ КЗП кода нового ключа защиты памяти.

Накопитель запоминающего устройства ключей памяти машины ЕС-1033 собран на микросхемах 155РУ2, имеющих более высокую степень интеграции по сравнению с микросхемами 155РУ1 и выполненных в стандартных 16-контактных корпусах. Каждая такая микросхема представляет собой оперативное запоминающее устройство емкостью 16 четырехразрядных слов. Кроме собственно запоминающих элементов, микросхема содержит дешифратор на четыре входа и 16 выходов, группу входных усилителей, объединенных по входу стробирования, и четыре выходных усилителя, как это показано на схеме рис. 9.8.

Рис. 9.8. Функциональный состав микросхемы 155РУ2



Дешифратор служит для выборки одной из 16 запоминающих ячеек по четырехразрядному адресу, подаваемому на входы А1 — А4; с выходами дешифратора связаны соответствующие адресные шины запоминающих ячеек. По входу стробирования, или входу разрешения выборки РВ, реализуется функция запрета: при действии на этом входе сигнала высокого уровня не может быть выбрана ни одна из 16 ячеек, т. е. не могут быть произведены ни запись, ни считывание. Записываемая информация подается на входы Д1 — Д4, которым отвечает группа стробируемых входных усилителей; запись может быть произведена только при подаче

на вход разрешения записи РЗ сигнала низкого уровня. Считывание информации производится с помощью выходных усилителей, по одному на каждый разряд (выходы С1 – С4). Отметим, что стробирование информации на входах дешифратора и усилителей записи производится за счет использования разрядных вентильных схем.

Каждая запоминающая ячейка состоит из четырех RS-триггеров, реализованных на двух двухэмиттерных транзисторах, связи которых показаны на рис. 9.9. В режиме хранения в адресной шине Ш действует

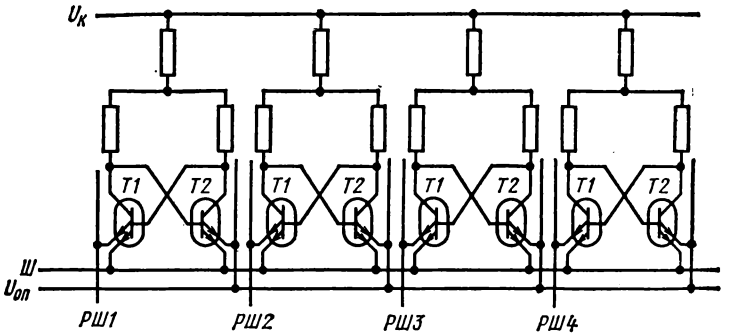


Рис. 9.9. Построение 4-разрядной ячейки памяти

сигнал низкого уровня, а при записи или считывании – сигнал высокого уровня. Считывание или запись осуществляются по общей для каждого разряда разрядной шине (РШ1, РШ2, РШ3 или РШ4), к которой подключается один из эмиттеров транзистора Т1 каждого из 16 триггеров данного разряда. Один из эмиттеров транзистора Т2 подключен к источнику опорного напряжения $U_{оп}$, который является общим для всего массива ячеек памяти, как и источник коллекторного питания $U_к$. Изменение потенциала на разрядной шине вызывает переключение тока элемента памяти либо в разрядную шину (случай считывания), либо в цепь источника опорного напряжения (случай записи). Усилители считывания выполнены по выходам с открытым коллектором, что позволяет объединять выходы микросхем в общие внешние разрядные шины.

Конструктивно ЗУ КЗП машины ЕС-1033 выполнено в виде двух модулей памяти, или двух типовых элементов замены. Каждый модуль обеспечивает хранение 256 четырехразрядных слов; очевидно, что два разряда второго модуля памяти могут считаться резервными, так как коды ключей памяти являются шестиразрядными. Общая организация одного модуля памяти показана на рис. 9.10. Четыре разряда адреса по кодовым шинам КША подаются на входы А всех шестнадцати микросхем типа РУ2. Выборка производится всегда только в одной микросхеме, определяемой выходом дешифратора адреса ДША, на котором формируется сигнал низкого уровня в соответствии со значением четырех старших разрядов адреса; дешифратор адреса выполнен на микросхеме 155ИД3. При записи коды ключей по кодовым шинам записи КШЗ подаются

параллельно на входы Д микросхем типа РУ2, а управляющий сигнал записи ЗП – на их входы разрешения записи (РЗ). Выдача считываемых кодов производится по КШЧ[К, 0–4] в узел сравнения кодов ключей, а также в регистр информации процессора.

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

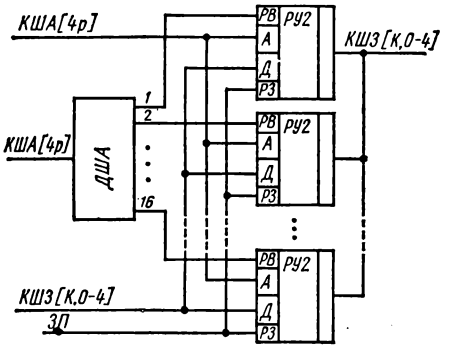


Рис. 9.10. Общая организация модуля памяти

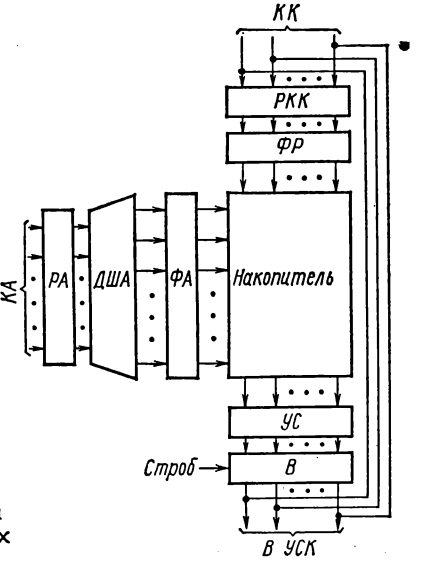


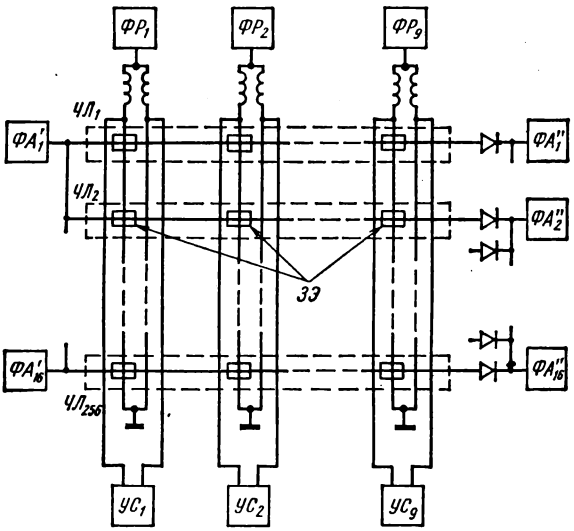
Рис. 9.11. Общая схема запоминающего устройства ключей защиты памяти с накопителем на тонких магнитных пленках

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 2*D*. 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 2*D*. Поэтому общая схема рассматриваемого устройства, приведенная на рис. 9.11, 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,

Рис. 9.12. Схема накопителя



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 вентилей, УСК – узел сравнения ключей. Цепь обратной связи по передаче кодов с выходов вентилей в РКК обеспечивает регенерацию считываемой информации.

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 [28]. 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. 9.12.

С целью уменьшения количества используемых адресных формирователей и упрощения дешифратора адреса в рассматриваемом ЗУ реализован принцип двустороннего дешифрирования с разбиением 8-разрядного адреса на две 4-разрядные части. Соответственно используются два дешифратора на 16 выходов каждый и две группы адресных формирователей $ФА'_1, ФА'_2, ..., ФА'_{16}$ и $ФА''_1, ФА''_2, ..., ФА''_{16}$. Выходы формирователей

посредством диодов попарно объединены по схеме элемента *И*. Таким образом, при расшифровке любого адреса адресный ток протекает в 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-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 — 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.

Глава 10

ИНТЕРФЕЙС ВВОДА-ВЫВОДА

10.1. Основные положения и понятия

Во всех машинах Единой системы передача информации, выполняемая по инструкциям (командам) ввода-вывода, организуется 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; при вводе 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 внешними устройствами (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 удобств 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 запоминающими устройствами (см. гл. 11). 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 каналов 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 нечетность). 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.

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 превышать 65 м, что определяется допустимыми задержками сигналов.

Интерфейс ввода-вывода обеспечивает:

стандартную организацию выполнения операций ввода-вывода незави-

симо 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 одиночное 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 однотипных 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.

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

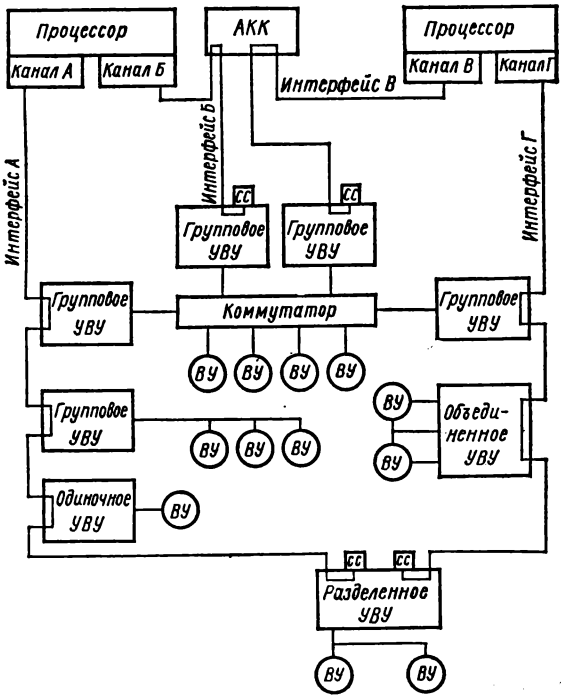


Рис. 10.1. Схема подсоединения внешних устройств

осуществляется посредством обмена управляющими сигналами. Сигналы канала и абонента связаны по принципу «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, где приняты следующие обозначения: УПД – усилители-передатчики, УПМ – усилители-приемники, СП – схема переключений и ретрансляции сигналов выборки. В соответствии с принятыми принципами $УВУ_1$ имеет высший приоритет по обслуживанию, а $УВУ_2$ – 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 определения времени. Ниже приведены наименования и условные обозначения линий.

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 (нулевой) 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 сигналами[1] 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

[1] Здесь и далее сигналы, проходящие по линиям, обозначаются, как и линии.

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 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, в которой действующий 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 от 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 принята информация, находящаяся на ШИН-А, и что эта информация может сниматься с ШИН-А. При выполнении команд ЗАПИСЬ 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 ЗАПИСЬ или УПРАВЛЕНИЕ сигнал ИНФ-А означает запрос от УВУ на выдачу каналом байта данных или информации управления.

Канал отвечает на сигнал ИНФ-А сигналами 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, в которой действующий 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, в которой действующий сигнал указывает, что выборка УВУ осуществле-

на, 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.

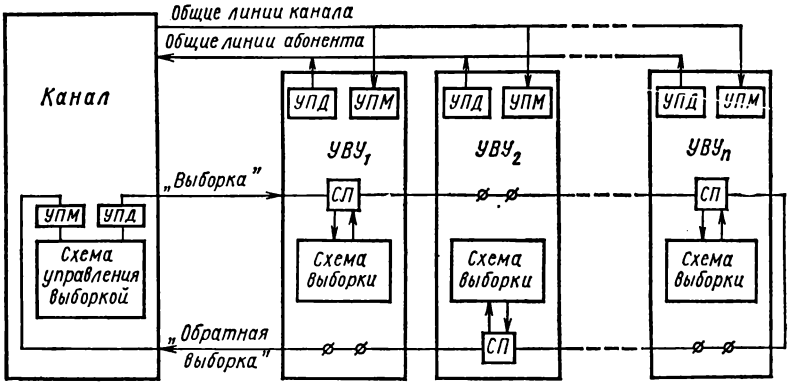


Рис. 10.2. Общая схема подключения УВУ к каналу через интерфейс ввода-вывода

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 линиям ВБР-К или ВБР-А осуществляется с помощью переключателя или путем перестановки ТЭЗ коммутации. Каждое УВУ должно обеспечивать распространение сигнала ВБР-К независимо от того, подключено 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, отвечающим 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 орга-

низует 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 использоваться 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: начальную выборку, передачу данных, окончание. Исполнение 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 монопольный режим выполнения операций, когда канал монополизирует 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 монопольном 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 монопольный 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, которые связаны с передачей информации между каналом и абонентом, к немедленно выполняемым — операции, которые не требуют передачи

байт данных (или другой информации) и для которых 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.

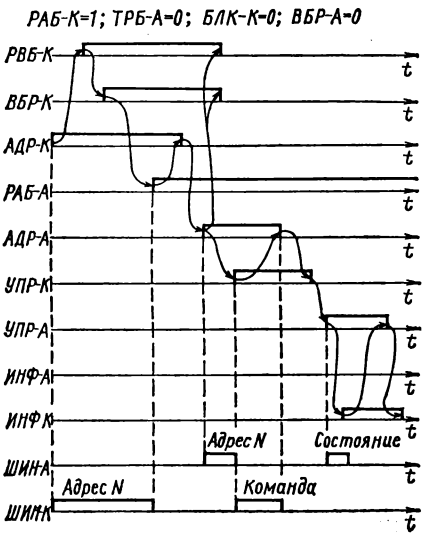


Рис. 10.3. Последовательность сигналов начальной выборки в мультиплексном канале

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 сигналы 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 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.

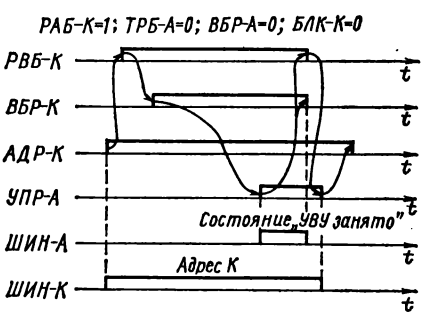


Рис. 10.4. Последовательность сигналов выборки занятого УВУ

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.

Выборка, вводимая УВУ. Обращение к каналу со стороны УВУ для передачи байта данных или байта состояния производится путем посылки 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 сигналы 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.

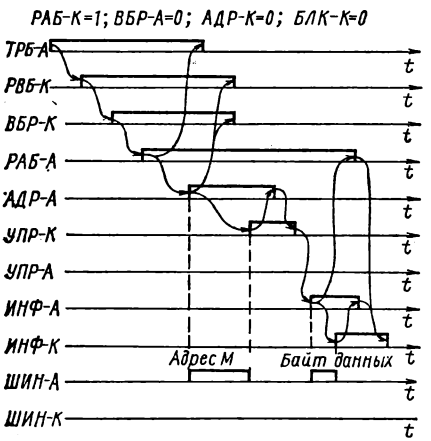


Рис. 10.5. Последовательность сигналов выборки, вводимой УВУ, на передачу данных

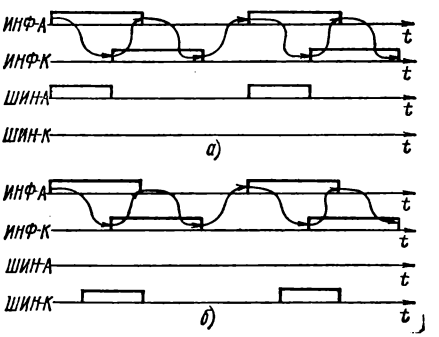


Рис. 10.6. Последовательность сигналов передачи данных: *а* — при считывании; *б* — при записи

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, 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, которое произошло до поступления байта состояния с указателем КАНАЛ КОНЧИЛ для текущей операции. Он 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 сфор-

мированы 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 определяет 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 старт-стопный 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 означает, 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.

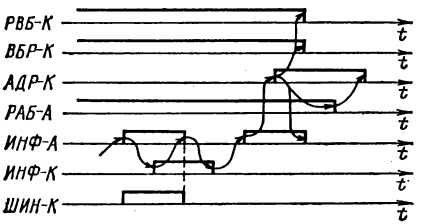


Рис. 10.7. Последовательность сигналов «Отключение от интерфейса»

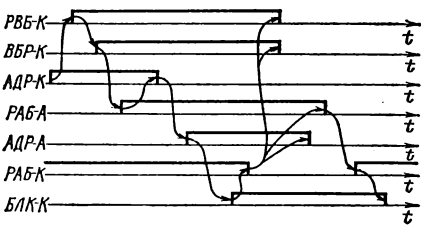


Рис. 10.8. Последовательность сигналов «Селективный сброс»

«Селективный сброс» используется для прекращения текущей операции в УВУ со сбросом всех указателей состояния для этой операции и остановом ВУ, выполняющего 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 занимается 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 программы, выполняется при нажатии на пульте управления ЭВМ кнопки **СБРОС**, при включении питания ЭВМ, при работе канала автономно от интерфейса, а также при выполнении 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 байтов производится по последовательности сигналов выборки, вводимой УВУ; при этом УВУ каждый раз посылает в канал запрос на обслуживание. Прерывание передачи 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 формиро-

вания 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 СБОЙ 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 нечетности.

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.

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».

Таблица 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

Таблица 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 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, противоположном 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 ЗАПИСЬ, 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; исключение 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.

Опознание 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 СБОЙ В УСТРОЙСТВЕ и указатель уточненного состояния ТРЕБУЕТСЯ ВМЕШАТЕЛЬСТВО.

Информация состояния (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	СБОЙ 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 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 ВУ, или групповыми УВУ. Он формируется при раздельном 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 введения 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 ВБР-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.

Глава 11

ВНЕШНИЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА ЕС ЭВМ

11.1. Состав и характеристики ВЗУ

Для запоминания больших массивов данных в ЕС ЭВМ используются запоминающие устройства, основанные на принципе записи информации на движущийся магнитный носитель. Среди них наибольшее распространение получили накопители на магнитных лентах (НМЛ) и дисках (НМД). НМЛ предназначены для хранения больших массивов информации (десятки Мбайт). Оперативное хранение достаточно больших массивов осуществляется главным образом с помощью НМД. Для обеспечения возможности сопряжения с каналами машин через интерфейс ввода-вывода и реализации стандартных процедур управления внешними запоминающими устройствами в составе технических средств ЕС ЭВМ предусмотрены устройства управления внешними устройствами УВУ.

Основные технические характеристики ряда накопителей и устройств управления ими приведены в табл. 11.1, 11.2 и 11.3.

Кроме накопителей на сменных магнитных дисках, в составе ВЗУ ЕС ЭВМ имеется накопитель на стационарных магнитных дисках. Примером такого накопителя является устройство ЕС-5064 емкостью 11 Мбайт. Оно обеспечивает скорость передачи данных, равную 1200 кбайт/с.

Таблица 11.1

Характеристики накопителей на магнитных лентах

Параметр	Шифр устройства		
	ЕС-5010	ЕС-5017	ЕС-5025
Емкость накопителя, Мбайт	20		20 и 33
Продольная плотность записи, бит/мм	8 и 32		32 и 63
Способ записи информации	БВН-1		Фазокодированный, БВН-1
Скорость записи-считывания, кбайт/с	64		64 и 126
Скорость движения ленты, м/с	2		
Время разгона ленты, мс	5,5	5	
Время останова ленты, мс	3,5	5	
Питание от трехфазной сети	380/220 В, 50 Гц		
Потребляемая мощность, кВ·А	2	2,2	
Габаритные размеры, мм	710×680×1720	790×750×1600	
Масса, кг	420	450	

Таблица 11.2

Характеристики накопителей на сменных магнитных дисках (НСМД)

Параметр	Шифр устройства			
	ЕС-5050	ЕС-5056	ЕС-5061	ЕС-5066
Емкость, Мбайт	7,25		29	100
Количество дисков в пакете	6		11	12
Количество рабочих поверхностей	10		20	20
Количество дорожек на каждой поверхности	203			411
Продольная плотность записи, бит/мм	29–44		60–90	159
Емкость одной дорожки, байт	3625		7250	12150
Среднее время доступа к информации (выборки), мс	90		75	32,5
Скорость записи-считывания, кбайт/с	156		312	806
Частота вращения дисков, об/мин	2400			3600
Потребляемая мощность, кВ·А	0,8	1,5	1,5	2,5
Габаритные размеры, мм	800×690×890	610×1050×1105	775×610×975	671×1131×1135
Масса, кг	240	260	180	280

Таблица 11.3

Характеристики устройств управления накопителями на магнитных лентах и дисках

Параметр	Шифр устройства					
	ЕС-5511	ЕС-5517	ЕС-5525	ЕС-5551	ЕС-5561	ЕС-5566
Число подключаемых накопителей	8					
Скорость передачи данных, кбайт/с	64		64 и 126	156	312	806
Число каналов ЭВМ, к которым может быть подключено УВУ	2					
Режим работы с каналом	Монопольный					
Способ обмена информацией с накопителем	Параллельный побайтный		Последовательный			
Первичное питание от трехфазной сети	220/380 В, 50 ± 1 Гц					
Потребляемая мощность, кВ·А	1,0		1,0	1,5	0,45	1,0
Габаритные размеры, мм	1200×750×1600 (стандартная стойка ЕС ЭВМ)					
Масса, кг	Не более 500		Не более 460	Не более 450		

В нем среднее время доступа к информации не превышает 10 мс. По этим параметрам данный накопитель значительно превосходит имеющиеся в номенклатуре ЕС ЭВМ накопители на магнитных ба-

рабанах ЕС-5033 и ЕС-5035. Например, емкость НМД в 2,5—5,0 раз выше емкости указанных НМБ. Большая емкость и малое время доступа к информации у такого НСМД, как ЕС-5066, также подтверждают преимущества памяти на магнитных дисках по сравнению с памятью на магнитных барабанах.

Внешние запоминающие устройства на магнитных лентах и дисках ЕС ЭВМ обеспечивают высокие скорости записи-считывания информации, достигающие сотен кбайт в секунду. Основные различия в возможностях использования этих устройств состоят в особенностях размещения информации на носителях (лентах и дисках) и обусловленных этим принятых способов доступа к ней.

Накопители на магнитных лентах принято использовать как устройства последовательного доступа к данным. В операционных системах ЕС ЭВМ организация управления обменом данными с НМЛ основывается на использовании принципа безадресного размещения больших массивов элементов данных. Элементы данных располагаются на лентах в виде определенным образом сформированных последовательностей. Выбор для обработки необходимых данных осуществляется путем поочередного «просмотра» этих последовательностей.

Накопители на магнитных дисках относятся к устройствам, обеспечивающим возможность произвольного выбора данных. Применяемая в них система адресации позволяет прямо обращаться к любым частям массивов данных, снабжаемым для этой цели специальными ключами. Для таких устройств, разумеется, не исключена также возможность и последовательного доступа к данным, как и в НМЛ.

Стандартизация подключения групп однотипных внешних запоминающих устройств к их устройствам управления обеспечивается с помощью интерфейса УВУ-ВЗУ. Из-за специфики построения, работы и использования различных накопителей каждому типу внешних ЗУ отвечает отдельный интерфейс. Основными считаются: интерфейс УВУ-накопители на магнитных лентах и интерфейс УВУ-накопители на магнитных дисках. Интерфейс УВУ-накопители на магнитных лентах имеет две модификации, которые отвечают использованию в НМЛ двух способов записи: а) без возвращения к нулю (БВН-1) и б) фазокодированного. Любой из указанных интерфейсов обеспечивает физическое подключение к соответствующему УВУ до восьми внешних ЗУ при логическом подсоединении к УВУ в любой момент времени только одного ВЗУ из подключенных. Как и в интерфейсе ввода-вывода, обеспечивается стандартизация выполнения операций в ВЗУ, включаемых в состав различных моделей ЕС ЭВМ, а также унификация последовательностей управляющих сигналов при работе с однотипными ВЗУ.

11.2. Накопители на магнитных лентах

Запись и размещение информации на магнитных лентах. В соответствии с государственными, отраслевыми и международными стандартами в ЕС ЭВМ применяются девятидорожечные магнитные ленты шириной

12,7 мм и длиной 750 м. Для лентопротяжных механизмов начало и конец рабочей части магнитной ленты отмечаются маркерами начала МНЛ и конца МКЛ ленты (рис. 11.1). В качестве маркеров используются отражающие свет наклеиваемые полоски клейкой ленты с алюминиевым покрытием. Эти полоски, длиной 30 ± 5 мм и шириной 4,8 ± 0,5 мм, помещаются на поверхности, противоположной рабочему слою, на удалении 4,6 м от начала и 7,6 м от конца ленты. Импульсный сигнал с фотоэлемента, воспринимающего отраженный световой сигнал, управляет включением тормозного механизма. Запись на ленту может начинаться от маркера начала ленты и заканчиваться в пределах 3 м после маркера конца ленты.

В рабочей части магнитной ленты служебная информация и данные размещаются в зонах. Минимальный размер зоны равен 18 байт. Максимальный размер зоны при постраничном обмене с ООП обычно не превышает 2048 байт. В общем случае зона может быть и большего размера. Все зависит от емкости основной оперативной памяти, выделяемой для обмена данными с зоной ленты. Конкретное число записываемых в зоне байт определяется числом в разрядах [48—63] управляющего слова канала УСК, задающего операцию записи (см. п. 13.1). Это количество может определяться и суммой указанных чисел нескольких УСК, связанных признаками цепочки данных. При такой записи одна зона может занимать, например, всю ленту. В ней при продольной плотности записи 32 бит/мм, вообще говоря, может быть помещено до 22 Мбайт.

В операционных системах ЕС ЭВМ управление данными, размещаемыми во внешних запоминающих устройствах, основано на использовании в качестве адресов условных наименований носителей и размещаемых на них массивов данных.

Логически самостоятельными *элементарными совокупностями данных* в массивах, имеющими вполне определенное смысловое содержание, *являются* так называемые *записи*. *Упорядоченный набор записей,* объединенных по каким-либо признакам, имеющий конкретное наименование, образует массив данных, который *называют файлом*. Понятие файла также распространяют на участки поверхностей носителей информации, где под общим названием хранятся совокупности определенным образом сгруппированных и размещенных записей. Файл может занимать, например, часть магнитной ленты, целиком ленту или несколько лент.

Для того чтобы иметь возможность именовать каждый носитель определенного типа и размера целиком, независимо от размеров файлов, введено понятие тома данных. Например, в случае магнитной ленты с понятием тома отождествляется катушка со стандартной магнитной лентой длиной 750 м, в случае дисков — пакет магнитных дисков. В определенном месте носителя, например в начале магнитной ленты, записывается условное наименование (имя) тома, его порядковый регистрационный номер, характеристика и принадлежность содержащейся в нем информации и т. п. Эта информация, сгруппированная на носителе определенным образом, образует так называемую метку тома. Метке тома размером, например, в 80 байт обычно отводят первую зону магнитной ленты (рис. 11.1). Кроме этой внутренней метки на ленте, для удобства

работы операторов, снаружи на специальной карточке, прикрепленной к катушке, указывается наименование и номер тома.

Как было указано выше, в зависимости от размера файла он может занимать часть тома, целиком том или несколько томов. В первых двух случаях имеет место соответственно одно или многофайловый том, в последнем случае — многотомный файл. На рис. 11.1 показан пример размещения на ленте только одного файла. В начале и конце файла записываются соответственно начальный и концевой 80-байтные

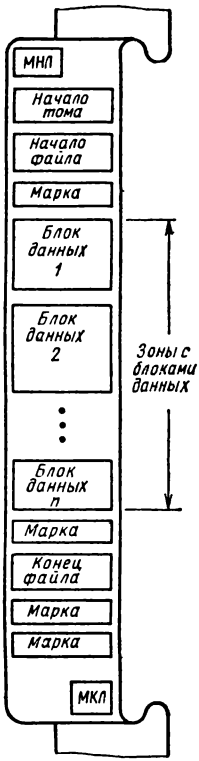


Рис. 11.1. Размещение информации на магнитной ленте

блоки (метки). В них содержится наименование и номер файла, порядковый номер тома для многотомного файла или порядковый номер файла для многофайлового тома, номер модификации файла, полученной при внесенных ранее изменениях, дата создания файла, срок хранения, вид операционной системы и др. В метке конца файла, в отличие от метки начала, указывается количество записанных в файле блоков данных. Наличие концевой метки дает возможность отыскивать файл и считывать содержащуюся в нем информацию при движении ленты в обратном направлении.

Данные, составляющие содержимое файла, записываются на магнитный носитель разделенными на блоки данных. В каждом блоке может быть либо одна запись, либо несколько. В последнем случае имеют место так называемые сблокированные записи.

На магнитной ленте блоки размещаются в отдельных зонах одинаковой или различной длины. Для более экономного использования емкости магнитных лент длины зон целесообразно выбирать соответствующими количествам байтов информации в записываемых блоках данных. В операциях обращения к магнитной ленте каждая зона рассматривается как одна физическая запись. Для считывания каждой зоны требуется отдельное (одно или несколько) управляющее слово канала УСК (см. п. 13.1). Считывание в обратном направлении возможно, когда блоки данных в файле имеют одинаковую длину.

Программы управления данными в операционных системах организуют последовательный просмотр зон с целью отыскания нужных записей. Просмотр осуществляется всегда либо с начала файла, либо с конца. Для обеспечения возможности такого просмотра операционная система выделяет в оперативной памяти не менее двух буферных областей, которые заполняются поочередно. После заполнения буфера, пока заполняется следующий, осуществляется поиск во введенных данных по

определенным признакам нужных записей. Обнаруженные необходимые записи отправляются из буфера в другие области ООП на обработку. При организации обмена данными ООП с НМЛ непосредственно через быстродействующие селекторные каналы такой процесс по скорости не уступает предварительному подводу нужной зоны, применявшемуся при отсутствии в ЭВМ операционных систем. Поэтому в современных НМЛ отпала необходимость в нумерации зон и наличии в устройствах счетчиков зон.

Зоны разделяются межзонными промежутками. Длины этих промежутков выбираются такими, чтобы в результате останова и последующего пуска ленты не было искажения и потери информации. При скорости движения ленты 2 м/с межзонные промежутки для торможения и разгона обычно находятся в пределах 12,7—15,2 мм. Лентопротяжные механизмы должны выдерживать такие малые промежутки. Это необходимо для обеспечения взаимозаменяемости НМЛ и их совместимости по магнитным лентам, содержащим ранее записанную информацию. При необходимости исключения из работы дефектных участков лент межзонные промежутки могут быть большей величины.

На 9-дорожечных лентах информация записывается побайтно (рис. 11.2). В каждой строке, образующей кадр записи, располагаются значения [0—7] разрядов байта и значение контрольного разряда k, определяемого по условию нечетности. По этому условию в любой строке всегда содержится нечетное число единиц, а в случае нулевого значения байта — единица только в контрольном разряде. Это дает возможность обнаруживать начало записи (начало зоны), подключив параллельно выходным цепям головки воспроизведения собирательную схему и фиксируя появление на ее выходе первого импульса. Последовательность импульсов с выхода этой схемы используется также для формирования синхронизирующих сигналов. На тех участках ленты, где нет записанной информации, синхросигналы отсутствуют.

Наиболее подвержены возможным механическим повреждениям края лент. Поэтому для уменьшения влияния на качество работы НМЛ этих повреждений наиболее информативные по статистическим исследованиям разряды байт и дорожка контрольного разряда размещаются ближе к середине ленты. Распределение разрядов по дорожкам показано на рис. 11.2.

В НМЛ ЕС ЭВМ применяются два способа записи: без возвращения к нулю с изменением полярности входного сигнала только при записи кода 1 (БВН-1) и фазового кодирования (ФК). Первый способ известен также как способ записи по двум уровням без промежутков. Как показано на рис. 11.3,*а*, при этом способе записи изменение направления тока в записывающей головке происходит только при записи кода 1. Соответственно этому изменяется намагниченность поверхности магнитной ленты, проходящей под головкой. Эти изменения магнитной индукции вызывают наведение ЭДС в обмотке головки воспроизведения. Таким образом, при считывании головка воспроизведения выдает сигналы той или иной полярности только при воспроизведении записи 1. Запись 0 распознается по отсутствию выходного сигнала головки воспроизведения в момент появления очередного синхроимпульса, формируемого рас-

смотренным выше способом. Исходная намагниченность магнитного слоя ленты до одного из уровней остаточной магнитной индукции, определяемых петлей гистерезиса предельного цикла, осуществляется головкой стирания. По ходу ленты (для режима записи) она располагается перед записывающей головкой. Способ записи БВН-1 используется при размещении на лентах информации с продольной плотностью 8 и 32 бит/мм (см. табл. 11.1). При плотности записи 32 бит/мм продольный интервал $d = 0{,}032 \pm 0{,}001$ мм (см. рис. 11.2). Ширина каждой дорожки равна 1,09 мм. Расстояние между средними линиями дорожек составляет 1,4 мм.

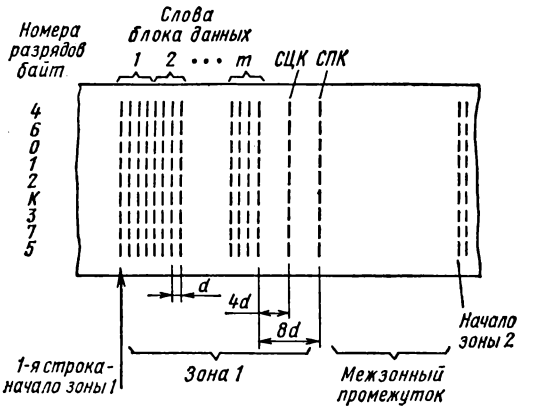


Рис. 11.2. Параллельно-последовательное размещение информации на 9-дорожечной магнитной ленте: СЦК — строка байта циклического контроля; СПК — строка байта продольного контроля четности

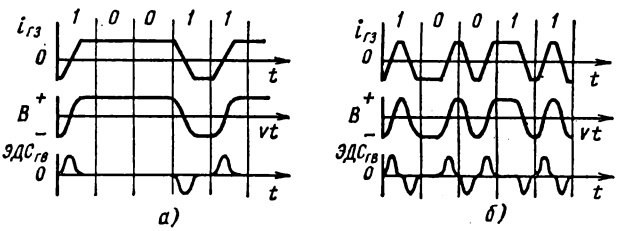


Рис. 11.3. Диаграммы, поясняющие процессы записи способами: *а* — БВН-1; *б* — ФК; $i_{ГЗ}$ — ток в обмотке головки записи; *B* — магнитная индукция магнитного слоя ленты под головкой; ЭДС$_{ГВ}$ — ЭДС, наводимая в обмотке головки воспроизведения

Способ фазового кодирования используется в НМЛ с продольной плотностью записи 63 бит/мм. При этом способе запись также осуществляется по двум уровням. Однако направление тока в записывающей головке изменяется в каждом такте записи (рис. 11,3,*б*). Изменения происходят таким образом, чтобы записи 1 во второй половине такта, например, соответствовал срез импульса тока, а записи 0 — фронт. Это приводит к тому, что при последовательной записи только единиц или только нулей частота изменения тока в записывающей головке в 2 раза выше по сравнению с записью чередующихся единиц и нулей. При воспроизведении коду 1 соответствует сигнал, например, отрицательной полярности во второй половине такта записи. Выделение таких сигналов обеспечивается соответствующим стробированием выходных цепей.

Побайтная запись обусловливает последовательно-параллельное размещение на лентах таких единиц данных, как полуслова, слова и двойные слова. Слово, например, занимает 4 строки (см. рис. 11.2). Записываемая информация сопровождается как побайтными контрольными разрядами по дорожке k, так и контрольными кодами, помещаемыми в конце каждой зоны. Последнее объясняется тем, что в НМЛ применяются еще два вида контроля записи: циклический и продольный.

При любом способе записи в конце блока данных в строке продольного контроля СПК помещается контрольный байт, формируемый путем поразрядного суммирования по mod 2 одноименных разрядов всех байт данных, записываемых в зоне. Все разряды этого байта формируются из условия четности.

При использовании этого условия, если считывание данных из зоны приходит без ошибок, то триггеры, в которых осуществляется накопление поразрядных сумм по mod 2, после подсуммирования байта продольного контроля оказываются в нулевом состоянии.

При плотности записи 32 бит/мм применяется циклический контрольный код, записываемый в строке циклического контроля СЦК. Циклический код формируется по определенному правилу путем применения операций циклического сдвига и поразрядного суммирования по mod 2. При циклическом сдвиге, например, в сторону младших разрядов значение вытолкнутого младшего разряда передается в освободившийся старший. Применение циклического кода дает возможность корректировать считываемую информацию при возникновении ошибок на одной из дорожек. В этом случае лента возвращается к началу зоны, и осуществляется повторное считывание с исправлением ошибок.

При плотности записи 8 бит/мм, соответствующей облегченному режиму работы НМЛ, использующего способ записи БВН-1, и при плотности записи 63 бит/мм по способу ФК, отличающейся высоким качеством, циклический контроль не используется. В общем случае СЦК и СПК размещаются, как показано на рис. 11.2.

Если используется только байт продольного контроля, то с промежутком в 4 строки от последней строки с байтом данных размещается СПК. Признаком конца зоны служит обнаружение интервала между строкой с последним байтом данных и строкой контрольного кода. В случае применения циклического и продольного контроля специальные схемы задержки чтения обеспечивают восприятие в конце каждой зоны обоих контрольных байт. При совместном использовании этих видов контроля циклический контрольный код служит для контроля чтения информации в ООП и в случае необходимости коррекции ошибок, продольный контроль используется для контроля процесса записи путем контрольного считывания записываемой информации.

Зоны с блоками данных отделяются от начальной и концевой меток файла специальными однобайтными разделительными зонами. В отличие от наклеиваемых маркеров начала и конца ленты эти записываемые разделительные зоны называют ленточными марками. Двумя марками после концевой метки файла отмечается также окончание каждого файла.

Для того чтобы марка воспринималась как зона, она состоит из двух строк. В первой из них содержится код специального символа, распозна-

ваемого аппаратными средствами как ленточная марка, а во второй — байт продольного контроля. В качестве специального символа может быть, например, код FF, 13 и т. п. В СПК этот код повторяется.

В оперативную память ленточная марка не считывается. При ее обнаружении для операционной системы в соответствующем разряде слова состояния канала делается отметка. Она указывает на возникновение особого случая во внешнем устройстве.

Для записи ленточных марок в системе канальных команд имеется специальная управляющая команда. Наличие на лентах разделительных марок и маркеров МНЛ обеспечивает возможность выполнения канальных команд перемещения лент вперед или назад до очередных марок, перемотки лент к маркеру МНЛ, полной перемотки лент на подающую катушку и разгрузки лентопротяжного механизма. После запуска НМЛ эти команды выполняются автономно без обмена информацией с каналом. Возможность обнаружения окончаний зон позволяет осуществлять перемещение ленты по специальным командам управления на зону вперед или назад.

Канальные команды записи и чтения оперируют блоками данных. Поэтому, если необходимо произвести изменение данных в частях зон, то операционная система перезаписывает эти зоны целиком. Все блоки данных до конца лент, располагающиеся после последнего измененного блока данных, необходимо перезаписывать вновь. Обычно корректируемый том формируется на другой ленте, назначаемой для размещения выводных файлов.

Сложность коррекции или восстановления информации на лентах вызывает необходимость принятия специальных мер, предотвращающих возможность случайной записи. Для этого, например, запись на ленту, предназначенную только для чтения, блокируется размыканием контактов в цепи записи. Они могут быть замкнуты лишь установкой на подающую катушку специального защитного кольца. При снятом кольце запись на ленту невозможна.

В процессе записи происходит контрольное считывание информации. Оно дает возможность осуществлять текущую проверку на нечетность и отсутствие перекоса записи каждой строки, а накапливая поразрядную сумму по mod 2 записываемых байт, производить продольный контроль записываемой зоны. Обнаружение любой ошибки вызывает требование прерывания. Характер ошибки, отражаемой в одном из байтов уточненного состояния ЗУ на магнитной ленте, определяется прерывающей программой. Для этого она по команде УТОЧНИТЬ СОСТОЯНИЕ вызывает из канала в процессор байты уточненного состояния.

В результате анализа причин сбоев может быть сделан вывод о дефектности участка ленты, куда осуществляется запись. Специальной программой этот участок автоматически пропускается. Вначале по команде СТЕРЕТЬ ПРОМЕЖУТОК стирается забракованная запись блока данных на участке 95—150 мм. Затем на следующем участке делается попытка повторить запись зоны.

Различаются между собой и конструкции накопителей.

Накопитель ЕС-5010. Это один из первых типов накопителей, нашедших применение в составе моделей ЕС УВМ. Он работает с устройством

управления ЕС-5511 и подключается к нему через интерфейс УВУ – НМЛ, информационные шины которого аналогичны информационным шинам интерфейса ввода-вывода.

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.

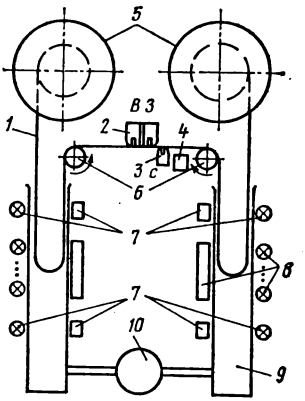


Рис. 11.4. Лентопротяжный механизм с двумя ведущими роликами:
1 — магнитная лента; *2* — блок записывающей (З) и воспроизводящей (В) магнитных головок; *3* — стирающая (С) магнитная головка; *4* — блок обнаружения маркеров МНЛ и МКЛ; *5* — катушки с лентой; *6* — ведущие ролики; *7* — фотоэлектрические датчики предельного положения петель ленты; *8* — фотоэлектрические датчики сервопривода катушек; *9* — вакуумная колонка; *10* — вакуумный насос

Перемещение магнитной ленты относительно неподвижного блока магнитных головок обеспечивается с помощью вакуумных роликов и двух следящих систем привода катушек (рис. 11.4). 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. Благодаря отсутствию жесткой связи между быстро реверсируемым участком магнитной ленты, находящимся между ведущими роликами, и основной массой ленты, намотанной на катушки, исключается опасность недопустимых натяжений и разрывов ленты.

Ведущие вакуумные ролики постоянно вращаются в противоположных направлениях. Когда нет необходимости перемещать ленту, через щелевые прорези на поверхности роликов подается воздух, создающий воздушные подушки между лентой и роликами. Это исключает трение поверхностей роликов о ленту, что уменьшает ее износ. Если требуется привести в движение ленту, то в воздуховодах ролика, вращающегося в нужном направлении, создается разряжение. Под действием разности атмосферного и внутреннего давлений лента прижимается к поверхности ролика и таким образом приводится в движение.

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 стирающей головкой 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 положения петель ленты в вакуумных колонках, прием команд из устройства управления НМЛ и выдачу команд в это устройство.

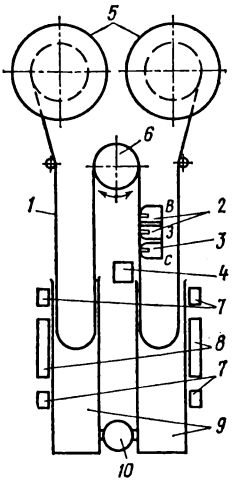


Рис. 11.5. Лентопротяжный механизм с одним ведущим роликом: *1* — магнитная лента; *2* — блок воспроизводящей (В) и записывающей (З) магнитных головок; *3* — стирающая (С) магнитная головка; *4* — блок обнаружения маркеров МНЛ и МКЛ; *5* — катушки с лентой; *6* — ведущий ролик; *7* — фотоэлектрические датчики предельного положения петель ленты; *8* — фотоэлектрические датчики сервоприводов катушек; *9* — вакуумные колонки; *10* — вакуумный насос

Для питания цепей накопителя имеются три стабилизированных источника с напряжениями −20; −6,3; +10 В и один нестабилизированный источник с напряжением −27 В.

Пульт управления предназначен для управления накопителем при подготовке его к работе и в автономном режиме.

Накопитель ЕС-5017. Устройство 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 мононролика, блок трансформаторов, вентилятор наддува, вакуумный насос, фильтр сетевой, унифицированные блоки питания, блок управления питанием.

Движение магнитной ленты относительно блока магнитных головок осуществляется с помощью мононролика и двух электродвигателей, управляемых двумя следящими системами привода катушек (рис. 11.5). Магнитная лента сцепляется с мононроликом за счет трения между его поверхностью и нерабочей поверхностью ленты. Ролик приводится во вращение при необходимости перемещения ленты. Привод реверсивного вала мононролика имеет малую инерционность. Следящая система привода мононролика обеспечивает постоянство его скорости. Устранение влияния инерционности катушек при пуске и останове магнитной ленты осуществляется, как и в накопителе ЕС-5010, за счет петель ленты в вакуумных колонках. Запись и воспроизведение информации производится сдвоенным блоком магнитных головок (в блоке девять головок записи и девять головок воспроизведения), к которому крепится стирающая головка, перекрывающая всю ширину ленты. Стирание ранее записанной информации осуществляется при записи на ленту. Блок магнитных головок подводится к магнитной ленте механизмом поворота.

Этот механизм отводит блок головок от ленты в режимах ее загрузки, разгрузки, перемотки и в аварийных ситуациях.

Накопитель ЕС-5025. В этом накопителе совмещены возможности работы как с лентами, в которых запись осуществляется по способу БВН-1 с плотностью 32 бит/мм, так и с лентами, в которых запись с плотностью 63 бит/мм производится по способу ФК. Это дает возможность использовать уже накопленные библиотеки магнитных лент, запись на которых выполнена с помощью накопителей разных типов. Накопитель 5025 создан на основе накопителей ЕС-5017 и ЕС-5014 с лентопротяжными механизмами монороликового типа.

Дальнейшим развитием накопителей типа ЕС-5025 является подсистема ВЗУ на магнитной ленте, состоящая из НМЛ типа ЕС-5003 и устройства управления накопителями ЕС-5503. В этой подсистеме нашли отражение последние достижения в области внешних запоминающих устройств, вычислительной техники, микроэлектроники. НМЛ ЕС-5003 могут работать с записями, имеющими продольные плотности 32 и 63 бит/мм. Скорость движения ленты до 5 м/с, что при плотности записи 63 бит/мм позволяет вести обмен данными со скоростью 315 кбайт/с. Заправка ленты в лентопротяжный механизм осуществляется автоматически, без прикасания к ней руками.

Обобщенная структурная схема НМЛ (рис. 11.6). Кроме лентопротяжного механизма (ЛПМ), осуществляющего реверсивное перемещение магнитной ленты относительно блока магнитных головок БМГ, и пневмосистемы ПС, предназначенной для создания вакуума и избыточного давления, необходимых для нормального функционирования ЛПМ, а также обеспечения его вентиляции, любой НМЛ содержит электронные схемы записи, воспроизведения и управления. Эти схемы связаны кодовыми и сигнальными линиями связи как между собой, так и с УВУ. Связь с УВУ осуществляется через шины малого интерфейса — в данном случае интерфейса УВУ-НМЛ (см. п. 11.4).

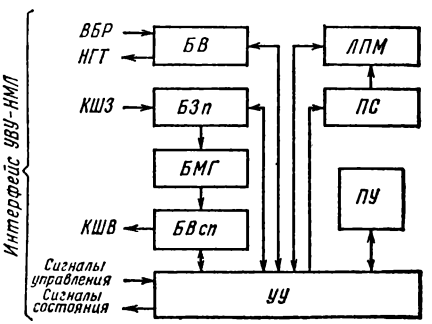


Рис. 11.6. Структурная схема накопителя на магнитной ленте

Сигналы выборки ВБР поступают в блок выборки (БВ), имеющийся в каждом НМЛ. Если выбранный НМЛ не готов к работе, его блок выборки посылает в УВУ сигнал неготовности НГТ. Готовый к работе НМЛ по сигналу выборки логически подсоединяется к УВУ, в результате чего разрешается прием из УВУ других управляющих сигналов, настраивающих накопитель на требуемый режим работы.

При записи байты информации, поступающие по кодовым шинам записи, записываются на магнитную ленту. Запись производится только

при движении магнитной ленты вперед (от МНЛ к МКЛ). Записываемые коды поступают вначале в регистр записи, имеющийся в блоке записи БЗп. С выходов этого регистра информация поступает в формирователи токов записи. Усилители воспроизведения одноименного блока БВсп производят контрольное считывание записываемой информации. Воспроизведенные байты информации как при контрольном считывании, так и по команде чтения передаются в УВУ по кодовым шинам КШВ. В процессе работы накопитель сигнализирует УВУ о текущем состоянии, посылая сигналы состояния.

Пульт управления служит для включения, выключения накопителя, а также для управления им в автономном режиме.

11.3. Устройства управления накопителями на магнитных лентах

Устройства управления УВУ предназначаются для преобразования командной информации, поступающей от каналов, в последовательности управляющих сигналов, особенности которых обусловливаются особенностями управляемых НМЛ. Иными словами, УВУ согласуют стандартные процедуры управления, реализуемые каналами, с индивидуальными особенностями конкретных НМЛ.

Для управления накопителями на магнитных лентах в системах команд каналов предусмотрены команды, приведенные в табл. 11.4.

Коды команд размещаются в [0—7] разрядах управляющих слов канала УСК. Кроме основных команд записи, считывания, считывания в обратном направлении, записи марки группы зон, важное значение для управления процессами ввода-вывода играют команды ПРОВЕРИТЬ ВВОД-ВЫВОД и УТОЧНИТЬ СОСТОЯНИЕ.

По команде ПРОВЕРИТЬ ВВОД-ВЫВОД УВУ выдает в канал байт состояния (см. с. 363). По нему операционная система определяет возможность выполнения каналом следующей команды. По команде УТОЧНИТЬ СОСТОЯНИЕ УВУ посылает в канал 6 байт уточненного состояния. В разрядах этих байт указывается более подробная информация о состоянии отдельных цепей УВУ, об обнаруженных ошибках, появившихся при выполнении предыдущей команды ввода-вывода.

Команды ЗАРЕЗЕРВИРОВАТЬ и ОСВОБОДИТЬ находят применение в тех случаях, когда накопители используются в составах вычислительных систем, в которых обращаться к внешней памяти могут разные процессоры. По первой из этих команд процессор может резервировать НМЛ за каналом, из которого данная команда посылается в УВУ. В результате УВУ будет реагировать только на команды, поступающие из данного канала. УВУ освобождается по получении команды ОСВОБОДИТЬ. По любой из этих команд УВУ посылает в канал 6 байт уточненного состояния.

Устройство управления для НМЛ, в котором запись осуществляется по методу БВН-1. Примерами таких УВУ могут служить устройства

Таблица 11.4

Команды управления НМЛ

Команда	Разряды байта кода команды							
	0	1	2	3	4	5	6	7
ЗАПИСАТЬ	0	0	0	0	0	0	0	1
СЧИТАТЬ	0	0	0	0	0	0	1	0
СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ	0	0	0	0	1	1	0	0
ПРОВЕРИТЬ ВВОД-ВЫВОД	0	0	0	0	0	0	0	0
УТОЧНИТЬ СОСТОЯНИЕ	0	0	0	0	0	1	0	0
ЗАПИСАТЬ МАРКУ ГРУППЫ ЗОН	0	0	0	1	1	1	1	1
СТЕРЕТЬ ПРОМЕЖУТОК	0	0	0	1	0	1	1	1
ШАГ НА ЗОНУ НАЗАД	0	0	1	0	0	1	1	1
ШАГ НА ГРУППУ ЗОН НАЗАД	0	0	1	0	1	1	1	1
ШАГ НА ЗОНУ ВПЕРЕД	0	0	1	1	0	1	1	1
ШАГ НА ГРУППУ ЗОН ВПЕРЕД	0	0	1	1	1	1	1	1
ПЕРЕМОТАТЬ	0	0	0	0	0	1	1	1
ПЕРЕМОТАТЬ И РАЗГРУЗИТЬ	0	0	0	0	1	1	1	1
ЗАРЕЗЕРВИРОВАТЬ	1	1	1	1	0	1	0	0
ОСВОБОДИТЬ	1	1	0	1	0	1	0	0
ХОЛОСТОЙ ХОД	0	0	0	0	0	0	1	1
ВКЛЮЧИТЬ КОРРЕКЦИЮ	0	0	0	1	1	0	1	1
ДИАГНОСТИРОВАТЬ	0	0	0	0	1	0	1	1
УСТАНОВИТЬ ПЛОТНОСТЬ 8 БИТ/ММ	1	1	1	1	1	0	1	1
УСТАНОВИТЬ ПЛОТНОСТЬ 32 БИТ/ММ	1	1	0	0	1	0	1	1
УСТАНОВИТЬ ПЛОТНОСТЬ 63 БИТ/ММ	1	1	0	0	0	0	1	1

ЕС-5511 и ЕС-5517, предназначенные для подключения к каналам накопителей на магнитной ленте ЕС-5010 и ЕС-5017 соответственно.

Устройство ЕС-5511 может подключаться через стандартный интерфейс как к селекторному каналу, так и к селекторному подканалу мультиплексного канала и работает только в монопольном режиме. Оно обеспечивает согласованную работу быстродействующих цепей каналов СК и МК с цепями низкого быстродействия, управляющими накопителями.

В УВУ 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 УВУ 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 УВУ и НМЛ, накопление и хранение информации о состоянии УВУ и НМЛ.

Основные функции УВУ по управлению работой НМЛ:

выбор заданного накопителя;

управление движением магнитной ленты в накопителе;

прием информации из НМЛ и передача ее 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.

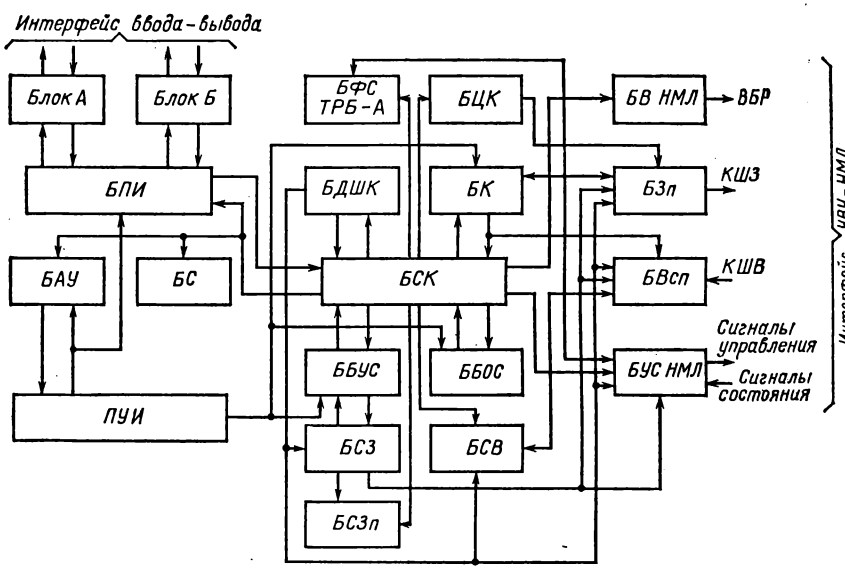


Рис. 11.7. Схема устройства управления ЕС-5511

Основные блоки устройства управления ЕС-5511 показаны на структурной схеме (рис. 11.7). Рассмотрим назначение этих блоков.

Блоки интерфейса *А* и *Б*, аналогичные по устройству и назначению, представляют собой группы линий связи, соединяющих УВУ с каналами СК или МК. В состав этих блоков входят также усилители-приемники и усилители-передатчики.

Блок подключения к интерфейсу БПИ осуществляет логическое подключение УВУ 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 сигналы, необходимые для установления связи с СК или МК, сигналы синхронизации работы блока в последовательности начальной выборки, 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 состояние УВУ 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 УВУ, и контроля работы определенных цепей УВУ. Он содержит все схемы обнаружения ошибок, кроме схемы циклического контроля.

Блок циклического контроля БЦК обеспечивает обнаружение и коррекцию (исправление) ошибок в информации, происшедших на одной дорожке магнитной ленты, а также обнаружение ошибок в информации в случае их возникновения на нескольких дорожках 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 (см. рис. 11.2), 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 совместно с анализом результатов контроля считанных байтов данных на нечетность 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 УВУ. Кроме того, в составе блока имеются цепи включения этих триггеров. Сигналы триггеров основного состояния поступают в регистр информации для последующей передачи в канал.

Блок байта уточненного состояния ББУС служит для выдачи в канал подробных сведений о состоянии УВУ и НМЛ, о конкретном сбое

или ошибке, появившейся при выполнении предыдущей операции ввода-вывода.

Блок формирования сигнала (БФС) ТРБ-А предназначен для выработки сигнала — требования абонента 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 контролирующих межзонные промежутки и расстояния между строками на ленте при записи и при воспроизведении информации, синхроимпульсов, необходимых для синхронизации работ УВУ и НМЛ, а также сигналов для своевременного включения цепей записи и воспроизведения.

Блок синхронизатора записи БСЗп формирует сигналы, синхронизирующие запись кодов байт 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 УВУ при считывании информации с ленты.

Блок выбора БВ НМЛ осуществляет логическое подключение выбранного НМЛ к УВУ, выдавая сигнал выборки ВБР. 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 перекоса, девяти одновибраторов, схемы продольного контроля на четность и схемы распознавания кода марки 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 УВУ 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 УВУ (на рис. 11.7 эти связи не показаны).

Пульт управления и индикации ПУИ предназначен для управления работой УВУ в автономном режиме, а также для визуальной проверки работы УВУ и НМЛ.

Устройство ЕС-5517 по своей структуре, выполняемым функциям и конструктивному оформлению очень близко к устройству ЕС-5511. Наиболее существенное их различие 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.

Устройство управления для НМЛ, в котором запись осуществляется по методу ФК. Примером такого УВУ может служить устройство ЕС-5514, предназначенное для управления накопителем ЕС-5014.

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 аплитудой 0,5 В, несущие информацию, закодированную в соответствии с фазокодированным способом записи. В конструктивном отношении устройства ЕС-5511 и ЕС-5514 полностью аналогичны.

Для управления НМЛ типа ЕС-5025, в котором реализована возможность записи как методом БВН-1, так и ФК, разработано УВУ ЕС-5525, основные технические данные которого приведены в табл. 11.3. В функциональном отношении в нем совмещены возможности устройств ЕС-5517 и ЕС-5514.

11.4.
Интерфейс УВУ — НМЛ

Этот интерфейс обеспечивает подключение к одному устройству управления до восьми накопителей на магнитных лентах. При использовании 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. 11.8. Все линии разделяются на четыре группы:

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 двадцати восьми информация передается от УВУ к НМЛ и по двадцати семи — 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 девять 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.

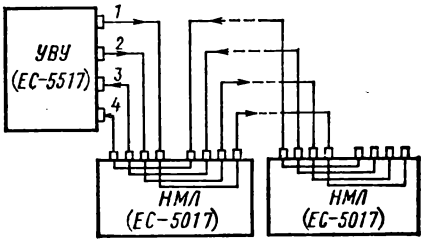


Рис. 11.8. Схема подключения накопителей на магнитных лентах к УВУ

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 восемь, 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 шесть 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 сигнал НЗЗ. Если сигнал НЗЗ отсутствует (на катушке ленты отсутствует кольцо защиты файла), 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, если 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	Сигналы НГТ
Выключено питание Не подключен к УВУ Нет ответа на сигнал выборки НМЛ	0	0
Не готов к выполнению операции Не переведен на работу с УВУ Выполняет действия по сигналу ПИР Требуется вмешательство оператора	0	1
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 выполняться 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 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. 11.9. 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, СЦК обозначет запись строки циклического контроля, а СПК – 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.

Кроме рассмотренных, в интерфейсе УВУ — накопители на магнитных лентах используются *сигналы режимов:*

ВСВ — включить счетчик времени;

УНП — установить низкую плотность.

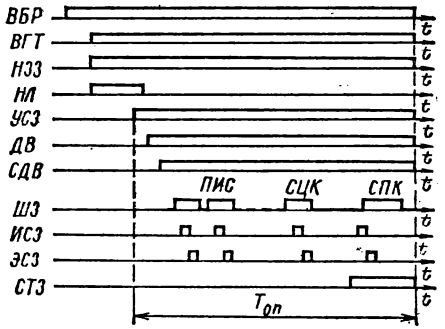


Рис. 11.9. Временна́я диаграмма выполнения операции записи

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.

11.5. Накопители на магнитных дисках

Широкое развертывание работ по созданию и совершенствованию НМД обусловлено стремлением иметь внешнюю память с произвольным доступом к информации, которая сочетала бы большую емкость с высоким быстродействием. В настоящее время емкость НМД достигает десятков и сотен миллионов байт, а скорость записи — считывания — сотен тысяч байт в секунду.

Рассмотрим сначала общие принципы построения НМД и расположения информации на магнитных дисках. В качестве примера рассмотрим накопитель на сменных магнитных дисках. Такие накопители получили наибольшее распространение.

Носители информации — магнитные диски (рис. 11.10), обычно изготовляются из алюминия и имеют ферролаковое или кобальтвольфрамовое покрытие толщиной 1—2 мкм. Сами диски имеют толщину, например, 1,27 мм и диаметр 356 мм. На обеих сторонах диска информация записывается по концентрическим магнитным дорожкам. Из дисков собираются неразборные пакеты. Число дисков, составляющих пакет,

может быть от 4 до 24 с расстоянием между дисками до 25 мм. Верхний и нижний диски пакета являются защитными. Их наружные поверхности (соответственно верхняя и нижняя), наиболее подверженные повреждениям, обычно являются нерабочими. Внутри пакета у рабочих дисков обе поверхности используются для хранения информации. Одноименные дорожки всех рабочих поверхностей пакета (имеющие одинаковые диаметры) образуют как бы концентрические информационные цилиндры. Это понятие применяется в адресной системе НМД.

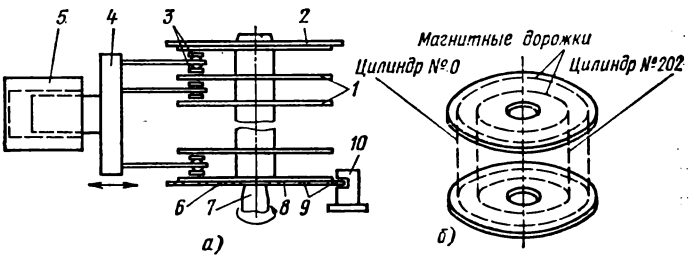


Рис. 11.10. Конструкция запоминающего устройства на сменных магнитных дисках: *а* — структурная кинематическая схема; *б* — диски и расположение на них дорожек записи; *1* — пакет дисков; *2* — верхний защитный диск; *3* — плавающие магнитные головки записи-считывания; *4* — каретка; *5* — линейный двигатель; *6* — сдвоенные прорези маркера начала оборота; *7* — шпиндель; *8* — нижний защитный диск с краевыми прорезями; *9* — секторные прорези; *10* — датчик секторных синхросигналов и сигналов-маркеров

Пакеты дисков хранятся в пылезащитных пластмассовых коробках. Верхняя часть коробки снимается с основания только при установке пакета в дисковод накопителя. При установке в рабочее положение пакет надевается на шпиндель электропривода вращения дисков, где фиксируется с высокой точностью.

Универсальные магнитные головки записи-воспроизведения располагаются на подвижных рычагах, которые жестко закреплены на каретке. Каретка приводится в движение электрическим или гидравлическим линейным двигателем. При движении каретки магнитные головки перемещаются вдоль радиуса дисков. Система управления линейного позиционирующего двигателя строится таким образом, что обеспечиваются быстрое перемещение и установка с высокой точностью магнитных головок на позицию требуемого информационного цилиндра.

В отличие от НМЛ запись на диски осуществляется бесконтактным способом. Применяются магнитные головки, «плавающие» на воздушной подушке примерно на расстоянии 2,5—3 мкм от рабочей поверхности. Воздушная подушка создается потоком воздуха, увлекаемого поверхностью диска. Подъемная сила, уравновешивающая прижимающее усилие пружинной подвески, создается за счет угла атаки, который имеет сферическая поверхность блока головок по отношению к воздушному потоку.

Из-за необходимости наличия воздушной подушки магнитные головки вводятся внутрь пакета только при достижении дисками достаточной скорости вращения. Достижение требуемой скорости определяется по

частоте импульсов, формируемых датчиком секторных синхросигналов (поз. *10* на рис. 11.10). Этот датчик вырабатывает сигналы в моменты прохождения мимо него прорезей, имеющихся в нижнем защитном диске пакета. Из этих прорезей две расположены в непосредственной близости друг к другу. По ним за каждый оборот пакета вырабатывается один сигнал-маркер, служащий для всех дисков меткой начала нового оборота.

В каждом плавающем блоке головок расположены две магнитные головки: универсальная головка записи-воспроизведения и головка туннельного стирания. Универсальная головка при записи полностью удаляет предыдущую запись, имевшуюся на дорожке. Головка туннельного стирания, установленная по направлению записи за универсальной головкой, создает стирающее поле на краях записываемой дорожки. Путем такого уменьшения ширины дорожки уменьшается взаимовлияние друг на друга соседних дорожек и увеличиваются допуски на точность подвода головок к нужному цилиндру.

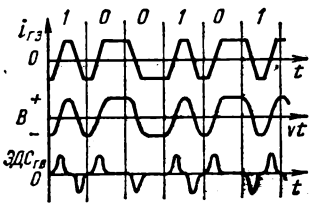


Рис. 11.11. Диаграмма, поясняющая способ записи по двум уровням с частотной модуляцией: $i_{ГЗ}$ — ток в обмотке головки при записи, B — магнитная индукция магнитного слоя диска под головкой; $ЭДС_{ГВ}$ — ЭДС, наводимая в обмотке головки при отсутствии тока записи

В накопителях на магнитных дисках применяются способы записи по двум уровням с частотной модуляцией (рис. 11.11.). Этот метод записи по сути дела с частотным кодированием двоичных символов похож на метод записи фазового кодирования, применяемого в НМЛ. При двухчастотном методе кодирования коду 1, например, ставится в соответствие в такте записи частота $2f$, а коду $0 - f$. Как следует из рис. 11.11, при записи 1 направление тока в обмотке универсальной головки изменяется дважды, а при записи 0 — только один раз. При воспроизведении считывание 1 распознается по наличию импульса любой полярности во втором полутакте.

Метод частотной модуляции позволил достичь в настоящее время продольной плотности неконтактной записи 40—90 бит/мм.

Размещение информации на магнитных дисках. Накопители на магнитных дисках относятся к ВЗУ с произвольным доступом к информации. В соответствии со стандартами, учитывающими рекомендации международной организации по стандартизации (ИСО), информация на дисках располагается записями на концентрических дорожках последовательным кодом. Дорожки в пределах диска и, следовательно, цилиндры нумеруются начиная с 0 от края диска к центру. Рабочие поверхности, а следовательно, и дорожки в пределах цилиндра, нумеруются сверху вниз, начиная с 0.

Наборы данных, состоящие из записей, в виде блоков постоянной или переменной длины записываются с межблоковыми промежутками

вдоль магнитных дорожек. Физический адрес блока на пакете как физической единицы информации определяется пятибайтным кодом структуры ССННR. В этом коде СС — двухбайтное двоичное число, указывающее номер цилиндра; НН — двухбайтное двоичное число, указывающее номер дорожки; R — однобайтное двоичное число — номер блока на дорожке.

Каждая дорожка, начиная от метки начала оборота (рис. 11.12), вначале содержит блок собственного адреса (идентификатор) дорожки БСА. В этом блоке сначала идет байт кода флажка БКФ, указывающий: пригодна или не пригодна дорожка для записи и основная она или запасная. Далее следует код номера данного цилиндра № Ц (СС),

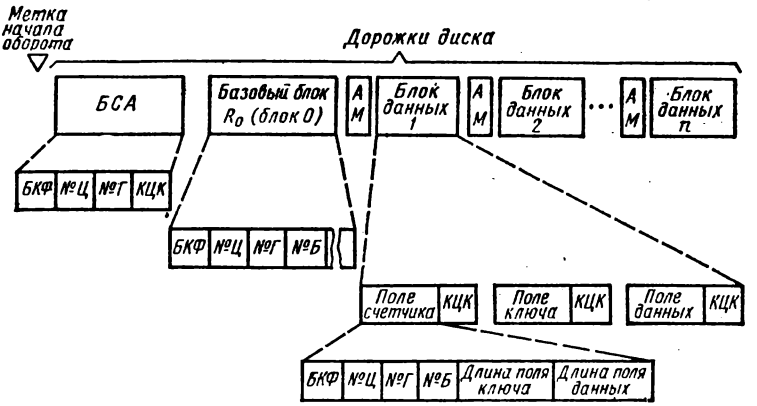


Рис. 11.12. Размещение информации на дорожке диска

за ним — код номера головки № Г (НН — собственный номер дорожки в цилиндре), и завершает блок двухбайтный код циклического контроля.

Блоки данных записываются на дорожку разделенными внутриблоковыми промежутками на две-три области. Нулевой блок, следующий за БСА, является служебным (базовый блок R_0). Он содержит информацию для операционной системы, указывая степень заполнения дорожки. Базовый блок начинается с флажкового байта, повторяющего БКФ блока собственного адреса.

За флажком следует адрес ССНН, состоящий из номера цилиндра № Ц и номера головки № Г. Содержимое этой части зависит от пригодности и использования дорожки. Если она основная и пригодная, то в адресной части базового блока повторяется адрес блока БСА, если же непригодная — то адрес запасной дорожки. На запасной дорожке, если она исправна и используется вместо одной из основных, в адресной части базового блока записываются номера цилиндра и головки, соответствующие этой дублируемой дорожке. Неисправная запасная дорожка никогда не используется и не дублируется. Во всех ее флажковых байтах делается пометка о ее непригодности. Переадресация на дублирующую запасную дорожку через неисправную запасную не допускается.

За кодом адреса в базовом блоке следует байт номера блока № Б. Содержимое этого байта в данном блоке всегда равно нулю. После указанных восьми байт при блоках данных одинаковой длины размещаются два байта циклического контроля, завершающие первую область базового блока R_0. В последующей области, отделенной от первой внутриблоковым промежутком, находится блок данных, также состоящий из восьми байт, не считая полуслова циклического контроля. В этих байтах записывается номер последнего занятого блока данных на дорожке и размер (в байтах) остающегося на ней свободного участка. При размещении данных в блоках переменной длины в первой области блока R_0 после № Б может содержаться дополнительная служебная информация.

Все блоки, в которых содержатся рабочие записи, имеют одинаковую структуру. Если используются ключи записи, то блоки состоят из трех полей: поля счетчика, поля ключа и поля данных. В поле счетчика указывается полный адрес блока, включая его номер (рис. 11.12), при этом на основной дорожке № Ц и № Г — те же, что и в БСА. В байте флажка на основной дорожке отмечается ее пригодность или непригодность. Этот флажок также копирует флажок БСА. В адресной части поля счетчика на запасной дорожке записывается полный адрес блока основной дорожки, которую дублирует запасная. За адресом следуют указатели длины полей ключа и данных.

В поле ключа содержится условное наименование блока данных, некоторые дополнительные пояснения к этому наименованию, принадлежность определенному файлу и т. п. Содержимое ключа формирует пользователь.

В поле данных непосредственно размещается рабочая информация, в конце которой, как и во всех предыдущих полях, записывают два байта циклического кода. Теоретически поле максимально возможной длины может содержать 65 535 байт. Однако практически поля данных не распространяются за пределы одной дорожки. Если один блок данных занимает всю дорожку, то вместе с БСА и базовым блоком на дорожку может быть помещено максимум 3625 байт. При более мелком дроблении записей необходимость включения межблоковых промежутков приводит к снижению полезной емкости дорожки.

Межблоковые промежутки имеют протяженности, эквивалентные записям двух — трех десятков байтов. В промежутках записывается служебная информация, необходимая для ввода в синхронизм схем записи-воспроизведения. Кроме того, в промежутках, предшествующих блокам данных, содержатся двухбайтные адресные маркеры АМ. По ним УВУ определяет приближение очередного блока данных. Непосредственно перед началом каждого блока располагается синхронизирующий байт (если в промежутке есть АМ, то сразу за АМ). Он указывает на то, что за ним следует первый байт блока.

Каждому пакету дисков ставится в соответствие понятие тома. Тому дается название, которое указывается в его 80-байтной метке. Метка тома записывается с начала 0-й дорожки 0-го цилиндра. Кроме названия, она содержит порядковый номер тома, указатели защиты информации, оглавления, имя и адрес владельца и др.

В НМД, являющихся ВЗУ с произвольным доступом к информации,

данные размещаются и организуются таким образом, что обеспечивается возможность быстрого отыскания нужных цилиндра, дорожки и расположенных на ней записей. Для этого формируется оглавление тома, местоположение которого указывается в метке тома. В оглавлении содержатся описания (метки) с указанием адресов частей всех наборов данных (файлов), размещаемых или уже имеющихся в томе. При этом организуется такая система меток, что подробно описывается занятость определенных областей, отведенных для тех или иных наборов, занятые и свободные места на каждой дорожке. В пределах набора данных последовательное размещение информации идет вдоль цилиндра сверху вниз и по мере перехода с цилиндра на цилиндр — в глубь пакета.

При начальном формировании тома вначале записываются его метка, оглавление и на каждой дорожке — ее собственный адрес и базовый блок R_0. Одновременно проверяются поверхности дисков. Если обнаруживаются дефектные дорожки, то они заменяются запасными путем записей в базовых блоках R_0 соответствующих указателей переадресации и пометками в флажковых байтах. Информация об использовании запасных дорожек указывается в оглавлении тома.

Накопитель на сменных магнитных дисках ЕС-5050. Накопитель предназначен для работы в составе всех моделей ЭВМ ЕС совместно с устройством управления ЕС-5551 и подключается к нему через интерфейс УВУ-НМД.

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 — 7,25 Мбайт;

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 имеются 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 960 до 1030 Па (860—1060 Мбар).

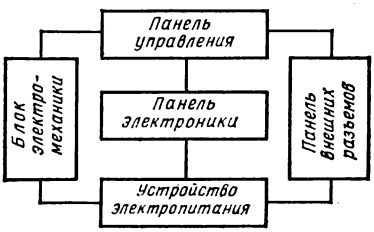


Рис. 11.13. Схема накопителя на сменных магнитных дисках ЕС-5050

Первичное питание НМД осуществляется от трехфазной сети переменного тока напряжением 380/220 В с отклонениями от номинала на $+10 \div -15\%$ и частотой 50 ± 1 Гц. Для питания электронного оборудования накопителя блок питания вырабатывает стабилизированные напряжения 5; 6,3; 20; —20; —40 В и нестабилизированные 27; 40 В.

Структурная схема НМД ЕС-5050 приведена на рис. 11.13. Панель электроники предназначена для размещения на ней типовых элементов замены (ТЭЗ), соединения их между собой и другими блоками и узлами накопителя. В блоке электромеханики 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 предназначена для соединения накопителя с устройством управления при работе в автоматическом режиме и с прибором контроля накопителя при работе в автономном режиме. Устройство электропитания обеспечивает электронное оборудование накопителя источниками напряжений указанных выше потенциалов. Панель управления 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.

Накопители на сменных магнитных дисках ЕС-5052 и ЕС-5056 (емкость пакета дисков каждого из этих накопителей равна 7,25 Мбайт) выполняют те же функции, что и накопитель ЕС-5050. Принципиальных различий между ними нет.

Накопитель на сменных магнитных дисках ЕС-5061. НСМД работает со сменным пакетом дисков ЕС-5261 или другими совместимыми с ним пакетами.

Основные параметры НСМД:

число дисков в пакете — 11;

число рабочих поверхностей – 20;
число магнитных дорожек на одной рабочей поверхности – 200 основных и 3 резервных;
емкость пакета – 29,176 Мбайт;
диаметр диска – 356,36 мм;
диаметр магнитной дорожки – от 330,5 до 226,9 мм;
покрытие – ферролаковое;
метод записи – двухчастотный без возвращения к нулю;
частота вращения пакета дисков – 2400 об/мин;
число каналов для передачи данных – 1;
число каналов для приема данных – 1;
число магнитных головок записи-считывания – 20;
скорость передачи данных – 312 кбайт/с;
максимальное время поиска цилиндра – 80 мс;
минимальное время поиска цилиндра – 20 мс;
плотность информации на магнитной дорожке – от 60 бит/мм (для дорожки № 000) до 90 бит/м (для дорожки № 202);
ширина дорожки – 0,152 мм;
расстояние между серединами соседних дорожек – 0,257 мм;
средняя потребляемая мощность – 1,5 кВ·А;
диапазон рабочей температуры – $+10 \div +35$ °С.

Как и другие типы накопителей на сменных магнитных дисках, НМД ЕС-5061 выполняет три основные операции: поиск, чтение и запись. Для выполнения любой из них необходимо раскрутить пакет дисков до номинальной частоты вращения 2400 об/мин, выбрать одну из 20 магнитных головок записи-считывания и установить все головки на один и тот же информационный цилиндр в соответствии с заданным адресом (в каждый данный момент только одна магнитная головка осуществляет запись или считывание информации). Информационные цилиндры образуют каждые двадцать дорожек с одинаковым номером, так что в пакете дисков имеется 203 цилиндра, расположенных концентрично.

Обмен информацией с оперативной памятью осуществляется через устройство управления накопителями на магнитных дисках ЕС-5561.

11.6. Устройство управления накопителями на магнитных дисках

Устройство управления ЕС-5551 предназначено для управления работой накопителей на магнитных дисках ЕС-5050, ЕС-5052 и ЕС-5056 и подключения их к каналам ввода-вывода. Устройство управления может быть подключено к селекторному каналу или к селекторному подканалу мультиплексного канала; оно может быть подключено также к двум каналам. К УВУ можно подсоединять до восьми накопителей.

Для управления накопителями в системе команд УВУ предусмотрено 40 команд. Их коды указываются в нулевых байтах управляющих слов канала, из которых составляются канальные программы управления вводом-выводом, УВУ и накопителями. Среди этих команд имеются команды, обеспечивающие установку блоков магнитных головок на позицию

требуемого цилиндра, поиск нужной записи на дорожке или цилиндре, чтение или запись различных полей на диске, управление вызовом байт состояния и уточненного состояния, обходом дефектных областей дисков, наложением запрещающих масок, резервированием и освобождением устройств. Поиск необходимых данных может осуществляться по собственным адресам, по номерам цилиндров и головок, по ключам, непосредственно по содержимому блоков данных. При этом отыскание требуемого адреса, ключа и т. д. может производиться при условии, что искомый параметр равен или больше некоторой величины, код которой находится в определенной ячейке ООП.

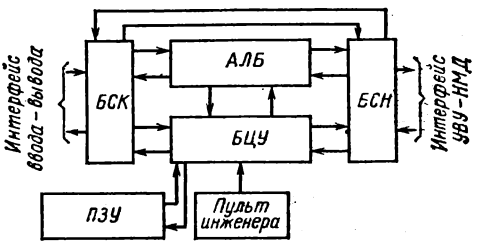


Рис. 11.14. Схема устройства управления ЕС-5551

Выполнение команд, поступающих из каналов ввода-вывода, осуществляется с помощью комплексной управляющей микропрограммы, которая хранится в блоке памяти микрокоманд (постоянном запоминающем устройстве ПЗУ), имеющемся в составе устройства управления (рис. 11.14). Комплексная управляющая микропрограмма состоит из 15 подпрограмм определенного целевого назначения. Среди них имеются подпрограммы: начальной выборки, декодирования канальной команды, передачи в канал байтов состояния и уточненного состояния, обработки байта флажка, чтения и синхронизации, приема байтов информации из канала и передачи на запись в накопитель и т. д. В ПЗУ также хранятся специальные тестовые микропрограммы, которые используются для проверки работоспособности УВУ и подключенных накопителей при работе устройства управления в автономном режиме. Все это помещено в ПЗУ трансформаторного типа с П-образными сердечниками емкостью 3072 48-разрядных микрокоманд. Время цикла ПЗУ — 0,5 мкс.

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 сопряжения с каналами (БСК). С помощью БСК осуществляется физическое подключение УВУ к шинам стандартного интерфейса ввода-вывода, посредством которого оно соединяется с селекторным каналом.

Основные функции БСК следующие:

согласование уровней сигналов стандартного интерфейса с уровнями сигналов УУ;

прием из каналов ЭВМ и контроль на нечетность адреса УВУ и накопителей, подключенных к УВУ; сравнение адреса УВУ с адресом,

принятым из канала; передача адреса в блок центрального управления (БЦУ);

логическое подключение УВУ к каналу ЭВМ по сигналам управления из БЦУ; прием из канала и контроль на нечетность команд, служебной информации, данных и передача их в БЦУ по сигналам БЦУ;

прием байтов состояния и данных из БЦУ и выдача их в канал;

логическое отключение УВУ от канала;

хранение служебной информации о состоянии накопителей, подключенных к УВУ, и выдача ее в АЛБ по сигналам из БЦУ; выдача в канал байта состояния УВУ при обращении канала к устройству управления, занятому в это время выполнением операции.

Блок центрального управления (БЦУ). Этот блок служит для формирования и выдачи управляющих и синхронизирующих сигналов во все другие блоки УУ. Выработка управляющих сигналов производится в такой последовательности: формируются 12-разрядные адреса микрокоманд управляющей микропрограммы, размещенной в ПЗУ, и осуществляется их контроль, затем адреса микрокоманд выдаются в ПЗУ, далее из ПЗУ принимаются и контролируются выбранные микрокоманды, производится дешифровка микрокоманд и выработка управляющих сигналов для всех блоков УВУ. После начального пуска устройства управления блок центрального управления всегда считывает из ПЗУ микрокоманду по нулевому адресу. Адреса последующих микрокоманд формируются в соответствии с информацией, содержащейся в предыдущей микрокоманде, с учетом состояния индикаторов переходов функциональных блоков УВУ.

Кроме того, блок центрального управления выполняет следующие функции:

считывание и прием команд канала, управляющей информации и данных из блока сопряжения с каналами и их обработку в соответствии с микрокомандами управляющей микропрограммы;

контроль данных и управляющей информации, принятой из блока сопряжения с каналами, и передачу их в блок сопряжения с накопителями по микрокомандам управляющей микропрограммы;

считывание и прием данных и информации о состоянии накопителей из блока сопряжения с накопителями, их контроль и передачу в блок сопряжения с каналами по микрокомандам управляющей микропрограммы;

формирование информации о состоянии УВУ и передачу ее в блок сопряжения с каналами;

останов УВУ при обнаружении ошибок в адресах микрокоманд, в самих микрокомандах и в арифметико-логическом устройстве;

проверку функционирования блоков устройства управления, накопителей и их взаимодействия по тестовой микропрограмме в автономном режиме работы.

Арифметико-логический блок (АЛБ). Блок служит для выполнения логических и арифметических операций над байтами информации, поступающей из блока сопряжения с каналами, блока сопряжения с накопителями или блока центрального управления. Например, в АЛБ вычисляется разность адресов цилиндров, выполняется контроль информации на нечетность, подсчитываются байты циклического контроля и т. п. Резуль-

тат 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 УВУ, которые (индикаторы) указываются в кодах микрокоманд управляющей микропрограммы.

Блок сопряжения с накопителями (БСН). Этот блок обеспечивает физическое подключение к УВУ до восьми накопителей, а также логическое подсоединение в нужное время (по сигналам из БЦУ) к УВУ 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 выдачу 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;

выдача в БЦУ сигналов, определяющих тип логически подключенного накопителя;

прием из логически подключенного накопителя сигналов о его состоянии и передача их в АЛБ и БСК;

логическое отключение УВУ от накопителя по сигналу из БЦУ.

Пульт инженера. На этом пульте размещены органы ручного управления в автономном режиме работы и световые индикаторы состояния ряда регистров устройства управления. С пульта управления осуществляется включение УВУ и его питания, пуск управляющей микропрограммы УВУ с заданного адреса ячейки ПЗУ и останов этой микропрограммы по заданному адресу, зацикливание участков управляющей микропрограммы, блокировка останова управляющей микропрограммы при ошибках, занесение заданных кодов информации на регистры БЦУ, включение УВУ в режим сканирования ПЗУ для проверки считывания микрокоманд управляющей микропрограммы, индикация состояния регистров БЦУ, сброс элементов УВУ.

11.7. Интерфейс УВУ — НМД

Этот интерфейс обеспечивает подключение к одному устройству управления ЕС-5551 до восьми накопителей на магнитных дисках ЕС-5050, ЕС-5052, 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, для выполнения операций записи или считывания в которых необходимо выбрать не только накопитель, но и соответствующий цилиндр, а также вполне определенную магнитную головку. Перед началом операции возможно перемещение магнитной головки на необходимую дорожку диска.

Общая схема подключения накопителей на магнитных дисках (НМД) к УВУ приведена на рис. 11.15. Обозначения для этой схемы: СЛ — селекторные линии; ШЛ — шинные линии; БР — блок резисторов, используемый для согласования линий.

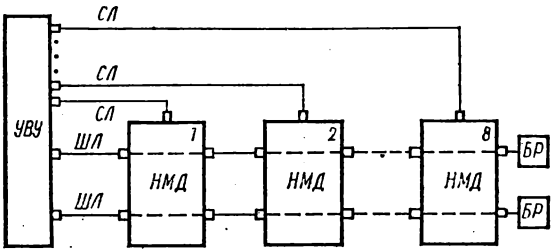


Рис. 11.15. Схема подключения накопителей на магнитных дисках к УВУ

Селекторные линии служат для передачи следующей информации:

СЛ1 — сигналов «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. 11.5.

Таблица 11.5

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 ОТВЕДЕНЫ;
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 сигналам 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, МШОС-2, МШОС-6 и МШОС-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 информационными сигналами состояния по МШОС-1, МШОС-2, МШОС-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 головки 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 заданного перемещения головок.

Глава 12

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 УВУ. 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.

В устройства ввода-вывода электромеханического типа входят 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 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 приводятся 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 типа ЕС-6012 и ЕС-6013. Эти устройства выполнены по одинаковым общим схемам и отличаются 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.

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 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, блок сопряжения с каналом БСК.

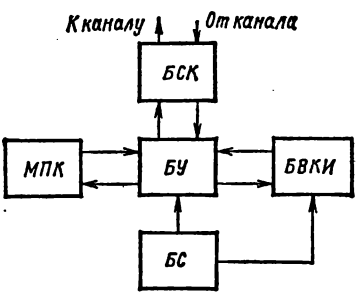
Механизм подачи, транспортировки и укладки перфокарт обеспечивает поочередное перемещение перфокарт от подающего магазина к приемному

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

Цифровая группа	Зональная группа: 12	11	0		12 0	12 11	11 0	12 11 0	12	11	0		12 0	12 11	11 0	12 11 0	
0	&	–	0	Пробел	{	}							ц	й	я	ь	8-1
1	A	J	/	1	a	j	–	ы	НЭ	СУ1	Д01	Д17			Д31		9-1
2	B	K	S	2	b	k	s	э	НТ	СУ2	Д02	СИН					9-2
3	C	L	T	3	c	l	t	щ	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	щ	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	ъ	ЗБ	Д07	АР2	КЛ					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	ПУС	АРТ	Д00	Д16	9-8-1
8-2	[	]	\	:	∂	к		Б	Д14	Д18	Д10	Д26				З	9-8-2
8-3	•	¤	,	#		л		Ц	ВТ	Д15	Д11	Д27	И		У	Ш	9-8-3
8-4	<	*	%	@	ф	м	т	Д	ПФ	РФ	Д12	СТП	Й	П	Ж	Э	9-8-4
8-5	(	)	–	'	г	н			ВК	РГ	КТМ	НЕТ		Я		Щ	9-8-5
8-6	+	;	>	=			ж	Ф	ВЫХ	РЗ	ДА	Д30	Л		Ь	Ч	9-8-6
8-7	!	^	?	"	и	п	в	Г	ВХ	РЭ	ЗВ	ЗМ			Ы	ЗБ	9-8-7

Рис. 12.1. Структура кода КПК-12

Рис. 12.2. Общая схема устройства ввода информации с перфокарт



карты, неподаче очередной карты и о положении основных деталей механизма. Механическая часть устройства рассчитана на работу в старт-стопном 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 байтов через БУ в БСК. 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 поколонные кодовые комбинаций 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 байтов данных, байтов 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 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

Таблица 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

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 канал 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 – ВУ КОНЧИЛО, в 6-м разряде – СБОЙ В УСТРОЙСТВЕ, в 7-м разряде – 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 по окончании передачи данных между устройствами и каналом. Передача данных прекращается либо по концу перфокарты, либо по сигналу 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 – ОШИБКА НА ШИН-К, 3-й разряд – СБОЙ В ОБОРУДОВАНИИ, 4-й разряд – для указателя ОШИБКА В ДАННЫХ, 5-й разряд – 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 устройства 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 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 построчной пробивки; карты под пуансоны, связанные с пробивными электромагнитами, подаются широкой стороной. Эти устройства рассчитаны 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, принимаемых 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. 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.

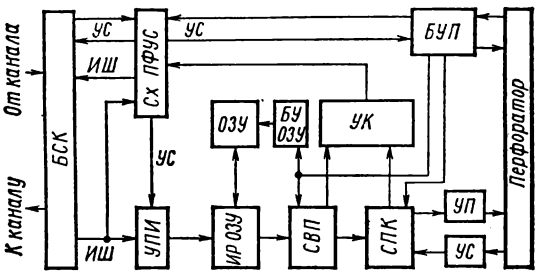


Рис. 12.3. Общая схема устройства вывода на перфокарты

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 Сх ПФУС, узел перекодирования информации УПИ, оперативное запоминающее 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 усилители 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, поступающих 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 Сх ПФУС, 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 вентилей, подключенных к первым шести выходам схемы преобразования, обеспечивает передачу в информационный регистр ОЗУ кодов 12, 11, 0, 1, 2 и 3-й позиций для одной колонки перфокарты, а вторая группа выходных вентилей — 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 вентилей. 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 последовательными 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.

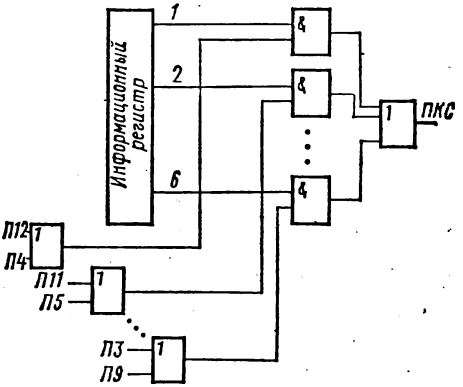


Рис. 12.4.
Схема выделения позиций

Схема выделения позиций обеспечивает «разворот» поколонного параллельно - последовательного кода в построчный. 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 концевого 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 и т. д. Циклы считывания из нечетных ячеек ОЗУ повторяются 6 раз. После этого 6 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 вентилей. При контроле отперфорированной карты схема преобразования преобразует 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.

Таблица 12.2

Команды	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 поступает 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 различие их 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 устройстве уже сформирован байт основного состояния, еще не посылавшийся 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 3-м разряде представляет указатель ЗАНЯТО, в 4-м — указатель КАНАЛ КОНЧИЛ, 5-м — ВУ КОНЧИЛО, в 6-м разряде — 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, или после приема 160 байт данных в режиме работы без преобразования кодов). 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-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.

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 механизмом АЦПУ-128-6). Оба 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 работе [17]. Дополнительно введено в основном управление перемещением бумаги от закольцованной четырехдорожечной перфоленты, что позволяет при печати делать пропуски до 64 строк. Скорость печати — 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, при некотором усложнении блоков управления обладают большей гибкостью при печати различных данных.

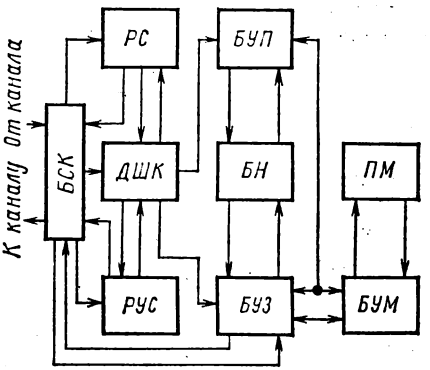


Рис. 12.5. Общая схема алфавитно-цифрового печатающего устройства

Общая схема алфавитно-цифрового печатающего устройства 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 каналом и устройством через стандартный интерфейс ввода-вывода. В нем, в частности, формируются сигналы идентификации, производится контроль поступающей из канала информации по четности, формируется контрольный разряд для байтов 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 байтов данных, представляющих собой коды символов, из канала и запись их в БН. При этом прием байтов 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 и 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 $128 \times 84 = 10\,752$ 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.

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

Таблица 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

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. Различие 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 с пропуском после печати. Различие 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; 3-й разряд 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 перемещения 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 СБОЙ 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 СБОЙ 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

Для организации непосредственного обмена информацией между оператором и вычислительной машиной через стандартный интерфейс ввода-вывода в ЕС-1022, ЕС-1033 и других моделях используется устройство, содержащее пишущую машину «Консул-260.1» и блок управления с пультом, или устройство ЕС-7077. Пользуясь клавиатурой пишущей машины, оператор может вводить данные в ЭВМ и одновременно печатать их на бумаге, например, красным цветом; при приеме информации из ЭВМ печать может производиться черным цветом. Устройство может работать как в мультиплексном, так и в монопольном режиме, причем монопольный режим обычно задается каналом, к которому оно подключено. Устройство ЕС-7077 по сравнению с устройством ЕС-7070, в котором используется пишущая машина «Консул-260», имеет улучшенные конструктивно-технологические характеристики и лучшие показатели надежности.

Печатающий механизм последовательного действия типа «Консул-260.1», как и механизм типа «Консул-260», обеспечивает скорость печати до

10 символов в секунду. На строке бумажной ленты (листа бумаги) печатается до 106 символов; количество различных печатаемых символов — 92. Обмен информацией с каналом осуществляется в коде ДКОИ. Любому символу (знаку) пишущей машины отвечает 8-разрядный байт (без учета контрольного разряда), 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.

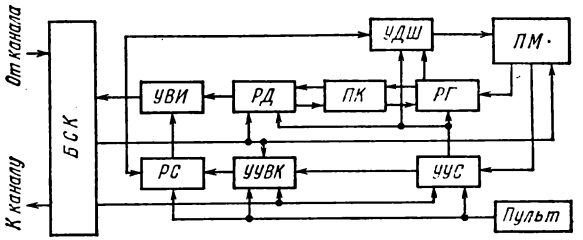


Рис. 12.6. Схема управления пишущей машиной при ее подключении к каналу

Схема управления устройства ЕС-7077 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 РГ и РД, узел дешифратора УДШ, пишущая машина ПМ и пульт оператора, именуемый также пультом инженера.

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.).

Каждый набираемый на клавиатуре пишущей машины символ с помощью комбинатора представляется в коде КОИ-7. Поэтому до передачи в канал коды символов должны быть преобразованы в 8-разрядные комбинации кода ДКОИ. Для такого преобразования используется выделенный на рис. 12.6. преобразователь кодов. 7-разрядные кодовые комбинации поступают в ПК из регистра РГ, куда они вводятся из пишущей машины. Сформированные в ПК 8-разрядные кодовые комбинации поступают в регистр РД. В составе ПК есть и обратный преобразователь из кода ДКОИ в код КОИ-7, а также схема сравнения кодов. При наборе некоторого символа до его печати производятся преобразование отвечающей ему комбинации кода КОИ-7 в комбинацию кода ДКОИ, обратное преобразование и сравнение кодов; печать выполняется только в случае совпадения кодов и с использованием схем узла дешифратора.

При выполнении команд по записи информации, т. е. при приеме данных из машины, комбинации по коду ДКОИ фиксируются в регистре РД. Далее обеспечивается их преобразование в комбинации по коду

КОИ-7, фиксируемые в регистре РГ. Из этого регистра кодовые комбинации подаются на УДШ, что и обеспечивает в конечном итоге печать соответствующих символов на бумажной ленте.

При выполнении команд считывания информации, т. е. при выдаче данных из устройства в канал, комбинации кода КОИ-7 из регистра РГ передаются в преобразователь кодов, на выходах которого образуются комбинации кода ДКОИ. Хранение байт данных в коде ДКОИ обеспечивается регистром РД. Из него байты данных через узел выдачи информации, содержащий группы вентилей и логических схем, передаются в БСК и далее в канал через интерфейс ввода-вывода. 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 РГ в исходное состояние; теперь вместо неправильно набранного можно печатать требуемый символ.

Узел управления синхронизацией по сигналам от ПМ вырабатывает наборы сигналов, обеспечивающие синхронизацию работы электромеханических узлов ПМ с работой электронных узлов блока управления. Сигналы запуска УУС поступают также из блока сопряжения с каналом и пульта оператора, на котором кроме органов управления располагаются элементы световой и звуковой сигнализации.

Устройство ЕС-7077 рассчитано на выполнение команд, приведенных в табл. 12.4, где использованы следующие обозначения: АВК – автоматический возврат каретки, ЗВК – запрет возврата каретки.

Команды ПРОВЕРИТЬ ВВОД-ВЫВОД, УТОЧНИТЬ СОСТОЯНИЕ И ХОЛОСТОЙ ХОД имеют то же значение, что и для устройств, рассмотренных в пп. 12.1–12.3. Выполнение этих команд в устройстве ЕС-7077 не имеет никаких принципиальных особенностей.

Команда УПРАВЛЯТЬ СИГНАЛИЗАЦИЕЙ (УПРАВЛЯТЬ ЗВУКОВОЙ СИГНАЛИЗАЦИЕЙ) принимается устройством даже тогда, когда оно находится в состоянии НЕ ГОТОВО. По этой команде на 3–

Таблица 12.4

Команда	Разряды байта команды								
	К	0	1	2	3	4	5	6	7
ПРОВЕРИТЬ ВВОД-ВЫВОД	1	0	0	0	0	0	0	0	0
УТОЧНИТЬ СОСТОЯНИЕ	0	0	0	0	0	0	1	0	0
ХОЛОСТОЙ ХОД	1	0	0	0	0	0	0	1	1
УПРАВЛЯТЬ СИГНАЛИЗАЦИЕЙ	0	0	0	0	0	1	0	1	1
ЗАПИСАТЬ С ЗВК	0	0	0	0	0	0	0	0	1
ЗАПИСАТЬ С АВК	1	0	0	0	0	1	0	0	1
ЧИТАТЬ	1	0	0	0	0	1	0	1	0

5 с включается звуковая сигнализация на пульте оператора и в последовательности начальной выборки в канал передается байт состояния устройства с указателями КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО.

По командам записи устройство принимает данные из канала и печатает их на бумажной ленте по одному символу. В конце сообщения команда ЗАПИСАТЬ С АВК вызывает автоматический возврат каретки и переход к последующей строке, т. е. перевод бумажной ленты 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 обычно исполняется пробел.

В нормальных условиях команды записи заканчиваются каналом при нулевом состоянии счетчика байт; 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, СБОЙ 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 СБОЙ В УСТРОЙСТВЕ.

Команда ЧИТАТЬ, по которой производится передача информации, набираемой на клавиатуре пишущей машины, 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. При поступлении из канала команды ЧИТАТЬ снимается блокировка с клавиатуры пишущей машины; на пульте включается лампочка транспоранта ЧИТАТЬ, сигнализирующая оператору о том, что можно набирать символы на клавиатуре ПМ. При этом до удовлетворения очередного запроса устройства на передачу информационного байта в канал клавиатура блокируется. Такой порядок работы сохраняется до окончания выполнения команды.

Выполнение команды ЧИТАТЬ заканчивается в следующих трех случаях:

канал прекращает выполнение операции по нулевому состоянию счетчика байт;

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 с клавиатуры ПМ, и оператор может повторно ввести в ЭВМ сообщение.

В любом случае окончания выполнения операции чтения производится блокировка клавиатуры ПМ, включение лампочки транспоранта 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.

Байт основного состояния устройства с пишущей машиной выдается в канал в тех же случаях, что и байт основного состояния других устройств, например печатающих. 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; 1-й и 2-й разряды байта основного состояния не используются; байт всегда имеет контрольный разряд.

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 никаких новых команд (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 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 ВУ КОНЧИЛО формируется при окончании операции, а также при переводе устройства из состояния НЕ ГОТОВО в состояние ГОТОВО. Указатель КАНАЛ КОНЧИЛ показывает каналу, что устройство может быть отключено от интерфейса.

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 клавиши АН; он посылается в канал всегда одновременно с указателем КАНАЛ КОНЧИЛ.

Байт уточненного состояния устройства ЕС-7077 выполняет те же функции, что и в других ранее рассмотренных устройствах. 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; разряды с 4-го по 7-й 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 пишущей машине;

не производится возврат каретки после подачи соответствующего сигнала.

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. Комплекс устройств отображения информации

Для организации эффективного взаимодействия большого числа операторов с ЭВМ используется комплекс устройств отображения информации, или комплекс ЕС-7920, являющийся дальнейшим развитием комплекса ЕС-7906 [19]. Такой комплекс имеет несколько модификаций, различающихся не только своими наборами средств ввода-вывода, но и принципами подключения к каналам ввода-вывода. В комплекс любой модификации всегда входит групповое устройство управления и устройство отображения (алфавитно-цифровой дисплей, выполненный на основе использования электронно-лучевой трубки) с дополнительной клавиатурой; к каждому устройству управления может быть подключено до 32 средств ввода-вывода вида устройств отображения или специальных печатающих устройств. Комплексы ЕС-7920 могут подключаться как к мультиплексным, так и к селекторным каналам машин Единой системы.

Комплексы первых двух модификаций, т. е. ЕС-7920-00 и ЕС-7920-01, содержат устройства управления ЕС-7922-00 и ЕС-7922-01 соответственно, подключаемые через интерфейс ввода-вывода (ИВВ) непосредственно к каналу (рис. 12.7); такие устройства управления называют еще локальными. Комплексы ЕС-7920-10 и ЕС-7920-11 содержат так называемые дистанционные устройства управления ЕС-7921-00 и ЕС-7921-01 соответственно, которые подключаются к каналам через модемы М, телефонные или специальные каналы связи КС, мультиплексоры передачи данных МПД и интерфейсы ввода-вывода, как это показано на рис. 12.7. Средства ввода-вывода информации, или периферийные устройства ПФУ, могут располагаться от устройств управления на расстояниях до 1200 м.

Устройства управления локального типа, как и устройства управления дистанционного типа, отличаются главным образом емкостью буферной

памяти, в которой можно размещать коды либо 480 символов, либо 1920 символов. В соответствии с этим к устройствам управления подключаются ПФУ различных возможностей; устройства управления с меньшей емкостью буферной памяти имеют код 00, а с большей — код 01. При обмене информацией устройств управления с ПФУ используется код КОИ-8; скорость обмена — до 864 кбит/с. Локальные устройства управления осуществляют обмен информацией с каналами в коде ДКОИ со скоростью порядка 250 кбайт/с. Дистанционные устройства управления

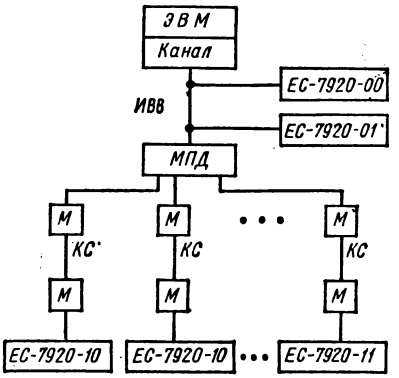


Рис. 12.7. Схема подключения комплексов отображения информации к ЭВМ

информацию, направляемую в каналы связи, кодируют по коду КОИ-7; скорости обмена информацией — 600, 1200, 2400 или 4800 бит/с, что определяется характеристиками используемых каналов связи. Устройства отображения (УО) являются техническими средствами, включаемыми в состав комплексов любых конфигураций. Само по себе устройство отображения представляет устройство вывода с индикацией информации на экране электронно-лучевой трубки. Могут быть использованы либо, устройство ЕС-7927-00, подключаемое к любым устройствам управления либо устройство ЕС-7927-01, подключаемое только к ЕС-7922-01 или к ЕС-7921-01; первое из них обеспечивает одновременную индикацию до 480 символов (12 строк по 40 знаков), а второе — до 1920 символов (24 строки по 80 знаков). Для всех устройств отображения принята единая номенклатура алфавитно-цифровых и специальных символов; всего используется 96 различных символов. Все символы отображаются по точечному методу; размер базовых матриц — 7×9 точек; схемы управления обеспечивают отображение символов без мерцания.

По управляющим сигналам из группового УУ устройство отображения выполняет:

запись всей информации, отображаемой на экране, из УУ в собственную буферную память и хранение этой информации;

считывание всей информации из буферной памяти в УУ;

передачу в УУ по запросу информации о состоянии устройства;

запоминание формата изображения, полученного из УУ;

отображение символов по трем градациям яркости, причем одна из них отвечает практически неотображаемой информации;

регламентацию возможностей обработки оператором различных категорий информации.

При подключении к устройству отображения клавиатуры оно становится средством не только вывода, но и ввода информации. Под управлением клавиатуры обеспечиваются:

занесение любого отображаемого символа в позицию экрана, указанную специальным указателем – курсором (чертой под знакоместом символа);

перемещение курсора в любую позицию экрана;

стирание всей или части информации, индицируемой на экране;

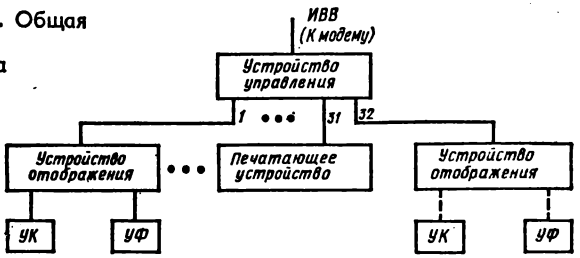
вставку информации в текст со сдвигом остатка вправо;

вычеркивание информации из текста со сдвигом остатка влево;

формирование запроса на прерывание в УУ и кода условия прерывания.

В составе комплексов ЕС-7920 могут быть использованы четыре различных устройства клавиатуры УК, подключаемых только к устройствам отображения, как это показано на схеме рис. 12.8. Устройство ЕС-0101-1-01 имеет клавиатуру типа пульта оператора, содержащую прописные и строчные буквы русского и латинского алфавита, а также математические знаки. Устройство ЕС-0101-1-02 имеет клавиатуру типа клавиатур устройств подготовки данных, позволяющую набирать буквы русского

Рис. 12.8. Общая схема комплекса ЕС-7920



и латинского алфавитов, математические знаки. Устройство ЕС-0101-1-03 имеет клавиатуру пишущей машины, позволяющую набирать буквы русского алфавита и математические знаки. Устройство ЕС-0101-1-04 имеет клавиатуру типа универсальной с латинским алфавитом. Все типы клавиатур оснащены электрической блокировкой клавиш и звуковой сигнализацией для оповещения оператора о срабатывании нажатой клавиши. Выбор типа клавиатуры всегда согласуется с основным назначением соответствующего устройства отображения.

Кроме УК к любому устройству отображения может подключаться устройство фотоселектора (УФ), называемое также просто фотоселектором. Фотоселектор представляет собой малогабаритный фоточувствительный приемник, который может воспринимать сигналы от отображаемых на экране УО знаков. Определение информации фотоселектором происходит только при его подводе вплотную к соответствующему участку экрана.

Печатающие устройства используются в качестве средств вывода из ЭВМ информации, по которой необходимо иметь отпечатанный документ. Возможно использование печатающих устройств ЕС-7934-00, подключаемых к любым групповым устройствам управления, и ЕС-7934-01, подключаемых только к ЕС-7922-01 или к ЕС-7921-01. Печатающие устройства различаются только емкостью внутренней буферной памяти; остальные их характеристики идентичны. Для них принят игольчатый матричный принцип печати при скорости печати не менее 66 символов в секунду. Номенклатура и графика символов – как и в устройствах

отображения. При печати в одной строке размещается 132 символа; число печатаемых копий – до трех.

Комплексы ЕС-7920 могут быть использованы для построения запросных (справочно-информационных) систем, систем сбора и подготовки данных, а также для выделения дополнительных пультов операторов. Если они применяются как основные технические средства запросных систем, то основные действия выполняются с помощью устройств отображения, которые обеспечивают быстрый доступ к массивам информации, хранящимся в памяти ЭВМ.

При использовании комплексов ЕС-7920 для сбора и подготовки данных устройства отображения заменяют практически устройства ввода с перфоносителей; при этом обеспечивается как обновление массива, так и занесение данных в режиме непосредственного подключения. Операция по обновлению исходного массива начинается запросом, необходимым для доступа к данным, подлежащим изменению. Затем эти данные обновляются оператором устройства отображения и поступают в машину. Возможен также и непосредственный ввод данных в исходный массив. При занесении в режиме непосредственного подключения данные, вводимые с клавиатуры, объединяются и заносятся в память машины, обеспечивающую быстрый доступ к информации; обычно это память на магнитных дисках. Программа обработки далее обращается к этому массиву, что освобождает ЭВМ от необходимости выполнения вспомогательных работ, связанных с обработкой данных, и обеспечивает повышение эффективности решения соответствующих задач.

Для всех случаев использования рассматриваемых комплексов характерным является то, что обмен информацией с ЭВМ начинается по инициативе оператора, работающего на устройстве отображения. Обычно работа происходит следующим образом. Оператор УО при помощи клавиатуры заносит на экран необходимую информацию, т. е. запрос на получение справки или исходного массива, информацию для ввода данных. Эта информация подготовляется оператором автономно, без связи с ЭВМ (соответствующей программой). Действия оператора по подготовке и редактированию информации, индицируемой на экране, происходят также без участия группового УУ благодаря наличию в устройстве буферной памяти. После подготовки информации на экране оператор вызывает прерывание при помощи одной из клавиш программного прерывания, имеющихся в составе клавиатуры устройства отображения.

В результате реализации прерывания в машину передается информация, подготовленная оператором. Затем происходит необходимый обмен данными между оператором и ЭВМ; данные выводятся на экран УО и считываются из УО по командам выполняемой программы. Команды и цепочки команд обычно инициируются по прерываниям от УО. По требованию оператора информация из машины может выводиться не только на индикацию, но и на печать при использовании одного из печатающих устройств, подключенных к групповому устройству управления.

Обмен информацией между оператором УО и машиной возможен как при неформатизованном, так и при форматизованном экране. При работе с неформатизованным экраном оператор использует его произвольно, т. е. имеет доступ к любому его участку. Все участки отображаемой

информации равнозначны, поэтому при выполнении команд считывания все символы передаются подряд без пропусков в порядке возрастания адресов знакомест.

При работе с форматизованным экраном последний приводится к определенному формату реализуемой в ЭВМ программой. Информация подразделяется на категории, причем данные, имеющие одинаковые характеристики, группируются в пределах одного поля экрана. Характеристика поля (соответствующих данных) определяется первым символом этого поля, называемого атрибутом поля. Символы, начиная с атрибута некоторого поля, до атрибута следующего поля, представляют информацию, принадлежащую этому полю.

Информация, разбитая на поля, называется форматизованной информацией, а экран устройства отображения, содержащий такую информацию, — форматизованным экраном. Символ-атрибут, определяющий все характеристики поля, задается реализуемой программой. Он никогда не индицируется на экране и недоступен оператору для его изменения. Итак, формат может быть задан только программным путем.

Символы-атрибуты хранятся в буферной памяти УО, т. е. занимают ее определенные ячейки. Поскольку они не индицируются, то соответствующие знакоместа на экране воспринимаются оператором как пробел. Основными характеристиками полей, которые они определяют, являются:

доступность для модификации оператором;
доступность для определения фотоселектором;
уровень яркости;
выделение поля как цифрового или алфавитно-цифрового.

Первая из указанных характеристик показывает, может ли оператор изменять информацию поля с помощью клавиатуры. По уровням яркости поле может быть определено как неотображаемое, как отображаемое с нормальной яркостью и как отображаемое с повышенной яркостью. Если поле выделено (определено) как цифровое, то в него, если оно доступно для оператора, можно заносить с клавиатуры только цифры и некоторые специальные знаки. В незащищенное алфавитно-цифровое поле можно заносить с клавиатуры любые символы.

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 типа 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 подключения УВУ с наивысшим приоритетом. Оно образует дополнительный, или вспомогательный, интерфейс, как это показано на рис. 12.9. В процессе работы устройство ЕС-4080 ре-

транслирует 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 быть использовано также для увеличения длины линий интерфейса до 100—120 м.

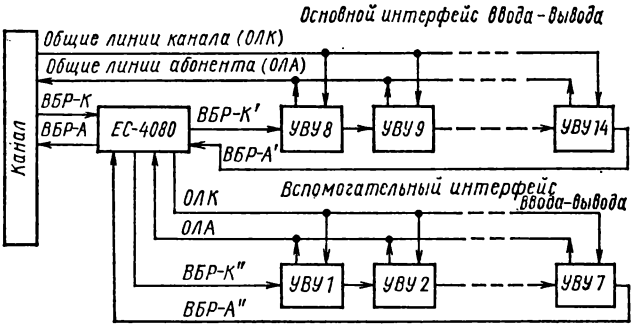


Рис. 12.9. Расширение возможностей интерфейса ввода-вывода при использовании логического ретранслятора

Если логический ретранслятор ЕС-4080 подсоединяется к стандартному интерфейсу ввода-вывода на месте подключения 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 обеспечиваются:

работа канала с устройствами управления внешними устройствами (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 общие 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).

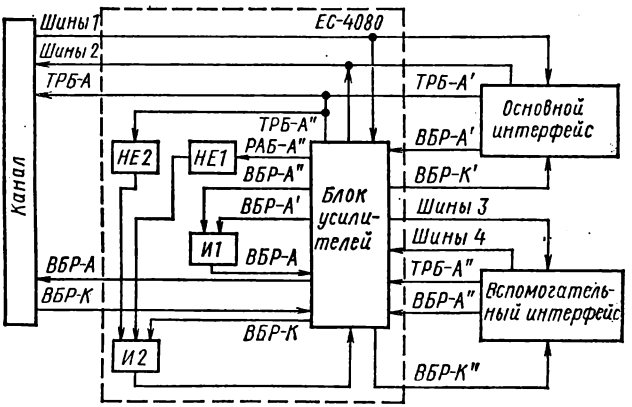


Рис. 12.10. Организация логического ретранслятора

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.

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, подключенных 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 обслуживает 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 НЕ1; 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 поколонную 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.

В блок считывания входит 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 второй 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 перфолентах. Нанесение информации на перфоленту 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 подготовленных лент, а также реперфорацию лент. Общий состав устройства показан на схеме (рис. 12.11).

Набор символов, коды которых набиваются на перфоленте, осуще-

ствляется 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.

В блок считывания входят два лентопротяжных механизма и соответственно две схемы фотосчитывания информации с перфолент (ПЛ1 и ПЛ2). При работе с двумя перфолентами обеспечивается их синхронная протяжка под фотодиодными узлами. Информация, считываемая с перфолент, направляется в блок контроля; при реперфорации лент необходимая информация из блока считывания в ПЛ-80 направляется через блоки контроля БК и управления БУ.

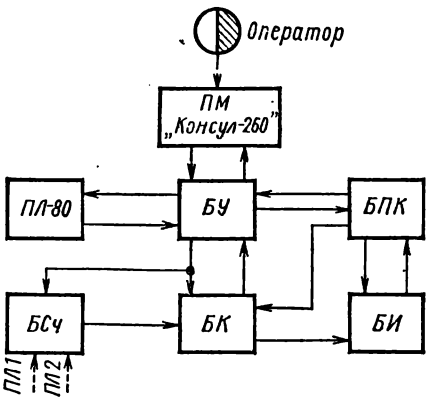


Рис. 12.11. Схема устройства подготовки данных на перфолентах

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 содержимого 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 перфоленты.

Блок управления устройства формирует управляющие сигналы и организует работу всех остальных частей устройства в различных 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 перфоленту и печатается на бланке;

режим реперфорации; по данным, считываемым 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 перфоленту 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.

Глава 13

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 двусторонние 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 УВУ (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 осуществляется 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 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 монополизирует 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 медленнодействующими 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.

Кроме мультиплексного и селекторных каналов, в некоторых машинах, например ЕС-1033, выделяют так называемый общий канал, одной из основных функций которого является организация приоритетного обращения к оперативной памяти со стороны каналов. При наличии общего канала в значительной степени упрощается блок управления оперативной памятью, так как заявки на обслуживание поступают в него только от двух абонентов — процессора и общего канала; в определенной степени упрощаются также мультиплексный и селекторные каналы. Более подробно общий канал рассмотрен в п. 13.5.

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:

$$\text{КВВ} \rightarrow \text{АСК} \rightarrow \text{УСК},$$

где КВВ — 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-1052), восьмиразрядный адрес внешнего устройства и адрес канала или по отдельной радиальной линии сигнал выборки данного канала (для каналов моделей ЕС-1050 и ЕС-1052). 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.

Рис. 13.1. Адресное слово канала

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 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.

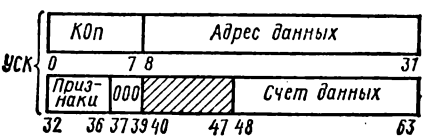


Рис. 13.2. Управляющее слово канала

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.

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 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.

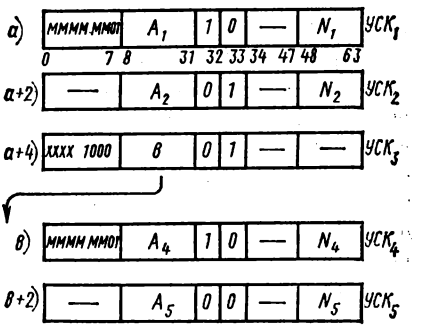


Рис. 13.3. Канальная программа

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 $УСК_1$, выбираемое по адресу *а)*, заданному в АСК; это же УСК начинает цепочку управляющих слов. $УСК_1$ и $УСК_2$ составляют цепочку данных, так как в 32-м разряде $УСК_1$ содержится единица. При выполнении операции записи (код ММММ ММ01) из области памяти, начинающейся ячейкой A_1, в заданное ВУ передается группа из N_1 байт; после этого осуществляется передача N_2 байт из ячеек A_2, A_2+1 и т. д. Переход к следующей группе УСК осуществляется по $УСК_3$, содержащему код операции перехода. В этом же $УСК_3$ содержится адрес *в)*, по которому выбирается $УСК_4$. Последнее УСК, т. е. $УСК_5$, имеет нулевые признаки цепочек, поэтому по выполнении этого УСК заканчивается операция записи, заданная инструкцией НАЧАТЬ ВВОД-ВЫВОД.

Из изложенного следует, что операции ввода-вывода заканчиваются автоматически, если просмотрена вся цепочка управляющих слов; при этом просмотр заканчивается тогда, когда при реализации последнего 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.

* Здесь и далее обозначение [ ] = 0 указывает, что в соответствующих разрядах содержатся нули.

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) и 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; он фиксируется в 34-м и 35-м разрядах ССП.

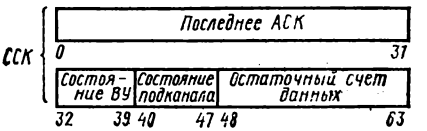


Рис. 13.4. Слово состояния канала

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).

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 понимают 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 мультиплексно-

го 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 (ПК) 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 следующие обозначения: РКЗ — регистр ключа защиты, РСАК — регистр адресов команд (регистр-счетчик адресов команд), РКО — регистр кода операций, РСАД — регистр адреса данных (регистр-счетчик адресов данных), РСБ — регистр счета байт, РП — регистр признаков (указателей, флажков), РСД — регистр счета данных, ДШКО — дешифратор кода операций, РКВВ — регистр команд ввода-вывода, РАК — регистр адреса канала, РПР — регистр прерываний, РАП — 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 подканала, 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 РСАК. По содержимому РСАК производится выборка УСК, отвечающего исполняемой инструкции НАЧАТЬ ВВОД-ВЫВОД, в РКО, РСАД, РСБ, РП и РСД. Так как

возможно выполнение цепочки команд, то после выборки начального УСК содержимое РСАК увеличивается на восемь 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.

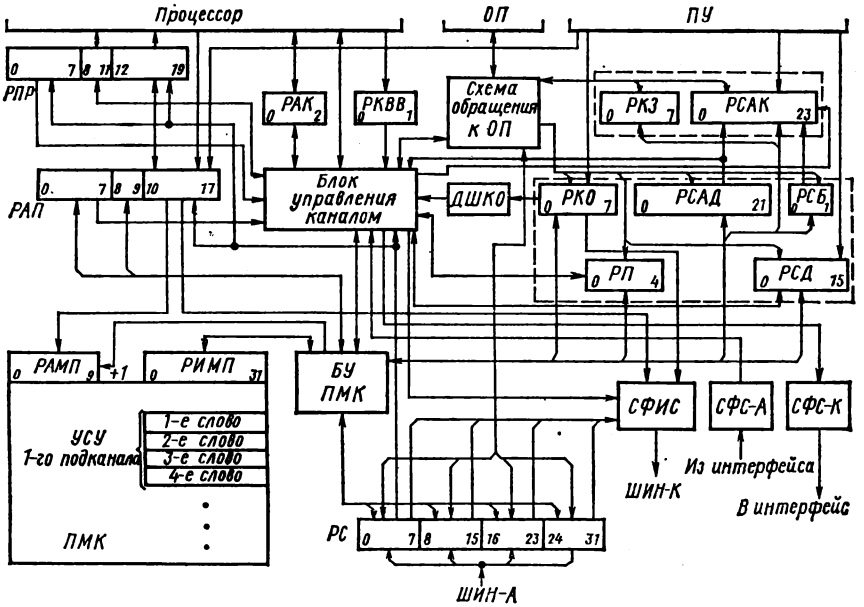


Рис. 13.5. Общая схема мультиплексного канала

• 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, содержащейся 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 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 канальной 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 подключилось к каналу 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 четыре соседние 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 вентилей. 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, в который входят четыре последовательные ячейки, для размещения управляющего слова устройства, иллюстрируется данными, приведенными в табл. 13.1.

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 вентилей; через группы входных вентилей осуществляется прием байт данных из ШИН-А и их распределение по частям регистра; через группы выходных вентилей осуществляется выдача байт данных через СФИС в ШИН-К. Часть регистра РС[0–7] имеет дополнительные группы выходных вентилей

Таблица 13.1

Размещение УСУ

Части УСУ (ячейки ПМК)	Разряды	Содержимое
1	[0–7]	РКЗ[0–7] – ключ защиты ОП машины
	[8–31]	РСАК[0–23] – адрес УСК в ОП
2	[0–4]	РП[0–4] – признаки в УСК
	[5]	Не используется
	[6–7]	РАП[8–9] – режим подканала
	[8–15]	РАП[0–7] – состояние подканала (ВУ)
	[16–31]	РСД[0–15] – счет байт данных
3	[0–7]	РКО[0–7] – код операции
	[8–29]	РСАД[0–21] – адрес данных в ОП
	[30–31]	РСБ[0–1] – счет байт данных
4	[0–31]	РС[0–31] – данные (слово данных)

для выдачи адресов ВУ и байт их состояний в РПР, РАП и блок управления каналом. Посредством полноразрядных групп входных и выходных вентилей регистр РС связан с ПМК через ее блок управления и со схемой обращения к ОП. Эти группы вентилей 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.

Режим 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 вводо-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.

Таблица 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	ССК записано	ССК сформировано	ССК записано
10	Канал или подканал занят	Канал или подканал занят	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 монопольно занимает оборудование канала, то для реализации рассматриваемой команды его адреса 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, содержащее это новое слово данных, запоминается в ПМК, где хранится до очередного сеанса связи с внешним устройством. Передача каждого байта через интерфейс уменьшает на единицу содержимое РСД и увеличивает на единицу содержимое РСБ. 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 $B_0B_1B_2B_3$. При

очередном сеансе связи с ВУ это слово помещается в РС. Принимаемый от ВУ байт данных B'_2 заменяет байт B_2 в исходном слове, которое принимает вид $B_0B_1B'_2B_3$. 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 $B_0B_1B'_2B'_3$. 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 предпоследнем 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 характеристики 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 работе [7].

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 – контрольных), а с внешними устройствами – побайтно (девять 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 девять 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 девять 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 байт используется девять шин, кода защиты памяти – пять шин, сигналов признаков и запросов – пять шин.

Селекторный канал включает в свой состав пять основных блоков (рис. 13.6):

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.

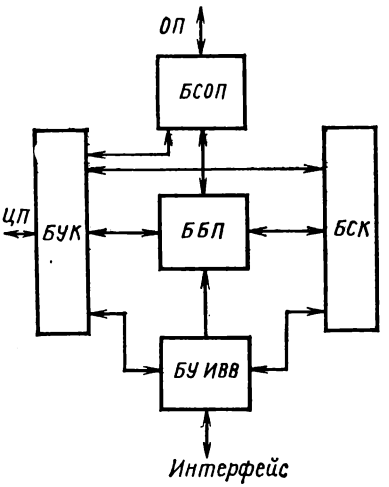


Рис. 13.6. Общая схема селекторного канала

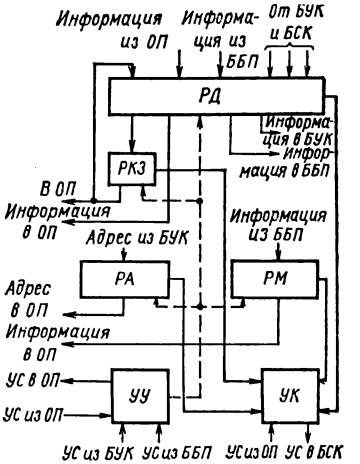
Блок БСК предназначен для выработки, фиксации и обработки всех возможных условий состояния канала и реализации совместно с БУК и БСОП прерываний. Блок БУК обеспечивает централизованное управление каналом при любых режимах работы; он непосредственно связан с ЦП и со всеми остальными блоками канала. Блоки 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 именуются управляющими сигналами (УС). При каждом обращении к ОП в БСОП формируется запросное слово, содержащее адрес, ключ защиты памяти, признак режима работы, маркеры и данные, если осуществляется запись. Запросное слово сопровождается 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.

Рис. 13.7. Блок сопряжения с оперативной памятью



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.

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, направляемых 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.

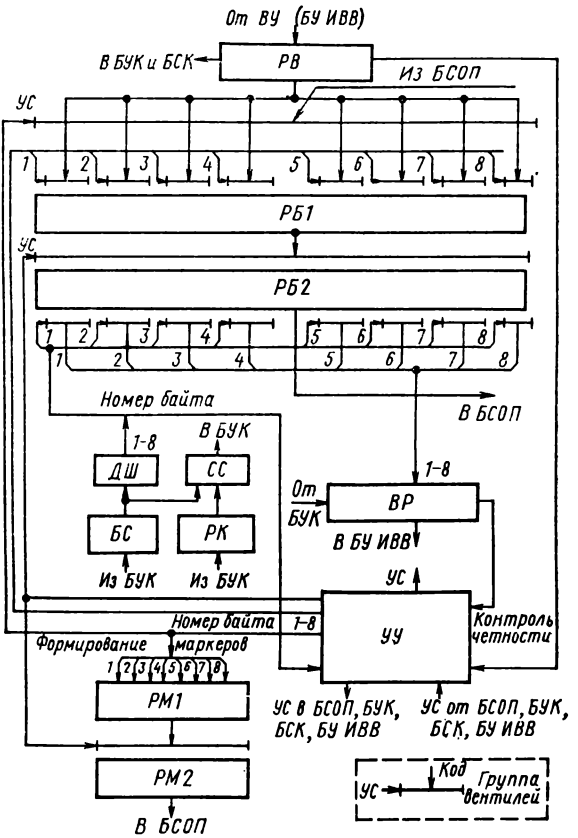


Рис. 13.8. Блок буферной памяти

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 обмене типа «память – внешнее устройство» двойное слово, размещенное в РБ2, с помощью восьми групп выходных вентилей, 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 содержимого счетчика производится посредством дешифратора ДШ, который на своих выходах формирует сигналы, управляющие работой выходных вентилей регистра РБ2; эти же сигналы используются в узле управления при формировании УС, управляющих работой входных вентилей 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 вентилей, управляемых сигналами, поступающими из узла управления.

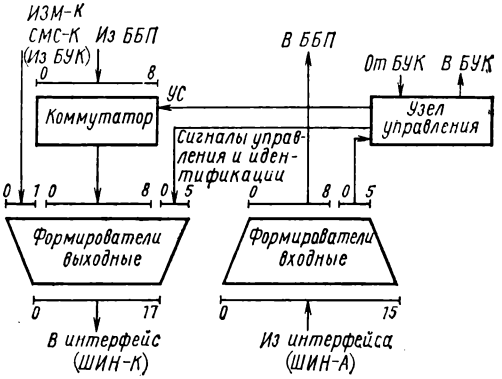


Рис. 13.9. Блок управления интерфейсом

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.

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 РА1 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 РА1 из регистра сумматора, где адрес данных изменяется на ± 8 единиц после каждого обращения к памяти по записи или считыванию данных, которые передаются двойными словами. При обращении к ОП содержимое РА1 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 → ВУ» или «ВУ → ОП», из регистра РА1 передаются на буферный счетчик блока ББП и в сумматор СМ для модификации счета в регистре РС8.

Первый регистр адреса УСК, или регистр РА2, служит для хранения адреса УСК; он имеет 21 информационный и 3 контрольных разряда. Адрес УСК вводится в регистр РА2 из блока БСОП. Возможно

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.

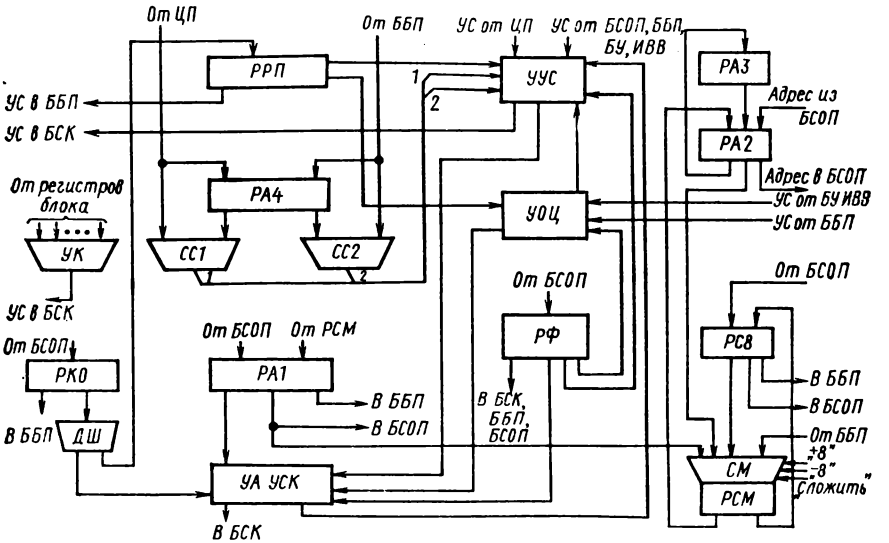


Рис. 13.10. Блок управления каналом

Второй регистр адреса УСК, или регистр РА3, предназначен для промежуточного хранения адреса УСК; он 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 непереданных 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 сбоев в одном из его блоков. Состояние Р означает, что канал работает, т. е. занят выполнением 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; производится отключение от процессора. 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. отвечает 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 производятся 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 его работы, 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 обычных подканалов, 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 обслужить 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 зависимости 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.

Таблица 13.3

Пропускная способность канала

Загрузка подканалов, кбайт/с					V, 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

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 составляться 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, формирование управляющих слов устройств (УСУ), передачи информации от оперативной памяти к ВУ через подканалы и от ВУ в оперативную память, обслуживания запросов на прерывания и установле-

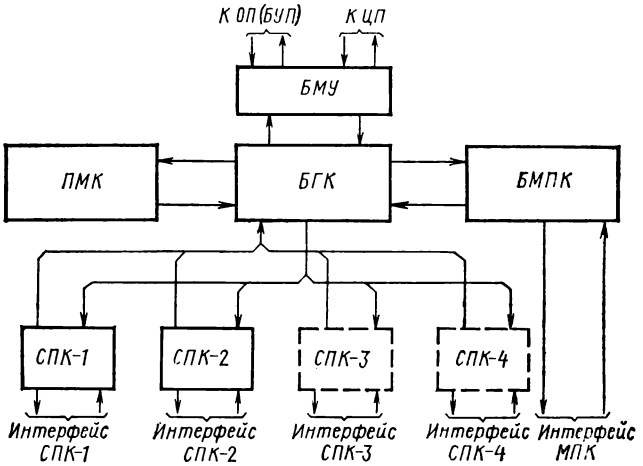


Рис. 13.11. Мультиплексный канал с селекторными подканалами

ния 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 содержит 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 $T_{выб} \leqslant 0{,}5$ 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 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 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 в 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 отсылает 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 выборка 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.

13.5. Каналы магистральной структуры

В каналах, как отмечалось выше, кроме других действий всегда осуществляется преобразование информации, включая выполнение операций сложения и вычитания. Поэтому структура их операционных частей близка к структуре операционных частей процессоров и может базироваться на принятых для всей машины общих принципах. Применительно к машине ЕС-1033 это выражается в широком использовании магистральных связей между отдельными функциональными узлами и блоками; соответственно каналы, как и процессор, имеют магистральную структуру. При этом кроме мультиплексного и селекторных каналов используется еще общий канал ОК, координирующий работу МК и СК при их взаимодействии с внешними устройствами, оперативной памятью и процессором.

Общая схема каналов, используемых в машине ЕС-1033, и их основные связи показаны на рис. 13.12. В состав комплекта всегда входят общий и мультиплексный каналы; селекторных каналов может быть как два, так и три. Функции мультиплексного и селекторных каналов заключаются, как и во всех других случаях, в обеспечении управления выполнением операций ввода-вывода. Однако выполнение некоторой части этих функций,

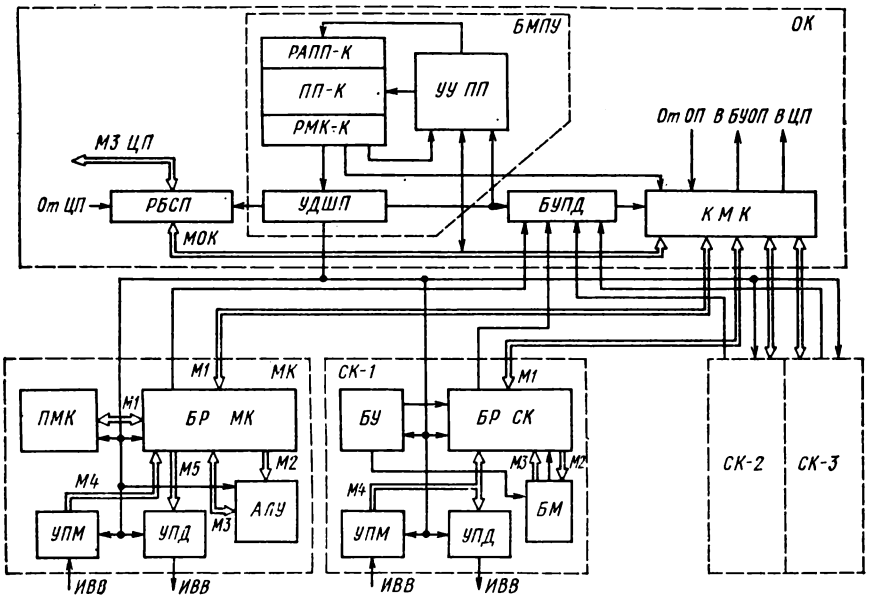


Рис. 13.12. Общая схема каналов магистральной структуры

особенно по обеспечению связей с процессором и оперативной памятью, передано общему каналу, который к тому же частично выполняет функции блока управления оперативной памятью, обеспечивая предварительное приоритетное обслуживание заявок на обмен информацией с оперативной памятью, поступающих от МК и СК. Мультиплексный канал обеспечивает скорость передачи данных до 50 кбайт/с в мультиплексном и до 300 кбайт/с в монопольном режимах при последовательной работе внешних устройств. Любой из селекторных каналов обеспечивает скорость передачи данных до 800 кбайт/с; при параллельной работе СК скорость передачи данных снижается и определяется временными характеристиками оперативной памяти.

Общий канал. Занимая промежуточное положение между оперативной памятью, процессором и мультиплексным и селекторными каналами, общий канал обеспечивает прием и исполнение инструкций процессора, подключение и отключение УВУ, передачу данных из ОП в каналы и из каналов в ОП, управление работой каналов, а также выполнение некоторых дополнительных действий. Его основными функциональными

частями, показанными на рис. 13.12, являются регистр буферной связи с процессором РБСП, блок управления передачей данных БУПД, коммутатор каналов КМК и общий блок микропрограммного управления БМПУ, в составе которого выделены постоянная память каналов ПП-К с регистром адреса РАПП-К, регистром микрокоманд РМК-К и узлом управления УУ ПП, а также узел дешифраторов полей микрокоманд УДШП.

Блок микропрограммного управления предназначен для формирования управляющих сигналов, координирующих работу всех каналов ввода-вывода машины в различных режимах; исключение составляет передача данных в селекторных каналах, которая осуществляется по сигналам, поступающим от местных тактовых цепочек. Постоянная память блока имеет емкость 2048 слов разрядности, принятой для микрокоманд каналов. При 64 основных двоичных разрядах каждая микрокоманда содержит два контрольных разряда. Таким образом, РАПП-К является одиннадцатиразрядным, а РМК-К — 66-разрядным. Микрокоманды разбиваются на 12 полей, включая поле, содержащее базовый адрес следующей микрокоманды. Принципы формирования исполнительного адреса следующей микрокоманды практически те же, что и для блоков микропрограммного управления процессоров. Адреса микрокоманд, а также необходимые управляющие сигналы формируются в узле управления постоянной памятью.

Из двенадцати полей микрокоманд семь непосредственно используются для формирования управляющих сигналов, координирующих работу каналов. Каждому из этих полей в УДШП соответствует отдельный дешифратор. Приемниками управляющих сигналов, формируемых на выходах дешифраторов, служат узлы общего, мультиплексного и селекторных каналов. Сигналы в МК и СК подаются параллельно, но воспринимаются только одним каналом, которому отвечает реализуемая в данный момент времени микропрограмма. Выборка работающего канала в соответствии с его номером осуществляется при действии сигнала по одной из селективных линий РАБ МК, РАБ СК-1, РАБ СК-2, РАБ СК-3. Прием сигналов от УДШП, предназначенных для узлов общего канала или только для мультиплексного канала, селективными линиями не управляется, т. е. обеспечивается подача этих сигналов непосредственно в управляемые цепи.

Регистр буферной связи с процессором служит для приема информации из центрального процессора ЦП и выдачи информации в ЦП, осуществляемых по управляющим сигналам, поступающим от ЦП, а также приема из регистров общего канала и передачи в них информации, осуществляемых по управляющим сигналам, поступающим от БМПУ. Обмен информацией с блоками процессора производится по магистрали МЗ ЦП, а с другими функциональными частями ОК — по центральной магистрали общего канала МОК. По своей структуре рассматриваемый регистр является обычным четырехбайтным регистром параллельного действия с несколькими управляемыми входами и выходами на каждый разряд. Из процессора в РБСП поступают код инструкции ввода-вывода, номер канала и адрес устройства, а по МОК — информация, связанная с окончанием операции по вводу-выводу.

Коммутатор каналов служит для организации потоков информации между оперативной памятью и работающими каналами. Через него передается числовая информация из каналов в ОП (БУОП) и общий канал, а также в каналы из ОП и из поля констант микрокоманд, находящихся в РМК-К. Основу коммутатора составляют вентильные схемы, коммутирующие магистральные линии каналов и оперативной памяти; они обеспечивают передачу четырехбайтных слов параллельными кодами. Непосредственное управление вентильными схемами обеспечивается в соответствии с номером канала, поступающим из блока управления передачи данных, а также сигналами, поступающими из БМПУ.

Блок управления передачей данных служит для формирования управляющей информации, выражаемой в основном управляющими сигналами, при обмене данными между ОП и каналами. В его состав входят приоритетная схема обращения к ОП, а также схемы синхронизации и управления потоками информации. Все запросы от каналов к ОП обслуживаются последовательно в соответствии с присвоенными приоритетами; при этом наивысшим приоритетом пользуется запрос за данными от СК-2; запросы за УСК от селекторных каналов имеют меньший приоритет по сравнению с запросами за данными, причем приоритетность между СК распределяется в соответствии с их номерами; низший приоритет имеют запросы от мультиплексного канала.

Мультиплексный канал. Основной функцией мультиплексного канала магистральной структуры, как и любого другого мультиплексного канала, является организация передач информации между оперативной памятью машины и медленнодействующими внешними устройствами при выполнении операций ввода-вывода. Этот канал может работать как в мультиплексном, так и в монопольном режиме, причем любая операция ввода-вывода может быть выполнена частично в мультиплексном режиме, а частично в монопольном режиме. Если он не занят обслуживанием какого-либо внешнего устройства, то организуется последовательный опрос подключенных к нему ВУ для выявления возможных запросов на передачу данных или прерывание, т. е. наличия сигналов ТРБ-А. Основными функциональными частями мультиплексного канала, показанными на рис. 13.12, считаются мультиплексная память, или память мультиплексного канала ПМК, блок регистров БР МК, арифметико-логическое устройство АЛУ, усилители-передатчики УПД и усилители-приемники УПМ; можно считать, что АЛУ и блок регистров составляют операционную часть мультиплексного канала. Все функциональные части МК связаны между собой магистралями М1, М2, М3, М4 и М5.

Все магистрали мультиплексного канала построены так же, как и магистрали процессора (см. п. 8.4); для кратковременного хранения информации их каждый разряд имеет запоминающий элемент; совокупность запоминающих элементов данной магистрали составляет соответствующий магистральный регистр. Магистраль М1 является основной информационной магистралью, связывающей мультиплексную память, блок регистров и канальный коммутатор общего канала; по ней передаются полные четырехбайтные слова и другая необходимая информация. Магистрали М2 и М3 связывают блок регистров и АЛУ; по ним передаются исходные и модифицированные адреса, а также коды, отвечающие

счету переданных между ОП и ВУ байт информации. Поэтому М2 и М3 выполнены как трехбайтные. По магистрали М4 в блок регистров передаются от УПМ принятые из интерфейса ввода-вывода ИВВ байты, содержащие передаваемую от ВУ информацию; по магистрали М5 осуществляется выдача через УПД во внешние устройства соответствующих байт информации. Во всех магистралях на каждый байт всегда предусматривается линия для контрольного разряда.

Блок регистров служит для хранения всей информации, необходимой для обеспечения функционирования активного подканала, отвечающего ВУ, обменивающемуся в данный момент времени информацией с оперативной памятью. В его состав входят регистры для хранения частей АСК и УСК, магистральные регистры, регистры, объединяющие отдельные триггеры, служащие для формирования и хранения значений интерфейсных и некоторых служебных сигналов, регистры для хранения адреса (номера) работающего ВУ, кода состояния ВУ, адреса ВУ, требующего прерывания, а также регистр данных, называемый также регистром коммутации данных, или регистром информационного слова.

Все регистры выполнены на микросхемах 155 ХЛ1, содержащих по два триггера с тремя управляемыми входами и тремя управляемыми выходами каждый; таким образом, каждый регистр по приему и выдаче информации может быть подключен к трем магистралям. Все регистры, как правило, имеют разрядность, кратную байту; если основная информация некоторого регистра имеет меньше восьми разрядов, то в его состав включают отдельные служебные триггеры. Для хранения частей АСК используются однобайтный регистр ключей защиты РКЗ и трехбайтный регистр адреса УСК – РАУСК; части УСК располагаются в однобайтном регистре кода операции РКОП, трехбайтном регистре адреса данных РАД, однобайтных регистре флажков РФ и регистре состояния канала РСК, а также в двухбайтном регистре количества байт РКБ. Регистр состояния устройства РСУ, регистр адреса устройства РАУ и регистр адреса прерывания являются однобайтными. Разрядность магистральных регистров отвечает разрядности соответствующих магистралей, а регистр коммутации данных РКД выполнен как четырехбайтный, что отвечает формату слов, принятому для оперативной памяти машины.

В регистрах рассматриваемого блока может производиться только смена хранимой информации; кроме того, входные и выходные связи РКД позволяют производить как побайтный, так и пословный ввод-вывод информации. Все изменения информации производятся только в АЛУ канала, с которым блок регистров связан магистралями М2 и М3. Общая структура связей регистров блока такова: с магистралью М1 связаны РКЗ, РАУСК, РКОП, РАД, РФ, РСК, РКБ, РСУ, РАУ, РКД, РАП, с магистралью М2 – РАУСК, РАД, РКБ, с магистралью М3 – РАУСК, РАД, РКБ, с магистралью М4 – РКЗ, РСУ, РКД, РАУ, РАП, с магистралью М5 – РКЗ, РСУ, РКД, РАУ и РАП. Ввод информации в регистры из магистралей и выдача информации из них в магистрали производятся по управляющим сигналам, формируемым узлами, непосредственно связанными с БМПУ общего канала.

Двадцатичетырехразрядное АЛУ предназначено для модификации адресов оперативной памяти, хранящихся в РАУСК и РАД, количества

байт, которые еще предстоит передать при обмене между ОП и ВУ по коду, содержащемуся в РКБ, и для анализа на нуль результата выполнения операции, что необходимо для окончания операции ввода-вывода по данному УСК.

Арифметическо-логическое устройство управляется микропрограммно от БМПУ общего канала. В качестве первого операнда (A) всегда используется содержимое соответствующего регистра, а в качестве второго операнда (B) — код, заданный в поле констант микрокоманды, находящейся в РМК-К. В АЛУ может быть реализована любая из следующих четырех функций:

$$F_1 = A;\ F_2 = A - B - 1;\ F_3 = A + B;\ F_4 = A - 1.$$

Равенство $B = 3$ справедливо в случае модификации содержимого РАД при выполнении операции обратного чтения; при модификации содержимого РАД в процессе выполнения операций записи и прямого чтения, а также в случае модификации содержимого РАУСК операнд $B = 4$; при модификации содержимого РАУСК, если в байте состояния устройства имеется признак МОДИФИКАТОР СОСТОЯНИЯ, операнд $B = 8$. Очевидно, что реализация F_1 не приводит к изменению содержимого соответствующего регистра, а F_4 используется для счета передаваемых байт данных.

Память мультиплексного канала служит, как и в других случаях, для хранения управляющих слов устройств УСУ, обменивающихся данными с оперативной памятью машины в мультиплексном режиме; каждое УСУ составляется из четырех 32-разрядных слов, не считая контрольных разрядов. Для минимального комплекта машины емкость ПМК отвечает 32 подканалам, т.е. ее накопитель имеет 128 четырехбайтных ячеек; возможно расширение емкости ПМК до 1024 ячеек, что отвечает 256 подканалам. Накопитель ПМК выполнен на микросхемах 155РУ2, каждая из которых обеспечивает хранение 64 бит информации.

Для хранения каждого УСУ отводится четыре ячейки ПМК, условно нумеруемые как 0,1,2,3 (по значениям двух младших разрядов адреса). В нулевой ячейке размещается содержимое РКЗ и РАУСК, в первой — содержимое некоторых служебных триггеров и РАД, во второй — содержимое РФ, РСК, РКБ и в третьей — содержимое РКД. Выбор ячеек производится по номеру устройства, поступающему из РАУ или РАП и составляющему восемь старших разрядов адреса; модификация двух младших разрядов производится микропрограммно. Обращение к ПМК производится в процессе реализации заданных УСК при общем функционировании МК, аналогичном описанному в п. 13.2 для мультиплексного канала однородной структуры.

Селекторные каналы. Любой селекторный канал, включаемый в комплект машины ЕС-1033, предназначен для монопольного обслуживания одного из подключенных к нему внешних устройств по обмену данными с оперативной памятью; как правило, к СК подключаются внешние запоминающие устройства. Основными функциональными частями селекторного канала магистральной структуры, показанными на рис. 13.12, являются блок управления БУ, блок регистров БР СК, блок модификации

БМ, усилители-приемники УПМ и усилители-передатчики УПД; можно считать, что блок регистров и блок модификации составляют операционную часть селекторного канала. Все функциональные части СК связаны между собой магистралями М1, М2, М3, М4; магистраль М5 используется как внутренняя для блока регистров и на рис. 13.12 не показана.

Все магистрали селекторного канала построены так же, как и магистрали мультиплексного канала или магистрали процессора; на каждый разряд имеется запоминающий элемент, а совокупность запоминающих элементов данной магистрали составляет магистральный регистр. Магистраль М1 связывает блок регистров с канальным коммутатором общего канала и является четырехбайтной; магистраль М2 также четырехбайтная; по ней коды из БР передаются в блок модификации; обратная связь БМ с БР осуществляется по трехбайтной магистрали М3. По однобайтной магистрали М4 производится пересылка информации, направляемой в интерфейс ввода-вывода ИВВ и принимаемой из ИВВ. Четырехбайтная магистраль М5 связывает между собой отдельные регистры и относится к внутренним для блока регистров.

Блок регистров служит для хранения всей информации, необходимой для выполнения заданной операции ввода-вывода при работе выбранного ВУ. Он практически выполняет функции единственного подканала селекторного канала. Для блока регистров СК приняты те же принципы построения, что и для блока регистров мультиплексного канала; имеются только небольшие различия по составу запоминающих узлов и служебных триггеров, объединяемых в регистры. Основу БР составляют, как и в случае мультиплексного канала, известные регистры РКЗ, РАУСК, РКОП, РАД, РФ, РСК, РКБ, РСУ, РАУ и РКД. Кроме них и регистров, объединяющих служебные триггеры, в состав БР селекторного канала дополнительно включены трехбайтные буферные регистры РБ1, РБ2, служащие для хранения адреса УСК при выполнении цепочки данных и частей самого УСК, и четырехбайтный регистр данных РД, выполняющий функции промежуточного буферного регистра при обмене данными с ОП. Использование буферных регистров в селекторных каналах, как отмечалось выше (см. п. 13.3), необходимо для исключения случаев потерь информации в процессе выполнения операций ввода-вывода.

Все регистры выполнены по тем же схемам, что и регистры БР МК, т.е. на базе микросхем 155ХЛ1. Магистралью М1 связаны РКЗ, РАУСК, РКОП, РАД, РФ, РСК, РКБ, РСУ, РАУ, РКД, РБ1, РБ2, РД, магистралью М2 – РАУСК, РАД, РКБ, РФ, РБ2, магистралью М3 – РАУСК, РАД, РКБ, магистралью М4 – РКОП, РСУ, РКД, РАУ; к магистрали М5 кроме РКД и РД подключен еще регистр адреса устройства. Обмен информацией между регистрами и магистралями производится под управлением как общего канала, так и местных тактовых цепочек.

Блок модификации предназначен для модификации адресов оперативной памяти, количества байт, которые еще предстоит передать при обмене между ОП и ВУ по коду, содержащемуся в РКБ, и для анализа на нуль результата выполнения операции с целью завершения обмена данными по используемому УСК. Основными частями блока являются АЛУ и схемы управления выполнением действий в АЛУ. Собственно

арифметическо-логическое устройство выполнено так же, как и АЛУ мультиплексного канала, и также реализует функции F_1, F_2, F_3 и F_4. Различие их состоит в том, что значения второго операнда поступают не из БМПУ общего канала, а формируются, как и управляющие сигналы, схемами блока модификации селекторного канала.

Блок управления служит для координации работы функциональных частей селекторного канала при выполнении операций ввода-вывода. Его схемы формируют основные управляющие сигналы, используемые в процессе работы СК. В качестве начальных управляющих сигналов используются сигналы, поступающие от БМПУ общего канала и от внешних устройств по линиям интерфейса. В обобщенный блок управления СК входит так называемый ограничитель времени простоя, используемый также и в составе МК. Этот ограничитель, построенный на основе счетчика времени, формирует сигнал ошибки в интерфейсе ввода-вывода, если интервал времени между сигналами ИНФ-К и ИНФ-А (УПР-А) превышает 30 с. Все сигналы, формируемые блоком управления, отвечают общим принципам и конкретизированным алгоритмам работы селекторного канала, аналогично рассмотренным в п. 13.3 для СК обычной структуры.

Глава 14

ОПЕРАЦИОННЫЕ СИСТЕМЫ

14.1. Основные понятия и определения

Прежде чем рассматривать конкретные варианты операционных систем ЕС ЭВМ, остановимся на основных понятиях и определениях, которые в дальнейшем будут широко использованы.

Задание — основная независимая единица работы ЭВМ, задаваемая извне потребителем машинного времени (абонентом, программистом, оператором). Задания не зависят друг от друга и могут выполняться одновременно. Обычно каждое задание 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 пакет, который образует входной поток пакетированных заданий.

Шаги (пункты) 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 область 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 независимых задач и ее ресурсы с помощью операционной системы распределяются между программами этих задач.

Раздел – часть основной оперативной памяти, выделяемая для программы потребителя (абонента). Число разделов ООП определяется числом одновременно реализуемых рабочих программ абонентов, каждый раздел отводится для одной рабочей программы. Размеры разделов могут изменяться. Если область ООП, выделенная для рабочих программ абонентов, состоит из одного раздела, машина может работать только в однопрограммном режиме.

При решении задач под управлением ДОС ЕС число разделов фиксировано и равняется одному, двум или трем.

В первом случае область ООП имеет один раздел – фоновый, во втором два – фоновый и один из разделов переднего плана (*F*1 или *F*2), и в третьем – все три раздела. Компоненты управляющей программы ДОС ЕС размещаются в области управляющей программы.

Фоновая программа – это рабочая программа, которая при ее выполнении размещается в фоновом разделе. Фоновые программы составляются для фоновых задач. Рабочие программы, размещаемые в разделах *F*2 и *F*1, 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 ОС. Сама же обработка, поиск, хранение, доставка записей из ВЗУ и ОЗУ осуществляются 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) вводится и для тома.

Метод логических устройств – это метод, обеспечивающий независимость программных компонентов ОС и рабочих 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 числа 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.

Исходный модуль — это программный модуль, составленный на любом из алгоритмических языков, используемых в системе.

Объектный модуль — это программный модуль, транслированный из исходного модуля и представленный в промежуточном, общем для всех трансляторов системы формате. В объектном модуле специфика исходного алгоритмического 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 модуль, представленный на машинном языке и не подлежащий дроблению при вызове его в ООП для выполнения.

Все программы в зависимости от того, в какой стадии подготовки они находятся (исходный модуль, объектный модуль, абсолютный модуль, или фаза), могут храниться в соответствующей библиотеке. Для каждого типа программных модулей имеется своя библиотека. В библиотеке абсолютных модулей содержатся в виде фаз программные компоненты самой операционной системы, а также готовые к выполнению рабочие программы абонентов, поэтому для функционирования ОС наличие этой библиотеки обязательно (две другие библиотеки могут отсутствовать). Любая программа поступает в оперативную память для выполнения только из библиотеки абсолютных модулей. В ЕС ЭВМ могут быть как системные, так и личные (индивидуальные, принадлежащие конкретным абонентам) библиотеки объектных и исходных модулей. Существование личной библиотеки абсолютных модулей, как правило, не допускается, так как для этого потребовалась бы слишком большая емкость запоминающих устройств.

Конкретная операционная библиотека 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.

Резиденция системы — это пакет магнитных дисков, на которых хранятся программы операционной системы.

14.2. Дисковая операционная система ДОС ЕС

Дисковая операционная система ДОС ЕС предназначена для повышения эффективности функционирования ЕС ЭВМ (за счет увеличения объема вычислительной работы, выполняемой машиной в единицу времени),

автоматизации процесса создания программ, сокращения периода времени между постановкой задачи на решение и получением результатов. Она представляет собой организованный набор программ (общее количество команд в наборе около 1,7 млн), обеспечивающих управление ресурсами ЭВМ и процессом выполнения рабочих программ абонентов. ДОС ЕС может быть применена в следующих моделях ЕС ЭВМ: ЕС-1020, ЕС-1022, ЕС-1030, ЕС-1032, ЕС-1033, ЕС-1035, ЕС-1040, ЕС-1050. Возможны для использования языки программирования: АССЕМБЛЕР ЕС ЭВМ, ФОРТРАН IV, БАЗИСНЫЙ ФОРТРАН, РПГ, ПЛ/1, КОБОЛ.

В состав минимальной конфигурации технических средств, необходимых для обеспечения работы ДОС ЕС, входят пульт оператора, накопители на сменных магнитных дисках для резиденции системы, устройства ввода с перфокарт. Минимальный объем ООП — 64К байт. Для большинства вычислительных работ используются также устройства вывода на перфокарты, АЦПУ, накопители на магнитной ленте, устройства ввода с перфоленты, устройства вывода на перфоленту.

В ДОС ЕС предусмотрена возможность генерации на конкретную конфигурацию технических средств и на выполнение определенных системных функций. Для проведения генерации используется дистрибутивная магнитная лента (содержащая полную версию ДОС) и требуются следующие технические средства: процессор, оперативная память объемом не менее 64К байт, пультовая пишущая машинка, два накопителя на сменных магнитных дисках, один накопитель на магнитной ленте, устройство ввода с перфокарт, устройство вывода на перфокарты, АЦПУ.

Основными программными компонентами ДОС ЕС являются [34, 45]:

управляющая программа;

системные обслуживающие и обрабатывающие программы;

трансляторы с используемых языков программирования.

Управляющая программа обеспечивает автоматическое выполнение программ (как рабочих, так и программ самой операционной системы) на машине, т. е. автоматический 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 машины и ДОС ЕС к работе;

программа УПРАВЛЕНИЕ ЗАДАНИЯМИ, обеспечивающая подготовку ДОС ЕС к выполнению пакета заданий;

ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ, с помощью которого осуществляется прием одиночного задания и подготовка системы к выполнению принятого задания;

СУПЕРВИЗОР, осуществляющий управление всем вычислительным процессом на машине;

УПРАВЛЕНИЕ ДАННЫМИ (система управления вводом-выводом), включающее программы для управления процессом обмена данными между ООП и внешними устройствами.

В состав системных обслуживающих и обрабатывающих программ входят:

РЕДАКТОР, служащий для формирования готовых к выполнению программ из объектных модулей, создаваемых трансляторами ДОС ЕС;

БИБЛИОТЕКАРЬ, представляющий собой систему программ, предназначенных для создания библиотек ДОС ЕС, отображения их состояния, корректировки и копирования;

ОТЛАДКА (АВТОТЕСТ, или отладочная программа), назначение которой состоит в оказании помощи программисту при отладке программ;

программы перезаписи (УТИЛИТЫ), осуществляющие перемещение любых последовательных файлов с одних носителей данных на другие;

специальные программы, предназначенные для выполнения функций, которые связаны с подготовкой и обслуживанием носителей данных;

программа СЕРВИС, выполняющая в оперативном режиме ряд функций по обработке информации произвольной структуры;

программы сортировки, служащие для сортировки файлов, состоящих из неупорядоченных записей, и для объединения нескольких файлов с упорядоченными записями в один упорядоченный последовательно организованный файл;

программа проверки устройств, обеспечивающая выполнение стандартных функций, которые необходимы при тестировании устройств ввода-вывода.

Ниже названные функциональные части управляющей программы рассматриваются более подробно.

Программа ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА. 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 ЕС 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;

набирает адрес этого НМД на переключателях загрузки, расположенных на пульте оператора;

нажимает кнопку ЗАГРУЗКА.

При этом автоматически в основную оперативную память 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: корректировка таблицы физических устройств в ядре СУПЕРВИЗОРА, запуск пакетной обработки в фоновом разделе.

Оператор имеет возможность оперативно влиять на состав УВВ, с которыми будет оперировать ДОС ЕС. 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 таблицу физических устройств. 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.

Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ. 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:

подготовка программы к выполнению;

назначение логическим устройствам физических устройств;

установка режима работы системы;

редактирование и запоминание информации о метках файлов;

запись информации в область связи СУПЕРВИЗОРА;

подготовка к запуску программ с контрольной точки;

обеспечение связи с программой накопления статистики заданий, написанной абонентом;

ведение статистики надежности.

Программа УПРАВЛЕНИЕ ЗАДАНИЯМИ включается в работу каждый раз, когда необходимо подготовить систему к выполнению очередного задания или его шага. Она вызывается СУПЕРВИЗОРОМ из библиотеки абсолютных модулей (из резиденции системы) и записывается в тот раздел области рабочих программ основной памяти, в котором должно выполняться подготовляемое задание или его шаг. 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 рабочей 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.

Программа ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ. Программа обеспечивает подготовку ДОС ЕС к выполнению одиночной программы. Она выполняет функции, аналогичные функциям программы УПРАВЛЕНИЕ ЗАДАНИЯМИ, но с некоторыми ограничениями и особенностями, которые 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 ООП.

Режим одиночных программ допустим только в разделах переднего плана. В качестве одиночных программ не могут выполняться системные обслуживающие программы, трансляторы, а также рабочие программы, требующие доступа к области связи.

Программа СУПЕРВИЗОР. СУПЕРВИЗОР представляет собой набор программных фаз, обеспечивающих контроль выполнения рабочих программ на всех этапах, начиная с момента ввода задания на их выполнение и до получения результатов работы. СУПЕРВИЗОР 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 рабочих программ.

В области управляющей программы выделяется участок, называемый *областью связи,* где содержится информация, которая в ходе вычислительного процесса может потребоваться СУПЕРВИЗОРУ и рабочим программам. Это – дата выполнения рабочей программы, имя этой программы, адрес начала и конца области рабочих программ и т. д.

СУПЕРВИЗОР включается в работу по сигналам прерывания и обеспечивает:

распознавание и обработку прерываний;

планирование работы каналов;

обработку сбоев внешних устройств, программных и машинных сбоев;

вызов программных фаз из библиотеки абсолютных модулей;

выполнение процедур, связанных с окончанием выполнения задания (завершение шага задания);

создание контрольных точек;

организацию связи с оператором;

обслуживание таймера (электронных часов);

обеспечение мультипрограммного режима работы, при котором одновременно выполняется до трех рабочих программ.

Рассмотрим подробнее функции СУПЕРВИЗОРА.

Распознавание и обработка прерываний. В ЕС ЭВМ существует несколько типов прерываний: прерывания от схем контроля, обращение к СУПЕРВИЗОРУ, программный сбой, прерывание ввода-вывода, внешнее прерывание. Все прерывания принимаются СУПЕРВИЗОРОМ.

Получив управление в результате прерывания, СУПЕРВИЗОР запоминает полное состояние прерванной программы, что необходимо для обеспечения возможности продолжить ее выполнение после обработки прерывания. Состояние прерванной программы запоминается в специальной области основной памяти, называемой *областью сохранения.* В режиме мультипрограммной работы каждый раздел основной оперативной памяти имеет свою область сохранения.

После установления типа прерываний 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.

Планирование работы каналов. Вся работа с внешними устройствами проходит через программу, которая называется *планировщиком каналов.* В функции этой программы входит планирование запросов на выполнение операций ввода-вывода (создание очередей 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.

Программирование ввода-вывода осуществляется обычно на так называемом *логическом уровне,* когда программист не должен 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 программирования ввода-вывода реализуется совокупностью программ, называемых логической системой управления вводом-выводом или планировщиком каналов.

Обработка сбоев. В случае появления сбоев при выполнении операций ввода-вывода действия, предпринимаемые СУПЕРВИЗОРОМ, зависят от вида сбоя. Так, сбои оборудования канала и сбои 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 рабочей программы.

При других видах сбоев реакция СУПЕРВИЗОРА зависит от специфики внешнего устройства, в котором произошел сбой, от момента сбоя и состояния канала, устройства управления ВЗУ или УВВ и внешнего устройства в этот момент, от режима обработки сбоев, установленного рабочей программой. Эта реакция может быть такой:

а) 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;

б) игнорирование сбоя, если он не приводит к искажению информации и не влияет на ход выполнения программы;

в) возврат управления рабочей программе для принятия необходимых мер по устранению последствий сбоя;

г) 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.

Вызов программных фаз. Одна из функций СУПЕРВИЗОРА – вызов программных фаз из библиотеки абсолютных модулей в основную память. В этой библиотеке могут храниться программы двух типов:

а) абсолютные – записываются только в ту область основной памяти, для которой они были отредактированы программой РЕДАКТОР; ядро СУПЕРВИЗОРА, большинство обслуживающих программ и рабочих программ являются абсолютными;

б) 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 рабочих программ, осуществляется подпрограммой ядра СУПЕРВИЗОРА, который называется *системным загрузчиком.*

Завершение шага задания. После окончания выполнения рабочей программы управление передается СУПЕРВИЗОРУ для реализации процедур завершения шага задания или всего задания. Управление передается как в случае нормального, так и в случае ненормального завершения рабочей программы, 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 или нового задания. При ненормальном завершении программы прекращается выполнение всего задания, выводится сообщение о причине прекращения его выполнения и для приема следующего задания управление передается программе УПРАВЛЕНИЕ ЗАДАНИЯМИ.

Создание контрольных точек. Если рабочая программа выполняется в течение длительного времени, СУПЕРВИЗОР дает возможность в процес-

ее 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 рабочей программы с некоторой промежуточной точки, а не с самого начала. Это дает возможность избежать непроизводительных потерь времени на повторение той части программы, которая ранее была выполнена правильно.

Организация связи с оператором. ДОС ЕС обеспечивает двустороннюю связь с оператором, предоставляя возможность оператору, с одной стороны, получать сведения о состоянии системы во время 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 машина. Все сообщения оператору система выдает на пишущую машинку и с нее же 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 его шага, напечатать таблицу распределения основной памяти, перераспределить область рабочих программ между разделами и т. д.

Директивные указания могут передаваться как управляющей программе ДОС ЕС, так и тем из обслуживающих и рабочих программ, которые имеют специальные подпрограммы для принятия и обработки таких указаний. Директивные указания, воспринимаемые и обрабатываемые управляющей программой, подразделяются на директивы программ ВНИМАНИЕ, ИНИЦИАТОР ОДИНОЧНЫХ ПРОГРАММ и 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, в результате обработки которого СУПЕРВИЗОР вводит в транзитную область программу ВНИМАНИЕ. Эта программа настраивается на прием 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.

Обслуживание таймера. В составе операционной системы имеется таймер, используя который, СУПЕРВИЗОР обеспечивает:

слежение за временем дня, которое рабочая 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 рабочей программы и передано управление по заданному в этой программе адресу;

установление интервала времени, в течение которого программе не разрешено выполнение.

Обеспечение мультипрограммного режима работы. Важной функцией СУПЕРВИЗОРА является обеспечение мультипрограммного режима работы. Одновременно может выполняться до трех независимых рабочих 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 простои 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 программы 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 состояния программ, таблица состояния системных файлов на дисках, таблица очередей к внешним устройствам и др. СУПЕРВИЗОР выполняет свои функции, используя содержимое этих таблиц.

Программа УПРАВЛЕНИЕ ДАННЫМИ (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 ЕС, обеспечивает контроль реализации канальных программ. Она включает средства для планирования и установления очередности операций ввода-вывода, контроля выполнением 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 следующие операции: считывание 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 ДОС ЕС применяются три метода доступа: методы последовательного, прямого и индексно-последовательного доступа.

Метод последовательного доступа — последовательная организация данных, при которой логические записи на внешнем накопителе располагают-

ся в том порядке, в котором они затем будут обрабатываться. В этом случае для доступа к требуемой логической записи файла необходимо перебрать все его предшествующие записи, 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 данного 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 преимущества 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 рабочую программу необходимых подпрограмм системы LIOCS, а также для указания действий, которые должны выполняться в ходе обработки файла.

При трансляции или редактировании рабочей программы с помощью императивных макрокоманд в нее включаются все подпрограммы системы LIOCS, затребованные этими макрокомандами. Обычно эти подпрограммы транслируются отдельно от рабочих программ и помещаются в библиотеку объектных модулей. Включение их в рабочую 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 существуют отдельные обобщенные модули. Аналогичные модули используются для каждого из типов внешних устройств, к которым может быть применен метод последовательного доступа. Подпрограммы этих модулей учитывают особенности и возможности методов доступа и специфику используемых внешних устройств.

Генерация ДОС ЕС. Пользователь получает ЭВМ вместе с ДОС ЕС, 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. Пользователю поставляется один из конкретных вариантов СУПЕРВИЗОРА, который хотя и может быть применен независимо от конфигурации конкретной машины ЕС ЭВМ, однако не учитывает особенностей этой машины и в нем отсутствуют многие необходимые для пользователя 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 модули всех программных компонентов ДОС ЕС 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.

Используя полученную ДОС ЕС и средства генерации системы, можно образовать конкретную ДОС ЕС, соответствующую составу оборудования машины и области ее применения. СУПЕРВИЗОР образованной системы учитывает конфигурацию машины и затребованные пользователем свойства, а в системные библиотеки помещаются только нужные пользователю компоненты ДОС ЕС.

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 в библиотеку абсолютных модулей;

формируются новые системные библиотеки, при этом из них удаляются модули тех программных компонентов системы, которые не предполагается использовать в машине.

Постановка задачи к решению на машине с помощью ДОС ЕС. Процесс постановки задачи к решению на машине ЕС ЭВМ включает следующие этапы:

планирование структуры программы и кодирование ее;

подготовку 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 из них использован свой язык, так как система обеспечивает независимую трансляцию каждой автономно подготовленной части и объединение их в одну программу затребованной структуры.

Программист может спланировать программу либо простой структуры, когда для выполнения вся программа 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 получаются объектные модули, которые затем подвергаются редактированию. Если объектный модуль некоторой программы не является в какой-то степени универсальным, т. е. не может быть использован при выполнении других программ или при выполнении той же программы, но в другое время, то после его редактирования он исчезает из системы. В противном случае объектный модуль отправляется в библиотеку объектных модулей. Во время редактирования программист с помощью управляющих операторов РЕДАКТОРА описывает структуры своей программы, указывая 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 оформить 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 самоперемещающимися, 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.

Версии ДОС ЕС. По мере совершенствования дисковой операционной системы ее возможности расширяются главным образом за счет добавления новых программных компонентов. Появляются все новые и новые версии (редакции) ДОС ЕС. Так, версии ДОС-1.1 и ДОС-1.2, выпущенные в 1972 г., обеспечивали работу ЭВМ только в однопрограммном режиме. Затем появились более совершенные версии ДОС-2.1, ДОС-2.2 и др.

Версия ДОС-2.1, выпущенная в 1976 г., совместима с предыдущими изданиями ДОС ЕС. Она предназначена для работы с ЭВМ ЕС-1020, ЕС-1022, ЕС-1030, ЕС-1033, ЕС-1035, ЕС-1045 средней конфигурации без виртуальной памяти.

По сравнению с предыдущими изданиями ДОС-2.1 содержит следующие основные изменения:

введены программные компоненты для работы с дополнительными периферийными устройствами (накопителями на магнитной ленте ЕС-5022, перфокарточными устройствами вывода ЕС-7013, ЕС-7014, печатающими устройствами ЕС-7034 и перфоленточными устройствами вывода ЕС-7021, ЕС-7024);

внесены изменения в программы сортировки (добавлена возможность работы с печатающим устройством ЕС-7034 и накопителем на магнитной ленте ЕС-5022);

включены дополнения в транслятор РПГ: предусмотрена возможность работы с устройствами ЕС-7013, ЕС-7014, ЕС-7034, ЕС-5022;

включены средства эмуляции ЭВМ «Минск-32» на модели ЕС-1035.

Версия ДОС-2.2 (выпущена в 1978 г.) по сравнению с ДОС-2.1 имеет дополнительные возможности: транслирование с языка РПГ-2, использование телекоммуникационного метода доступа с очередями, использование более расширенного состава устройств ввода-вывода.

14.3.

Операционная система ОС ЕС. Состав и функции ОС ЕС

Операционная система ОС ЕС предназначена для использования на ЭВМ Единой системы, начиная с модели ЕС-1020, и, кроме того, на всех машинах, программно совместимых с ЕС ЭВМ. Она обеспечивает повышение производительности машин при решении широкого круга научно-технических, экономических и специальных задач путем управления распределением всех имеющихся ресурсов, включая машинное время процессора, емкость ОП и ВЗУ, устройства ввода-вывода, любые программы, содержащиеся в различных библиотеках. Кроме того, ОС ЕС предоставляет возможность оператору (программисту) взаимодействовать с вычислительной системой в удобных для него формах, а также возможность гибкого и упорядоченного изменения состава самой системы при расширении сферы ее применений.

ОС ЕС обеспечивает работу следующих технических средств: процессоров ЕС ЭВМ и соответствующих устройств ОП, пульта оператора, НМД, НМЛ, печатающих устройств, перфокарточных и перфоленточных устройств ввода, перфокарточных и перфоленточных устройств вывода, графических и алфавитно-цифровых дисплеев, графопостроителей, устройств телеобработки. Для эффективного применения ОС ЕС емкость основной памяти должна быть не менее 128К байт.

Минимальный состав технических средств, необходимых для функционирования ОС ЕС, включает центральный процессор, устройство основной памяти емкостью не менее 64К байт, мультиплексный канал, селекторный канал, пишущую машинку для связи оператора с системой, два устройства прямого доступа (НМД), перфокарточное устройство ввода или НМЛ, перфокарточное устройство вывода или НМЛ, печатающее устройство.

Состав и структура ОС ЕС определены с учетом удовлетворения двух главных требований: универсальности и возможности настройки на некоторый конкретный круг решаемых задач.

Универсальность ОС ЕС заключается в возможности ее использования во всех областях применений и при всех конфигурациях моделей ЕС ЭВМ. Универсальность операционной системы обеспечивается наличием различных средств автоматизации программирования, модульностью ее структуры и применением единой системы внутренних связей, что позволяет комбинировать системные компоненты различным образом, возможностью включения в ОС ЕС новых программ, независимостью программ и данных от конфигурации устройств ввода-вывода.

Возможность настройки ОС ЕС на конкретный круг задач, решаемых вычислительной системой, реализуется в процессе генерации ОС, когда формируется рабочий вариант операционной системы, ориентированной на конкретный класс решаемых задач и определенную конфигурацию технических средств.

Для ОС ЕС 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 байт), 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, включающих 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 байт.

В мультипрограммном режиме с переменным числом задач, который целесообразно применять при емкости основной памяти свыше 512К байт, 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-1033 можно организовать двухпроцессорную систему, включающую два процессора, до восьми блоков основной памяти емкостью 256К байт 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 ЕС обеспечивает работу ЕС ЭВМ как в режиме пакетной обработки, так и в режиме разделения времени, когда осуществляется оперативное взаимодействие центрального процессора и основной памяти с удаленными и локальными абонентскими пунктами (терминалами) в реальном масштабе времени.

Операционная система ОС ЕС состоит из управляющей и обрабатывающих программ. Основные функции ОС реализуются с помощью управляющей программы и заключаются в следующем: управление данными (хранение, поиск и обслуживание данных), управление заданиями (ввод и

планирование заданий, поступающих непрерывным потоком), управление задачами (последовательное или приоритетное выполнение каждой единицы работы). Управляющая программа определяет порядок выполнения обрабатывающих программ и программ потребителей в различных режимах работы и обеспечивает их необходимым сервисом.

В состав обрабатывающих программ включены трансляторы, сервисные программы, поставляемые разработчиком, а также обрабатывающие программы, написанные пользователями. Набор обрабатывающих программ может быть расширен усилиями как разработчика, так и пользователей.

Обрабатывающие программы предназначены для сокращения объема работы и времени, затрачиваемых на написание, подготовку и выполнение программ пользователей.

ОС ЕС включает *трансляторы* 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 включают редактор связей, загрузчик, программу СОРТИРОВКА — ОБЪЕДИНЕНИЕ, набор программ-утилитов, ТЕСТРАН и программы обслуживания системного программиста.

Редактор связей, как уже отмечалось, предназначен для объединения объектных модулей в одну программу, которую можно загрузить 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 (без записи в библиотеку) осуществляет загрузку отредактированной программы в основную память, что позволяет сократить время на редактирование и выполнение программы.

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 информации на магнитных лентах или дисках. Программа СОРТИРОВКА — ОБЪЕДИНЕНИЕ используется независимо от других программ. Кроме того, она может непосредственно использоваться программой, составленной трансляторами с языков КОБОЛ и ПЛ/1.

Программы-утилиты выполняют следующие функции:

передачу информации из одного ВЗУ или УВВ в другое (передача производится с изменением формата и редактированием);

редактирование, перераспределение и обновление данных в библиотеках;
изменение индексной структуры каталога системной библиотеки;
печать сведений о наличии данных и программ, каталогизированных в библиотеке системы.

ТЕСТРАН — это средство автоматизации отладки программ. Он позволяет в определенных точках выполнения отлаживаемой программы записывать состояние регистров и определенных областей основной памяти. ТЕСТРАН позволяет, кроме того, прослеживать передачи управления в процессе выполнения программы, что помогает обнаружению ошибок в ее логике. При работе ТЕСТРАНа можно производить выборочное редактирование и вывод на печать результатов отладки.

Программы обслуживания системного программиста, включенные в состав ОС ЕС, выполняют следующие функции: распечатку и модификацию данных на НМД в загрузочных модулях в пакетном режиме или в режиме диалога с пульта оператора, выдачу информации об очередях заданий и содержимом основной памяти, восстановление данных с дефектных дорожек магнитных дисков.

14.4.

Управляющая программа ОС ЕС. Управление данными

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, предоставляют ему единый способ идентификации и размещения программ и данных и единые методы обращения к данным после их размещения.

Комплекс программ управления данными состоит из супервизора ввода-вывода, управления наборами данных, управления каталогом и программ методов доступа.

Основные функции супервизора ввода-вывода состоят в обеспечении одновременной работы процессора и каналов ввода-вывода, обслуживании очереди запросов на обслуживание каналов, регистрации и исправлении ошибок ввода-вывода.

Вся операционная система, рабочие программы пользователей и данные, обрабатываемые этими 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. Различие разных типов данных заключается в самом характере этих данных и способах их использования. Так, набор записей данных обрабатывается программой пользова-

теля, 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 ЕС 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 ЕС имеется обширный набор программ-утилитов, 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 рабочей 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 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 методов доступа получают из описаний этих наборов, где, в частности, есть такие сведения, как организация данных, физическое размещение, размер наборов данных.

В ОС ЕС реализованы следующие методы доступа к данным:

последовательный метод доступа с очередями;

базисный последовательный метод доступа:

библиотечный метод доступа;

базисный прямой метод доступа;

индексно-последовательный метод доступа с очередями;

базисный индексно-последовательный метод доступа.

Эти методы отличаются особенностями пересылки и организации данных. Используются два метода пересылки данных: доступ с очередями и базисный.

При доступе с очередями происходит оперирование индивидуальными записями и предоставляется максимально возможное количество средств автоматического выполнения операций ввода-вывода. При этом обеспечивается автоматическая синхронизация обработки программ и выполнения ввода-вывода. Записи, выбранные и записанные по специальным макрокомандам, доставляются в основную память с упреждением, т. е. до того, как они действительно запрашиваются. Следовательно, программы управления данными, реализующие такой вид пересылки данных, должны иметь возможность упреждать запросы. Доступ с очередями может использоваться только для выборки записей, упорядоченных последовательно.

При базисном доступе часть работы по управлению данными перекладывается на программиста, который вынужден принимать более непосредственное участие в управлении операциями ввода-вывода. Пересылка информации по специальным макрокомандам осуществляется не индивидуальными записями, как в предыдущем случае, а блоками. Автоматическая синхронизация обработки программы и выполнения ввода-вывода не обеспечивается, поэтому, прежде чем может быть использована информация, введенная в основную память, необходимо убедиться в завершении операции ввода.

Применяются следующие виды организации данных: последовательная, библиотечная, индексно-последовательная и прямая.

Последовательный метод доступа с очередями базируется на доступе с очередями при последовательной организации данных. При выполнении задачи выдается требование на ввод или вывод логической записи. Синхронизация выполнения задачи процессором, блокирование и разблокирование записей, завершение операций ввода-вывода — все эти операции реализуются программами метода доступа.

В случае базисного последовательного метода доступа, когда используется базисный доступ и последовательная организация данных, обмен информацией между основной оперативной памятью и внешними устройствами осуществляется физическими блоками. Операции блокирования и разблокирования записей, а также синхронизация выполнения задачи процессором и завершения ввода-вывода должны программироваться программистом. Проверка завершения операции ввода-вывода производится по специальным макрокомандам.

При использовании библиотечного метода доступа последовательные данные организованы в виде библиотек и располагаются на устройствах прямого доступа. Библиотечный набор данных состоит из независимых групп последовательно организованных наборов данных (разделов). Каждому разделу присвоено основное имя и, кроме того, несколько дополнительных имен. Имена разделов библиотечного набора данных в алфавитном порядке записываются в специальной части этого набора — справочнике библиотеки. Имени раздела в справочнике библиотеки сопоставлен адрес этого раздела на томе. Адрес, относящийся к дополнительному имени, может быть адресом любой записи внутри раздела. С помощью библиотечного метода доступа можно осуществлять эффективный поиск и обработку раздела по его имени, а также добавлять или замещать разделы в библиотечных наборах данных, которые обычно используются для хранения программ.

В случае базисного прямого метода доступа применяется прямая организация данных, которая обеспечивает возможность произвольной обработки данных, и базисный способ их пересылки. Структура набора данных определяется программистом. Запись и чтение могут производиться по относительным или абсолютным адресам внутри набора данных. Для того чтобы прочитать или записать блок данных, необходимо указать действительный адрес (т. е. устройство прямого доступа, номер цилиндра, дорожки и позицию блока) или же адрес относительно начала набора данных. Адрес может указывать точку внутри набора данных, начиная с которой производится поиск свободного места.

Если блоки данных снабжены ключами, то в составе программ методов доступа должны быть программы для поиска блока с ключом, который указан. Прямая организация данных допустима только на устройствах прямого доступа. Способ доступа с очередями использовать нельзя.

Индексно-последовательный метод доступа с очередями базируется на использовании индексно-последовательной организации данных и доступе с очередями. Как известно, в индексно-последовательной организации данных расположение записей в наборе данных поставлено в зависимость от ключа каждой записи. Данные организованы в порядке возрастания ключей логических записей. В набор данных включается несколько уровней индексов, с помощью которых устанавливается связь между ключом записи и ее физическим адресом на накопителе. Индексно-последовательная организация данных может использовать каждый из видов доступа. Индексно-последовательный метод доступа с очередями позволяет выполнять последовательную обработку любой группы записей из набора данных. Операции синхронизации выполнения задачи процессором и завершения ввода-вывода осуществляются программами метода доступа.

При использовании базисного индексно-последовательного метода доступа можно производить непосредственную обработку отдельной записи по ее ключу. Запись может быть считана, обновлена, ликвидирована или вставлена в набор данных. Операции синхронизации выполнения задачи процессором и завершение ввода-вывода реализуются программистом.

Таким образом, из рассмотрения методов доступа видно, что они представляют собой комбинации организации наборов данных и видов доступа к данным (видов пересылки данных). Базисный доступ используется для всех видов организаций данных, а доступ с очередями — только для последовательных и индексно-последовательных 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 наборами данных 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.

14.5. Управляющая программа ОС ЕС. Управление заданиями

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 выполняет следующие функции:

анализ входного потока заданий, во время которого осуществляется просмотр введенных наборов данных, выделение, интерпретация и анализ управляющих операторов, подготовка необходимых 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 их последовательно или в соответствии с приоритетами;

связь между оператором и системой.

Перед обработкой отдельного задания оно может быть объединено с другими 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 входная 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 программистов 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 непосред-

ственно в задание как часть общего входного пакета. Точно так же 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, в режиме одной задачи (последовательный планировщик), либо параллельно с учетом приоритетов, в режимах мультипрограммирования с фиксированным или с переменным числом задач (приоритетный планировщик). Он выполняет три функции: чтение-интерпретацию, инициирование-завершение и обработку системного вывода.

Функции чтения-интерпретации заданий 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 для выполнения очередного пункта задания 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 их установки 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 его установки 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 одном устройстве вывода, либо одновременно на нескольких. Запись, как и чтение, может выполняться либо одновременно, либо независимо от обработки заданий. Последовательный планировщик выполняется в динамической части основной памяти между выполнениями пунктов заданий. Он обрабатывает только один поток заданий, причем переход от обработки одного задания в потоке к другому осуществляется автоматически, без вмешательства оператора.

Приоритетный планировщик планирует порядок, в котором должны обрабатываться задания, в соответствии с назначенными для заданий классами и приоритетами и с учетом имеющихся в наличии ресурсов вычислительной системы. Интерпретируемые задания размещаются по 15 очередям в соответствии с классом заданий. Внутри классов задания располагаются по приоритетам. Классы и приоритеты приписываются заданиям программистом и могут быть изменены оператором. При параллельном планировании обрабатывается несколько заданий (но не более 15) или пунктов задания одновременно и осуществляется совместное использование ресурсов системы.

Главный планировщик предназначен для выполнения функций прямой и обратной связи между оператором и системой, т.е. 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 сообщающие системе текущее время и дату.

Команды оператора обычно вводятся в систему с такого устройства, как пультовая печатающая машина, 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: оператора задания, исполнительного оператора, оператора определения данных, командного оператора, разделительного оператора, пустого оператора. Параметры, закодированные в этих операторах, позволяют 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 томе и об устройстве, о размещении набора данных, описывают метки набора данных.

Командный оператор используется оператором системы для помещения тех 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 данных 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. Все возможности управления заданиями доступны пользователям на любом языке программирования. Система ОС ЕС отвечает на те запросы, которые сделаны при описании задания на языке управления заданиями.

14.6.

Управляющая программа ОС ЕС. Управление задачами

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 следующие:

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 требуется неоднократно и одновременно для нескольких различных задач, то она загружается 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 исполнением. 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.

Таким образом, в управляющей программе ОС ЕС используются несколько типов СУПЕРВИЗОРов:

СУПЕРВИЗОР для однопрограммного режима (PCP), или для режима с одной задачей;

СУПЕРВИЗОР для мультипрограммного режима с фиксированным числом задач (MFT);

СУПЕРВИЗОР для мультипрограммного режима с переменным числом задач (MVT);

Первый из этих СУПЕРВИЗОРов рассматривается как основа для всех вариантов управляющей программы, а остальные СУПЕРВИЗОРы являются его функциональным расширением.

СУПЕРВИЗОР для PCP выполняет следующие функции: обработку прерываний всех видов, обеспечение выполнения одной задачи, распределение основной памяти, защиту основной памяти, нахождение программы в библиотеке по ее имени, загрузку программных модулей и передачу им управления, загрузку программных модулей, имеющих оверлейную структуру, и передачу управления между сегментами такого модуля, планирование и управление операциями ввода-вывода, обеспечение связи программиста со схемным таймером, завершение заданий и обработку аварийных ситуаций, регистрацию ошибок оборудования для диагностики, запись контрольных точек.

СУПЕРВИЗОР для режима MFT по сравнению с СУПЕРВИЗОРом для режима PCP обладает целым рядом особенностей. В режиме MFT основная память делится на две основные области: системную и динамическую. Динамическая область делится на разделы. Оператор может изменять число и размеры разделов. Число разделов может быть изменено в пределе, установленном при генерации ОС ЕС. Максимальное число разделов равно 52. Все разделы являются независимыми по отношению к планированию и инициализации заданий. Каждый раздел может содержать в любое время только одну активную

задачу, поэтому одновременно может быть выполнено до 52 задач. Из 52 разделов максимум 15 могут использоваться для выполнения заданий. Каждый из этих 15 разделов может, кроме программ задания, содержать также несколько программ прямого системного вывода. Если ОС ЕС допускает подзадачи, то общее число одновременно выполняемых задач и подзадач не должно превышать 255, причем сюда входят как системные задачи и подзадачи, так и задачи и подзадачи пользователей. Приоритет каждой задачи (или подзадачи) определяется номером раздела, в котором она размещается. Задачи, выполняющиеся в разделах с большими адресами, имеют более высокий приоритет.

В ходе вычислительного процесса обеспечивается защита разделов ОП ключами защиты памяти. Обеспечивается также контроль за совместным использованием ресурсов вычислительной системы. Оператору выдается расширенная информация о состоянии вычислительной системы. При генерации ОС ЕС можно обеспечить режим квантования времени между задачами определенных разделов, причем оператор может изменять параметры этого режима.

СУПЕРВИЗОР для режима MVT требует не менее 512К байт основной памяти. В этом режиме динамическая область основной памяти, доступная для программ пользователей, делится не на разделы, а на зоны, причем деление осуществляется динамически, а размер зоны определяется в управляющих операторах задания. Каждое вызванное для выполнения задание оперирует внутри зоны. Одновременно могут выполняться до 15 независимых заданий. В каждой зоне может выполняться несколько задач. Их приоритет определяется числом в пределах от 0 до 255К и устанавливается при инициировании пункта задания или при образовании задачи.

В режиме MVT имеется дополнительное средство, называемое свертыванием-развертыванием. Это средство позволяет осуществлять временное динамическое распространение конкретного задания за пределы отведенной ему зоны. При этом низкоприоритетные задания вытесняются во внешнюю память, а затем, когда основная память освобождается, вытесненные задания вновь загружаются и продолжается их выполнение.

Эффективность использования возможностей управления задачами 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 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 и предельного приоритетов. Первый из них используется СУПЕРВИЗОРом при рассмотрении запросов на ресурсы системы, а второй определяет ограничения на изменение первого 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 операцию 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 пользование таблицей 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. Эти функции 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 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 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 байт), выделяемый 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:

посылки запроса СУПЕРВИЗОРу из любой точки программы на выдачу даты и текущего времени;

посылки запроса СУПЕРВИЗОРу после заданного интервала времени; по этому запросу СУПЕРВИЗОР помещает задачу в состояние 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;

установления того, сколько истекло запрошенного времени, или отмены 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). Задачу, не входящую в группу квантования (т. е. не входящую в группу задач, выполняющихся в режиме квантования), подготовляют и выбирают в соответствии с ее текущим приоритетом. Следовательно, такую задачу выбирают только тогда, когда она является задачей более высокого приоритета по сравнению с задачами группы квантования и находится в состоянии готовности.

14.7.
Трансляторы. Версия ОС ЕС

Как уже отмечалось, операционная система ОС ЕС обеспечивает функционирование системы программирования, включающей трансляторы с языков программирования АССЕМБЛЕР, РПГ, ФОРТРАН, ПЛ/1, АЛГОЛ, КОБОЛ. Результатом работы трансляторов являются объектные программы (модули), имеющие стандартный формат. Объединение нескольких объектных модулей в один, который в дальнейшем может быть загружен в основную память и выполнен, производится редактором связей. Трансляторы имеют средства отладки, которые используются для динамической проверки всей рабочей программы или ее части в соответствии с указаниями, записанными на исходном языке.

АССЕМБЛЕР. Язык 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 программист получает:

символическую адресацию элементов программы;

мнемонические коды команд, символически определяющие функции команд;

разнообразные способы представления данных;

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 обеспечивает преобразование программ, написанных на машинно-ориентированном языке АССЕМБЛЕР, в объектный код. Для его работы требуются не менее 44К байт основной памяти, вспомогательные наборы данных, набор данных для входной информации, набор данных, содержащий системные микроопределения, четыре набора данных для выходной информации и некоторые программы в кодах машины. Все эти наборы данных организованы последовательно, за исключением набора данных системных макрораспределений, имеющего библиотечную организацию.

Важной частью языка АССЕМБЛЕР являются операторы условной компиляции, которые позволяют изменять порядок обработки операторов или производить выборочную обработку операндов исходного модуля транслятором АССЕМБЛЕРа.

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. Характерной особенностью таких задач является

незначительный удельный вес вычислений и большой объем ввода-вывода.

Язык РПГ позволяет:

описывать типы и структуры 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. Рабочие файлы транслятора РПГ размещаются на дисках и лентах. Для его работы необходимо не менее 15К байт основной памяти, три набора данных на магнитных дисках или лентах, устройства для системного ввода и вывода на печать.

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. Для его работы требуются не менее 80К байт основной памяти, а также устройства системного ввода и вывода на печать.

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 связей.

ПЛ/1. В состав ПЛ/1 входят язык ПЛ/1, транслятор, преобразующий исходный модуль в объектный, и библиотека подпрограмм. Язык ПЛ/1 — один из наиболее универсальных и высокоразвитых языков программирования. Он в одинаковой степени пригоден для программирования как научно-технических, так и экономических задач.

Транслятор с языка ПЛ/1 требует наличия не менее 44К байт основной памяти и устройств для системного ввода и вывода на

печать; он может использовать интервальный таймер и два рабочих набора данных. Транслятор позволяет программисту использовать различные типы данных, получать программы с модульной структурой, редактировать и преобразовывать данные различных типов, производить сортировку, динамически выделять и освобождать основную память для областей данных переменных размеров, использовать встроенные функции ПЛ/1 и разнообразные средства отладки, а также средства условной компиляции, которые позволяют изменять порядок обработки операторов или производить их выборочную обработку.

Библиотека подпрограмм является частью системной библиотеки объектных модулей; она включает подпрограммы, предназначенные для выполнения служебных функций, обеспечения диагностики при отладке программы и реализации встроенных функций языка. Включение подпрограмм в рабочую программу при ее обработке осуществляется редактором связей.

АЛГОЛ. В состав АЛГОЛ входят язык АЛГОЛ-60, транслятор с этого языка и библиотека подпрограмм.

Язык АЛГОЛ-60 представляет собой проблемно-ориентированный алгоритмический язык, используемый для программирования научно-технических задач.

Для работы транслятора требуются не менее 44К байт основной памяти, а также системные средства ввода и вывода на печать и три рабочих набора данных на внешних запоминающих устройствах, одно из которых должно быть устройством прямого доступа. При его работе используются базисный последовательный и последовательный с очередями методы доступа.

Библиотека подпрограмм представляет собой библиотечный набор данных. В него входят подпрограммы для выполнения стандартных математических функций и процедур ввода-вывода, определенных в языке АЛГОЛ-60.

КОБОЛ. В состав КОБОЛ входят алгоритмический язык КОБОЛ, транслятор и библиотека транслятора.

Язык КОБОЛ является широко распространенным проблемно-ориентированным языком программирования, используемым для программирования экономических задач. Он включает ядро и ряд функциональных модулей, обеспечивающих следующие операции: обработку таблиц, последовательный и произвольный доступ к файлам, сортировку, сегментацию, генерацию отчетов, обработку файлов с индексно-последовательной организацией, связь между программами, записанными на других языках, отладку на уровне исходного языка и некоторые другие операции, реализующие возможности операционной системы.

Транслятор с языка КОБОЛ требует наличия не менее 17К байт основной памяти, трех вспомогательных наборов данных на устройствах одного и того же типа, устройств для системного ввода и вывода на печать.

Библиотека транслятора содержит наиболее часто употребляемые стандартные подпрограммы.

Версии ОС ЕС. Рассмотренные структура и возможности операционной системы ОС ЕС характерны для версии ОС-4.1 ЕС. Она является

дальнейшим развитием системы ОС-4.0 ЕС и совместима со всеми предыдущими версиями ОС ЕС.

По сравнению с ОС-4.0 ЕС версия ОС-4.1 ЕС имеет следующие основные изменения и дополнения.

1. Предусмотрена работа с целым рядом дополнительных периферийных устройств, в частности предусмотрена возможность использования алфавитно-цифровых дисплеев ЕС-7066 и графических дисплеев ЕС-7064 в качестве консоли оператора; обеспечена динамическая реконфигурация периферийных устройств, позволяющая оператору осуществлять обмен томов, а также переключать запросы на ввод-вывод с одного периферийного устройства на другое.

2. Разработан и реализован новый, более совершенный общий телекоммуникационный метод доступа (ОТМД), представляющий собой обобщенную систему управления вводом-выводом, в который методы обработки данных распространяются в сферу телеобработки. Метод ОТМД функционирует под управлением ОС-4.1 ЕС с мультипрограммированием: его средства содержат обширный набор подпрограмм ввода-вывода, управления сообщениями, трансляции, обработки ошибок и редактирования. Метод обеспечивает реконфигурацию сети телеобработки на логическом уровне, приоритетное планирование, коммутацию и динамическое упорядочение сообщений в их очереди, автоматический сбор статистических данных о работе системы телеобработки, связь с оператором и прикладными (рабочими) программами пользователей. Обеспечивается обслуживание не только устройств телеобработки, но и локальных дисплеев ЕС-7906.

3. Введен новый компонент – удаленный ввод заданий в диалоговом режиме. Задания вводятся через абонентский пункт АП-70 в центральную ЭВМ, затем выполняются в режиме пакетной обработки, и результаты отсылаются на абонентский пункт. Пользователю предоставлены удобные средства редактирования программ в диалоговом режиме – синтаксические анализаторы для программ, составленных на языках ПЛ/1 и ФОРТРАН.

4. Включены новые программы обслуживания системного программиста, которые применяются для сбора и распечатки информации о состоянии операционной системы. Введены программы, которые дают возможность оперативно вносить изменения в наборы данных на томах прямого доступа.

5. Получено дальнейшее развитие средств восстановления системы после сбоев; эти средства включены для модели ЕС-1022, а также для каналов ввода-вывода моделей ЕС-1050 и ЕС-1060.

В более совершенных версиях операционной системы (ОС-6.0 ЕС, ОС-6.1 ЕС и др.) реализованы мультипрограммные режимы работы с виртуальной памятью, режим разделения времени с более сложными вариантами дисциплин обслуживания запросов абонентов, совместная работа двух и более процессоров.

Глава 15

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 оконечными устройствами данных (ОУД).

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 быстродействия процессорной аппаратуры средства АПД выполняют с различным быстродействием. Например, модели ЕС ЭВМ соответственно их быстродействию 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. Она управляет обменом информацией, а также осуществляет ее хранение и обработку. Абонентские пункты — массовые звенья. Через АП абонент получает возможность дистанционного доступа к ЭВМ вычислительного центра. Звено данных могут образовывать 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 в режиме *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. Основной пункт, послав запрос, игнорирует встречный запрос от второстепенного пункта. В режиме подчинения один из пунктов (обычно ЭВМ) 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 составляются из служебных символов и формируемых из них последовательностей. Символы (см. табл. 4 и 5 приложения) для работы, например, по телефонным каналам связи кодируются согласно таблице стандартных 7-элементных кодов обмена информацией КОИ-7 (см. табл. 2.3).

Процесс обмена информацией разбивается на следующие 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 системы телеобработки, работающими под управлением ОС ЕС.

Программное обеспечение системы телеобработки дает возможность использовать все средства операционных систем. Оно также осуществляет контроль и диагностику технических средств с целью обнаружения и устранения 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 выходящих из ЭВМ сообщений.

Соответственно этому первая часть обеспечивается программами управления передачей, а вторая – программами 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. Такие программы имеются в операционных системах. Программы обработки составляются пользователями с учетом специфики решаемых задач и способов обработки информации.

В операционной системе ОС ЕС имеются два вида комплексов

общих управляющих программ, предназначенных для обслуживания систем телеобработки. Это – программы базисного БТМД и общего ОТМД телекоммуникационных методов доступа. Программы БТМД предназначены в основном для организации приема и передачи данных в звеньях ЭВМ–АП и ЭВМ–ЭВМ. При этом обеспечивается управление абонентскими пунктами, передачей сообщений с автоматическим обнаружением и исправлением ошибок, преобразованием кодов и т.д. Программы ОТМД, содержащие функции БТМД, обеспечивают также управление очередями сообщений и заданий на их обработку. Эти программы дают возможность реализации мультипрограммных режимов работы более высокого уровня.

15.2. Структуры звеньев передачи данных

Обмен информацией между двумя ЭВМ или между ЭВМ и абонентскими пунктами может происходить по следующим каналам связи:

телеграфным некоммутируемым;
телеграфным коммутируемым;
телефонным некоммутируемым (выделенным);
телефонным коммутируемым;
широкополосным;
физическим – обычным проводным или кабельным линиям.

Последний способ связи возможен при небольшом удалении источников от 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 для связи ЭВМ с абонентскими пунктами, при их удалении на расстояние 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.

В настоящее время для передачи цифровых данных на большие расстояния применяются также различные каналы радиосвязи, в том числе и каналы спутниковой связи. Однако эти виды связи в данной книге не рассматриваются.

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 аппаратурных 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 данных 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.

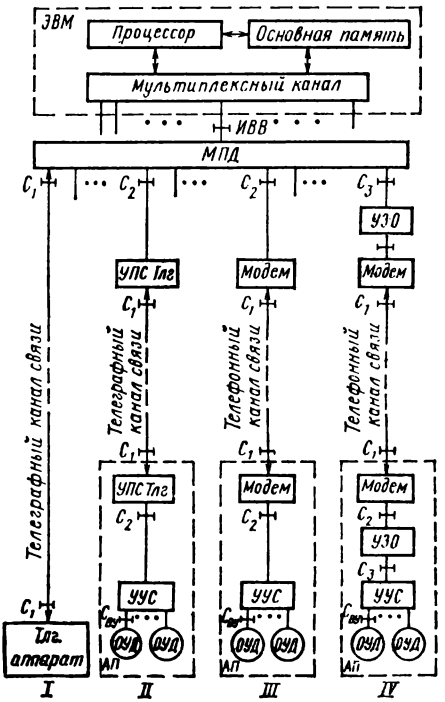


Рис. 15.1. Структуры звеньев данных на основе некоммутируемых телеграфных и телефонных каналов связи

Мультиплексор передачи данных МПД, представляющий собой автоматический многоканальный коммутатор, предназначается для сопряжения ЭВМ с аппаратурой передачи данных. По командам, поступающим из машины, он направляет информацию в нужные линии связи. Принимая с разных направлений также по командам из ЭВМ информацию, поступающую по каждой линии связи с относительно 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 интерфейса 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 соответственно с МПД и устройствами защиты от ошибок УЗО.

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 $С_1$. 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 международном 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 $С_2$. 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 осуществляются 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 $С_1$.

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, 600, 1200, 2400, 4800 и 48 000 бит/с.

С МПД и УУС модемы могут непосредственно соединяться через стыки $С_2$ (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 предназначаются 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 $С_2$. Соединение же УЗО с МПД и УУС осуществляется через стыки $С_3$. Эти стыки позволяют одновременно вести побайтную пере-

дачу 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. У них обеспечивается возможность автома-

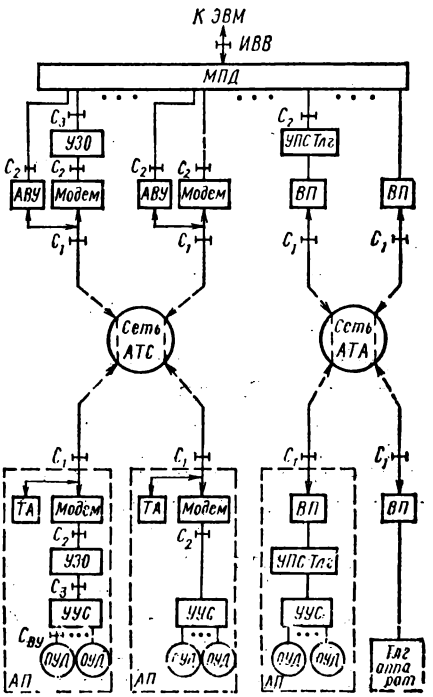


Рис. 15.2. Структуры связей с абонентскими пунктами через автоматическую телефонную станцию АТС и автоматическую станцию абонентского телеграфа АТА

тического 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 оконечными устройствами данных ОУД (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 автомати-

ческое соединение с абонентскими пунктами, посылая соответствующие запросы. Установление соединений со 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 вводо-выводное устройство — «терминал» — быстродействующий буквопечатающий аппарат со специальной клавиатурой, имеющий больше знаков, чем обычный телеграфный), 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 (оконечными устройствами) ввода-вывода специализированного или универсального типа. Средства телеобработки на основе ЕС ЭВМ комплектуются универсальными оконечными устройствами. В состав их могут входить:

телетайпы;

пишущие машинки;

быстродействующие печатающие устройства типа АЦПУ;

устройства перфорации и ввода-вывода ленточного типа;

Таблица 15.1

Устройства ввода-вывода информации для абонентских пунктов ЕС ЭВМ

Устройство	Шифр	Краткая техническая характеристика
Устройство ввода с перфокарт	ЕС-6111	**Скорость считывания — до 50 колонок в 1 с** Носитель информации — 80-колонная перфокарта Режим работы — стартстопный с остановом на каждой колонке Способ считывания — фотоэлектрический Направление движения перфокарты — прямое и обратное (управление от внешних сигналов или панели управления) Реверс возможен после считывания 80-й колонки
Устройство ввода с перфокарт	ЕС-6012	**Скорость считывания — до 500 карт в 1 мин** Носитель информации — 80-колонная перфокарта Режим работы — стартстопный, покартный Способ считывания — фотоэлектрический, считывание — поколонное
Устройство ввода с перфолент	ЕС-6121	**Скорость считывания — до 300 строк в 1 с** Носитель информации — бумажная перфолента с 5 и 8 дорожками Режим работы — стартстопный Способ считывания — фотоэлектрический
Устройство ввода с перфолент и карт с краевой перфорацией	ЕС-6191	**Скорость считывания — до 40 знаков в 1 с в стартстопном режиме и до 100 знаков в 1 с в непрерывном режиме** Носители информации — бумажные перфоленты с 5 и 8 дорожками и карты шириной 76, 2 мм или 82,6 мм с краевой перфорацией Режим работы — стартстопный и непрерывный Направление движения носителя — прямое и обратное как для лент, так и для карт с краевой перфорацией
Устройство вывода на перфокарты	ЕС-7111	**Скорость перфорации — до 50 колонок в 1 с** Носитель информации — 80-колонная перфокарта Режим работы — стартстопный с остановом на каждой позиции
Устройство вывода на перфокарты	ЕС-7012	**Скорость перфорации — до 250 карт в 1 мин** Носитель информации — 80-колонная перфокарта (перфорация по широкой стороне карты) Режим работы — стартстопный, покартный
Устройство вывода на перфоленту	ЕС-7121	**Скорость перфорации — 150—200 строк в 1 с** Носитель информации — бумажная перфолента с 5 и 8 дорожками Режим работы — асинхронный Способ перфорации — электромеханический

Продолжение табл. 15.1

Устройство	Шифр	Краткая техническая характеристика
Устройство вывода на перфоленты и карты с краевой перфорацией	ЕС-7191	**Скорость перфорации — 33—50 знаков в 1 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-7032	**Скорость печати — до 900 строк в 1 мин** Носитель информации — стандартная фальцованная бумага с краевой перфорацией и подсечками Количество разрядов в строке — 128 Количество знаков (символов) в разряде — 84
Пишущая машинка «Консул-260.1»	ЕС-7172	**Скорость печати — до 15 знаков в 1 с** Носитель информации — стандартная рулонная бумага Кодирование информации — по ГОСТ 13052—74 Количество печатаемых символов — 93 на двух регистрах Количество печатных знаков в строке — 106
Печатающее устройство	ЕС-7934	**Скорость печати — не менее 40 символов в 1 с** (в ЕС-7936 — 66 символов в 1 с); Способ печати — матричный, игольчатый (матрица 7 × 9) Количество изображаемых символов — 96 Количество символов в строке — 132
Устройство отображения информации на электронно-лучевой трубке с групповым управлением — ЕС-7921	ЕС-7927	**Скорость обмена данными по Тлф ЛС — 600, 1200, 2400, 4800 бит/с** Количество одновременно отображенных знаков: в модели ЕС-7927-00 — не менее 480, в модели ЕС-7927-01 — не менее 1920
Устройство отображения информации на ЭЛТ с функциональной и алфавитно-цифровой клавиатурой (ЕС-0101-1-01...04)	ЕС-7925	**Скорость обмена данными по Тлф ЛС — 600, 1200, 2400, 4800, 9600 бит/с** Количество одновременно отображенных знаков: в модели ЕС-7925-00 — до 480 символов, в модели ЕС-7925-01 — до 1920 символов 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 в 1 с** Количество различных вводимых символов — до 96 Плотность записи на магнитной ленте — 8 строк на 1 мм Максимальная длина зоны — 180 знаков Длина магнитной ленты — 185 м

устройства перфорации и ввода-вывода карточного типа;
накопители на магнитных лентах;
индикаторы на электронно-лучевых трубках (дисплеи) с клавиатурой.

Краткие технические характеристики ряда образцов 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.

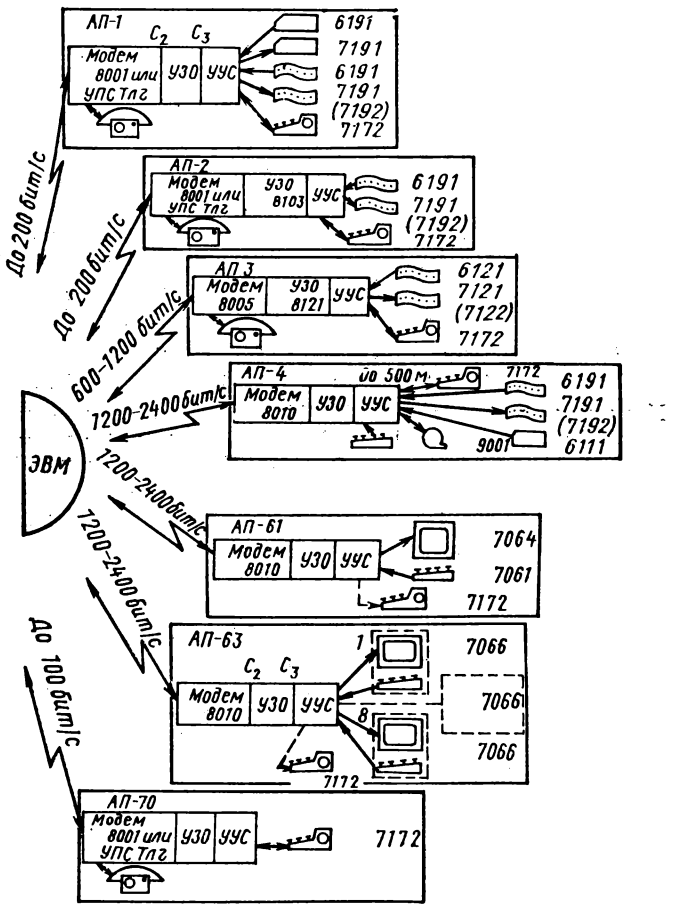


Рис. 15.3. Структура абонентских пунктов ЕС ЭВМ

Не считая самостоятельных абонентских пунктов на основе телеграфных аппаратов, связываемых с ЭВМ телеграфными линиями, для совместной работы с машинами Единой системы предусмотрены абонентские пункты (рис. 15.3) АП-1, АП-2, АП-3, АП-4, АП-61, АП-63, АП-70 и др. Пункты различаются возможными скоростями обмена данными, что обусловлено применяемой аппаратурой ввода-вывода и типами используе-

мых 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). АП-1, разработанный в НРБ, 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 обмениваться данными в полудуплексном режиме через все типы мультиплексоров ЕС ЭВМ.

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 и перечисленных выше носителей. Кроме пунктов типа АП-1, двухпунктовые и многопунктовые 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, будучи как 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 работать 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 ЕС-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 200 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 производятся 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 перфоленты или карта с краевой перфорацией возвращается к началу блока данных, переданных 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 число повторных попыток передачи определяется управляющей программой машины. В звеньях обмена данными непосредственно между абонетскими 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). АП-2, модель 1 которого разработана в ВНР, модель 2 в СССР, предназначается для обмена данными с ЭВМ Единой системы или аналогичным АП-2 по коммутируемым телефонным каналам связи со скоростью до 200 бит/с. Данные могут передаваться синхронно в полудуплексном или дуплексном режиме. 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 сопрово-

ждается выдачей пары символов: АР1 КП. Эти символы посылает в линию связи та сторона, по инициативе которой происходит разъединение.

Абонентский пункт АП-3 (ЕС-8503). АП-3, разработанныйв ВНР и НРБ, предназначается 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. В этом режиме 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). АП-4, разработанный в СССР, предназначен для работы с ЭВМ в двухпунктовых звеньях данных при связи через некоммутируемые телефонные каналы с четырехпроводным окончанием. Скорость обмена данными может достигать 2400 бит/с. Режим обмена — полудуплексный.

Помимо устройств ввода-вывода перфоленточного типа, устройства ввода с перфокарт и печатающего устройства 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. Связь устройств ввода-вывода с групповым устройством управления осуществляется через 22-проводной интерфейс — «Интерфейс-Т».

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. Кроме того, абонентский пункт может осуществлять обработку данных. Для этого устройство управления УУС выполнено программируемым. Оно по сути дела представляет собой несложное вычислительное устройство, имеющее в своем составе арифметико-логический блок с однобайтным сумматором, магнитное оперативное запоминающее устройство (16—32К байт) с циклом обращения 3 мкс и временем выборки 0,7 мкс, блок шести базовых и индексных регистров, блок синхронизации и управления, блок сопряжения с устройствами ввода-вывода, блок сопряжения с линией связи, пульт управления и сигнализации. Блок управления обеспечивает выполнение 51 команды. Команды — одно- и двухадресные.

Программируемое устройство управления в режиме обработки обеспечивает возможность осуществления сортировки и сжатия данных при формировании сообщений, выдаваемых в линию связи, редактирования информации перед выводом ее на устройства вывода, перекодирование информации. С помощью программ, находящихся в ОЗУ, можно обрабатывать данные, находящиеся на разных носителях, осуществлять перезапись информации с одного устройства на другое. Данные в ОЗУ адресуются с точностью до байта. Поле памяти разбито на страницы по 256 байт каждая.

В режиме подготовки данных информация, вводимая с клавиатуры

устройства последовательной печати, временно запоминается в ОЗУ. Формируемые таким образом сообщения затем выводятся на магнитную ленту.

В режиме обмена данными с ЭВМ абонентский пункт либо принимает информацию, либо выдает ее в канал связи. При передаче в канал связи могут выдаваться как заранее подготовленные сообщения, так и срочные, непосредственно набираемые на клавиатуре печатающего устройства. Категория срочности специальным образом указывается. При этом более срочные сообщения могут прерывать выдачу менее срочных, например, передаваемых массивами.

Абонентский пункт АП-61 (ЕС-8561). АП-61, разработанный в СССР, предназначается для работы с ЭВМ главным образом в режиме «диалога». Такие пункты оборудуются специальными пультами с функциональными и алфавитно-цифровыми клавиатурами, связанными с устройствами отображения информации на экранах электронно-лучевых трубок, и пишущими машинками.

Вводимая с клавиатуры пульта 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 осуществляется либо строками, в которых может находиться до 80 знаков, либо целыми сообщениями длиной до 960 знаков. Передаваемые данные кодируются стандартным 7-элементным 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). АП-63, разработанный в СССР, аналогично пунктам АП-61 в основном предназначается 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. Различие их 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 всегда в режиме подчинения 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. Если в качестве АП-63 используется групповой дистанционный комплекс ЕС-7920-10, то индикаторы и печатающие устройства, общее количество которых может достигать 32, располагаются на расстоянии до 1200 м от УУС.

Абонентский пункт АП-70 (ЕС-8570). АП-70, разработанный в НРБ и СССР, 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 в режиме соперничества.

По составу оборудования — это самый дешевый пункт. При отсутствии необходимости в передаче или приеме данных машинка типа «Консул-260.1» 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 использованию пунктов АП-70.

Режимы работы абонентских пунктов. 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 ЭВМ многих устройств ввода-вывода, а также 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 временно́е уплотнение информации во внутренних кодовых магистралях машины.

Телефонные и телеграфные каналы являются бо́лее медленно действующими цепями обмена информацией по сравнению с линиями связи устройств ввода-вывода, непосредственно подключенными к машине. По Тлф и Тлг линиям связи данные передаются последовательно, причем информационные посылки перемежаются со служебными (адресными, управляющими и т. п.). Поэтому, помимо временно́го 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 предусмотрено несколько типов мультиплексоров, среди которых отечественной промышленностью выпускаются МПД-1А, МПД-2 и МПД-3, технические параметры и возможности применения которых указаны в табл. 15.2.

Таблица 15.2

Технические параметры и возможности применения МПД ЕС ЭВМ

Тип МПД	Диапазон скоростей передачи данных, бит/с	Число обслуживаемых каналов связи в режимах передачи		Используемые линии связи	Обслуживаемые абонентские пункты	МПД, с которыми возможен обмен данными	АПД, возможная для использования в линиях связи	Модели ЭВМ, совместно с которыми преимущественно используются МПД
		полудуплексных	дуплексных					
МПД-1А ЕС-8400	50–4800	15	7	Тлг коммутируемые и некоммутируемые, Тлф коммутируемые и некоммутируемые	АП-1, АП-2, АП-61, АП-63, АП-70, Тлг А	МПД-1А	Модем-200, Модем-2400, Модем-4800	ЕС-1020, ЕС-1022, ЕС-1030, ЕС-1033
МПД-2 ЕС-8402	50–4800	8–176	4–88	Тлг коммутируемые и некоммутируемые Тлф коммутируемые и некоммутируемые	АП-1, АП-2, АП-3, АП-4, АП-61, АП-63, АП-70, Тлг А	МПД-1А, МПД-2, МПД-3	Модем-200, Модем-1200, Модем-2400, Модем-4800, АВУ Тлф, УЗО-200, УЗО-1200	ЕС-1033, ЕС-1050(52), ЕС-1060
МПД-3 ЕС-8403	50–48 000	1–4	1–2	Тлг коммутируемые и некоммутируемые, Тлф коммутируемые и некоммутируемые, широкополосные	АП-1, АП-2, АП-3, АП-70 Тлг А	МПД-2, МПД-3	Модем-200, Модем-1200, Модем-2400, Модем-4800, Модем-48000, АВУ Тлф, УЗО-1200 УЗО-48000	ЕС-1030, ЕС-1033, ЕС-1040, ЕС-1050(52), ЕС-1060.

Примечание. Мультиплексоры МПД-1А, МПД-2, МПД-3 могут работать как в двухпунктовых, так и в многопунктовых звеньях данных.

Мультиплексор передачи данных МПД-1А (ЕС-8400). Это устройство предназначается для работы с абонентскими пунктами АП-1, АП-2, АП-61, АП-63, АП-70, телеграфными аппаратами и другими МПД-1А. 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 и обеспечивает:

установление соединений с абонентами;

прием (передачу) данных по линиям связи;

адаптирование информации, получаемой из линии связи или из мультиплексного канала;

ввод (вывод) данных в мультиплексный канал.

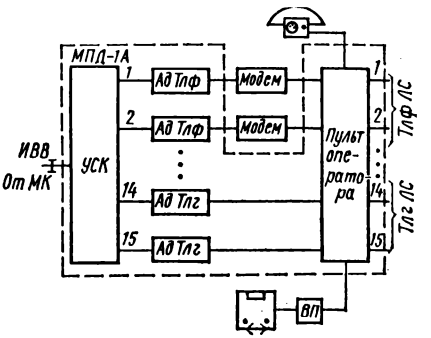


Рис. 15.4.
Структурная схема МПД-1А

Структурная схема МПД-1А показана на рис. 15.4. В состав мультиплексора входят:

устройство сопряжения с мультиплексным каналом (УСК);

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 телефонным каналам и 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: Ад-1, Ад-2, Ад-3, Ад-4, Ад-6.

Адаптеры Ад-1 предназначаются для управления обменом данными с телетайпами 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 и Ад-6 используются при работе с

абонентскими пунктами АП-61 и АП-63. Через МПД-1А связь с этими АП возможна по коммутируемым и некоммутируемым телефонным линиям.

Для обмена данными адаптеры через УСК подключаются к интерфейсу мультиплексного канала. УСК передает от канала к адаптеру команды, определяемые программой машины, а в обратном направлении — байты состояния и уточненного состояния (адаптера 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.

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.

Таблица 15.3

Основные команды ввода-вывода мультиплексора

Название команды	Код команды на ШИН-К (К 0123 4567)	Действия, выполняемые по команде
ЧТЕНИЕ	0 0000 0010	Происходит прием данных из линии связи, формирование информационных байт и передача их в мультиплексный канал по ШИН-А
ЗАПИСЬ	0 0000 0001	Происходит передача данных, поступающих из мультиплексного канала по ШИН-К, в линию связи
ПРОВЕРИТЬ ВВОД-ВЫВОД	1 0000 0000	Байт состояния, хранящийся в регистре состояния адаптера, передается в мультиплексный канал по ШИН-А. Байт уточненного состояния из регистра уточненного состояния адаптера передается в мультиплексный 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

Содержание байтов состояния и уточненного состояния дано в табл. 15.4 и 15.5.

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.

Таблица 15.4

Байт состояния

Номер бита байта	Значение единичного состояния бита	Примечание
0	Внимание	
1	Модификатор состояния	
2	Конец работы УВУ	
3	Занято	Устанавливается, когда от канала поступила новая команда, а Ад еще выполняет предыдущую
4	Конец работы канала	
5	Конец работы устройства	
6	Сбой в устройстве	Устанавливается при возникновении необычной ситуации, сопровождающейся указателем в байте уточненного состояния
7	Особый случай	Устанавливается, когда из линии связи принят символ «Конец передачи» или отрицательный ответ на опрашивание

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

Байтам информации, передаваемым в канал, присваивается контрольный разряд (9-й), приводящий количество единиц в девяти 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 (стандартного интерфейса ввода-вывода ИВВ).

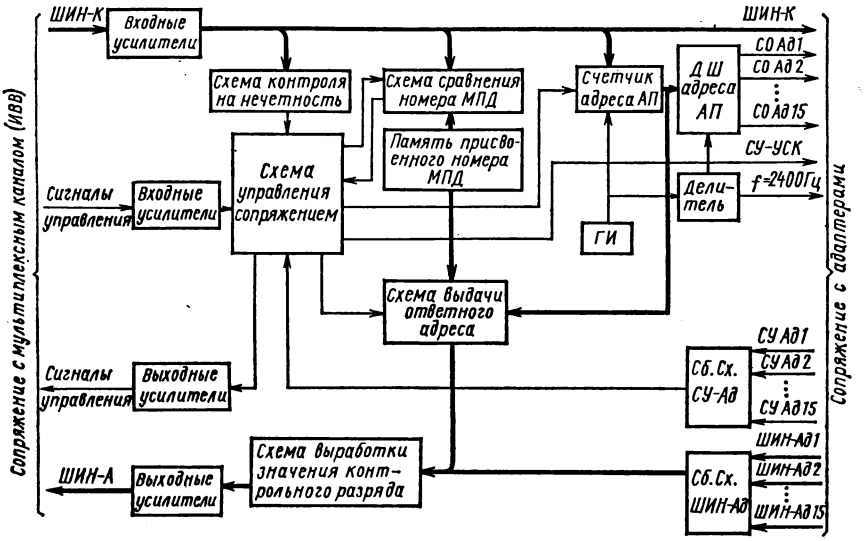


Рис. 15.5. Структурная схема устройства сопряжения с каналом

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 закоммутированным 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.

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.

После приема и анализа 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 (сканирование) 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, восприняв 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. Импульсы с ГИ с частотой 153,6 кГц вновь начинают поступать на вход счетчика адреса АП. Возобновляется сканирование цепей сигналов опроса СО Аді с целью обнаружения следующего адаптера, который, находясь в активном режиме, готов передать каналу или получить от него очередной байт данных.

По окончании операции ввода-вывода через какой-либо адаптер, как и в ответ на полученный код команды в процессе начальной выборки, происходит автоматическая *выдача каналу байта состояния*. 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 байт состояния, формируемый 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 и 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 случае 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 посылки 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 (см. табл. 15.3). При исполнении таких команд процедуры начальной выборки адаптера и обмена данными выполняются аналогично рассмотренному выше.

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 мультиплексору.

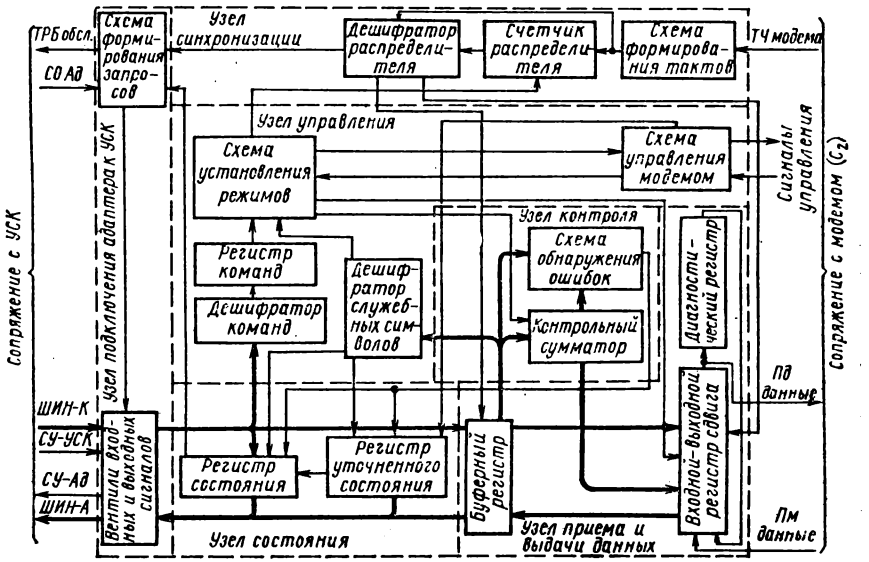


Рис. 15.6. Структурная схема телефонного адаптера

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, приходящему из УСК, открываются вентили входных или выходных сигналов, тем самым разрешая прохождение информации от УСК или обратно.

Во входно-выходном регистре сдвига узла приема и выдачи данных информация, идущая к модему, преобразуется из параллельного 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. Таким образом, тактовая синхронизация принудительно осуществляется модемами.

Адаптер Ад-3 соединяется с модемом через стык C_2 серии 100, содержащий цепи (табл. 15.6): данных, управления, синхронизации и заземления.

Таблица 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	+	—
Аппаратура передачи данных готова	107	+	—
Подключить АПД 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. Для передачи в дуплексном режиме необходимо задействование двух адаптеров.

В узле состояния формируется информация о состоянии адаптера, модема, АП. Эта информация (см. 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 байтов передаваемых в ЛС блоков данных осуществляется в узле контроля. Основу его составляют восемь параллельно работающих сумматоров по mod 2 накапливающего типа и схема обнаружения ошибок. Она содержит схему анализа сумматоров 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 по mod2 в одноименных разрядах всех байт каждого проходящего через адаптер блока данных. Счет и накопление осуществляются в указанных выше сумматорах.

При передаче в ЛС контрольный байт, полученный в результате суммирования по mod 2 одноименных разрядов всех передаваемых байт, сопровождает блок данных (передается последовательно за символом КТ – «Конец текста»). Когда же идет прием из ЛС, к полученному с помощью сумматоров контрольному байту поразрядно по mod 2 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 каждого байта или при контроле правильности приема целых 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 процессами приема и передачи данных происходит по командам, принимаемым адаптером через УСК из мультиплексного канала, а также по управляющим символам (см. табл. 4 приложения), сопровождающим передаваемую информацию от канала или из ЛС. Управляющие символы предназначаются для управления процессами передачи сообщений и осуществления необходимых переключений в мультиплексорах и другой аппаратуре звеньев данных. Они кодируются согласно таблице стандартного семиэлементного кода КОИ-7.

Команда, приходящая в адаптер из мультиплексного канала, сопровождается стробирующим сигналом ПРИНЯТЬ КОМАНДУ, формируемым в УСК. По стробирующему сигналу команда 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 управляющих символов НТ – «Начало текста» или НЗ – «Начало заголовка». Триггер сбрасывается с появлением на входах дешифратора кодовых комбинаций, соответствующих управляющим символам: КБ – «Конец блока» или КТ – «Конец текста» и при приеме (передаче) контрольной последовательности блока КПБ или текста КПТ. Режим текста сбрасывается также при приеме символа «Забой». При выходе из режима текста по окончании передачи, вслед за выдачей в канал связи КПБ, адаптер автоматически переключается на режим ПРИЕМ.

На узел управления также возлагаются функции установления символьной синхронизации (синхронизация приема-передачи байт). При каждом

изменении направления передачи ввод в синхронизм инициируется установкой в единичное состояние триггера, вырабатывающего сигнал ФАЗИРОВАНИЕ. Этот триггер сбрасывается в нулевое состояние по окончании фазирования, после установления символьной синхронизации.

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 переходит к опросу следующих адаптеров и, следовательно, абонентских пунктов.

Таким образом, для адаптера характерными являются два режима: прием и 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, например имеет место нарушение четности, в регистр уточненного состояния заносится признак КОМАНДА ОТВЕРГНУТА, а в регистр состояния – СБОЙ В УСТРОЙСТВЕ. Признак КОМАНДА ОТВЕРГНУТА устанавливается также, когда устройство занято выполнением другой команды.

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 ИНФ-К, сбрасывающий в УСК сигнал РАБ-А. Происходит сброс в нулевое состояние регистра состояния и начинается 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, образующее блок данных, в общем случае имеет следующий формат СИН СИН НТ < текст > КТ КПТ. В рассматриваемом адаптере контрольная последовательность КПТ – контроль-

ный байт, формируется по матричному способу контроля. В зависимости от особенностей каналов связи символов СИН может быть больше двух. Символы СИН и НТ или НЗ при формировании контрольной последовательности текста или блока не учитываются.

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, считчик 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 (см. 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, например, 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 mod2 с содержимым сумматора. Если полученная сумма равна нулю, то операция заканчивается занесением в регистр состояния только 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 посылается 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 входно-выходной регистр сдвига, а каналу дается запрос на следующий байт данных. Запрос возникает при установке в адаптере в единичное состояние триггера, вырабатывающего сигнал ТРБ-Д. Получив этот сигнал, УСК посылает в мультиплексный канал сигнал ТРБ-А. 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 сформированнный 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 мультиплексному каналу после выработки признаков КОНЕЦ РАБОТЫ КАНАЛА и КОНЕЦ РАБОТЫ УСТРОЙСТВА. При этом в адаптере специальным триггером вырабатывается сигнал

«Требование выдачи состояния» ТРБ-С, который поступает в УСК. Обнаружив этот сигнал во время опроса адаптера, УСК выдает каналу сигнал ТРБ-А, если от канала нет сигнала блокировки БЛК-К. 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;

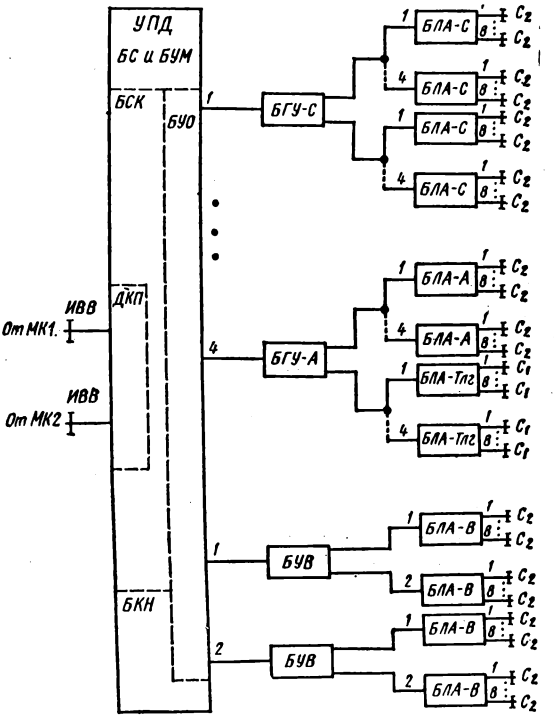


Рис. 15.7. Структурная схема МПД-2

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.

Мультиплексор передачи данных МПД-2 (ЕС-8402). Этот мультиплексор обеспечивает возможность работы с абонентскими пунктами всех типов, а также, при обмене информацией между машинами, со всеми из указанных выше мультиплексорами, в том числе и другими МПД-2. Связь может осуществляться по низко- и среднескоростным каналам связи. Требование необходимости обслуживания большого числа каналов связи (см. табл. 15.2) обусловило более сложную структуру мульти-

плексора (рис. 15.7) с многоступенчатой, многоцикловой организацией управления.

Мультиплексор МПД-2 в своем составе содержит:

блоки линейных адаптеров, синхронных БЛА-С, асинхронных БЛА-А и вызова БЛА-В;

блоки группового управления синхронными БГУ-С и асинхронными БГУ-А адаптерами;

блоки управления вызовом БУВ;

устройство управления передачей данных УПД.

Массовыми, завершенными в функциональном отношении являются блоки линейных адаптеров. Они осуществляют побитный прием (передачу) данных по каналам связи. Обмен данными по телефонным каналам осуществляется через модемы, которые подключаются к выходам линейных адаптеров через стыки C_2 серии 100 (см. табл. 15.6). Каждый блок синхронных и асинхронных адаптеров может обслуживать до восьми каналов связи. Поэтому он содержит восемь линейных адаптеров и общий для них узел управления. Последний обеспечивает выбор адресуемого адаптера и управление приемом (передачей) им очередного бита данных. Принимаемый бит транслируется в БГУ, который, накапливая их, формирует байты данных, соответствующих обслуживаемым направлениям. При передаче очередные байты данных, находящиеся в БГУ, побитно через линейные адаптеры выдаются в каналы связи. Таким образом, линейный адаптер содержит в основном триггеры и логические схемы, обеспечивающие сопряжение блока группового управления с одним из каналов связи через стандартный стык C_2. Они функционируют в соответствии с принятыми для этого стыка алгоритмами обмена в различных режимах информационными, управляющими и индикационными сигналами. Асинхронные линейные адаптеры БЛА-А и БЛА-Тлг могут производить обмен данными со скоростями 50, 100 и 200 бит/с. Для синхронных адаптеров БЛА-С возможны скорости обмена 1200, 2400 и 4800 бит/с.

Количество синхронных и асинхронных адаптеров, включаемых в МПД-2, зависит от количества и скоростных показателей обслуживаемых каналов связи при учете того, что максимальная пропускная способность мультиплексора не может превышать 40 кбайт/с. Поэтому, например, при скорости передачи 1200 бит/с мультиплексор может работать со 176 каналами связи, при 2400 бит/с – со 120 каналами и при 4800 бит/с – с 64 каналами в полудуплексном режиме. В дуплексном режиме в каждом из указанных случаев количество подключаемых каналов в 2 раза меньше, так как на каждое направление задействуется по два адаптера.

Как показано на рис. 15.7, к каждому БГУ может быть подключено до восьми блоков БЛА. Всего в МПД-2 может одновременно использоваться до четырех блоков БГУ.

Блоки группового управления обеспечивают сопряжение блоков линейных адаптеров с УПД. В БГУ осуществляется буферное хранение байт данных для каждого из работающих направлений. В процессе приема (передачи) в БГУ реализуются процедуры символьной синхронизации.

В режиме начальной выборки через БГУ производится адресный выбор нужного БЛА. Для этого в УПД формируется 6-разрядный код адреса. Старшие три разряда служат для выбора необходимого БГУ, а младшие — активизируемого линейного адаптера. Между процедурами установления соединений и задания режимов работы адаптеров блоки БГУ с помощью счетчиков адресов осуществляют сканирование «своих» линейных адаптеров для обслуживания их запросов на прием или передачу очередных битов данных.

Во время соединения с линейным адаптером управляющая информация о режиме работы и направлении передачи берется из 32-разрядного управляющего слова БГУ. В разрядах этих слов также накапливаются принимаемые из каналов связи байты данных, либо временно запоминаются байты данных, передаваемых в каналы связи. Для хранения управляющих слов в каждом БГУ имеется полупроводниковая малая оперативная память емкостью 64 32-разрядных двоичных слова, цикл записи-считывания которой не превышает 1 мкс. Полный цикл обработки одного управляющего слова в БГУ равен 8 мкс.

Это время составляет внутренний цикл работы БГУ, в течение которого осуществляется обмен информацией с выбранным адаптером. Кроме этого, имеется внешний цикл работы БГУ, определяемый устройством УПД. Этот цикл определяет периодичность опроса устройством передачи данных блоков БГУ. Временные соотношения внутреннего и внешнего циклов выбираются таким образом, чтобы после каждых восьми соединений БГУ с адаптером происходило соединение БГУ с УПД для обмена очередным байтом данных. В то же время частота внешних циклов должна быть такой, чтобы за время между очередными двумя соединениями УПД с БГУ последний мог по крайней мере по одному разу опросить все 64 направления.

В свою очередь, УПД в режимах приема-передачи данных, кроме связи с БГУ, периодически соединяется с мультиплексным каналом. Это внешние циклы работы УПД. Связь с мультиплексным каналом может осуществляться и в произвольные моменты времени, например, по требованию машины, когда необходимо включить или выключить звено данных, проверить ввод-вывод, уточнить состояние, зарезервировать или освободить канал МПД и т. п.

В составе устройства УПД имеются:

блок стандартного сопряжения с мультиплексным каналом ввода-вывода БСК;

блок общего управления БУО;

блок микропрограммного управления БУМ;

блок синхронизации БС;

блок контроля и наладки БКН.

Через БСК УПД сопрягается с одним или двумя мультиплексными каналами соответственно одной или двух ЭВМ ЕС. Связь с каналами осуществляется через стандартные интерфейсы ввода-вывода. Для обеспечения возможности подключения к двум каналам в БСК имеется двухканальный переключатель ДКП. С его помощью возможно, например, поочередное автоматическое подключение к каждому из каналов.

Управление взаимодействием всех блоков УПД осуществляет БУО.

В этом блоке имеется ОЗУ емкостью 1024 36-разрядных слова. В нем хранятся управляющие УСЛ и информационные ИСЛ слова линий. Для каждого направления связи предусмотрено два 36-разрядных УСЛ и два 36-разрядных ИСЛ. Эти четыре слова выбираются из ОЗУ в регистры БУО при каждом установлении связи либо с БСК, либо с БГУ. В обоих случаях формируется 10-разрядный код адреса. В нем два старших разряда определяют область памяти, закрепленную за группой адаптеров, обслуживаемых одним БГУ, шесть следующих разрядов – адрес, соответствующий выбираемой линии связи БГУ, и два младших разряда определяют адрес слова (00 – УСЛ1, 01 – УСЛ2, 10 – ИСЛ1 или 11 – ИСЛ2). Длительность цикла работы БУО слагается из времени выборки из ОЗУ четырех слов: двух УСЛ и двух ИСЛ, их обработки и обратной записи в ОЗУ. Она равна 16 мкс.

В управляющих словах находятся коды команд, задающие и определяющие режимы работы направлений МПД, байты состояния и уточненного состояния, текущая контрольная последовательность передаваемого или принимаемого блока данных и другая информация, необходимая для организации работы и сигнализации о качестве функционирования соответствующих направлений МПД и звеньев данных в целом. В информационных словах находятся байты данных, проходящих через МПД по соответствующим направлениям. При этом особенность работы мультиплексора МПД-2 состоит в том, что с мультиплексным каналом он обменивается в групповом режиме. В каждом цикле соединения УПД с мультиплексным каналом через БСК в канал передается или из него принимается 4 байта информации. Это повышает пропускную способность интерфейса ввода-вывода и УПД мультиплексора.

Управление работой УПД при выполнении различных команд, распознавание различных управляющих символов и т. п. осуществляется блоком микропрограммного управления. Блок синхронизации вырабатывает

Таблица 15.7

Цепи стыка с АВУ

Номер цепи	Наименование цепи	Направление цепи		Номер цепи	Наименование цепи	Направление цепи	
		от МПД-2	к МПД-2			от МПД-2	к МПД-2
201	Сигнальное заземление			208	Цифровой сигнал (2^2)	+	–
202	Запрос вызова	+	–	209	Цифровой сигнал (2^3)	+	–
203	Линия данных занята	–	+	210	Запрос следующей цифры	–	+
204	Дальняя установка подключена	–	+	211	Цифра выдается	+	–
205	Прекратить вызов	–	+	212	Защитное заземление		
206	Цифровой сигнал (2^0)	+	–	213	Индикатор электропитания	–	+
207	Цифровой сигнал (2^1)	+	–				

синхроимпульсы различных серий, необходимые для координации работы блоков УПД, БГУ и БЛА. Контроль и проверка аппаратуры МПД в различных режимах осуществляются с помощью блока БКН.

Для работы по коммутируемым телефонным каналам МПД-2 может сопрягаться с автоматическими вызывными устройствами АВУ. Набор номера абонента и установление соединения в коммутируемом канале связи осуществляются с помощью блока управления вызовом БУВ. Блоки БУВ управляют АВУ с помощью блоков линейных адаптеров вызова БЛА-В. В МПД-2 может быть до двух блоков БУВ, каждый из которых может управлять двумя блоками БЛА-В. К каждому БЛА-В может быть подсоединено до 16 АВУ. Связь с АВУ осуществляется через стык C_2 серии 200 (табл. 15.7).

Мультиплексор передачи данных МПД-3 (ЕС-8403). Устройство предназначается для обеспечения связи машин с абонентскими пунктами по низко-, средне- и высокоскоростным каналам связи. Связь может осуществляться 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 обеспечения ее требуемой достоверности.

В зависимости от используемых линий связи и требований к достоверности передачи данных в составе АПД могут быть (см. 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. 15.8. Входящие в состав Единой системы модемы делятся на низко-, средне- 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.

Среднескоростными являются модемы 1200 и 2400. Они предназначаются для организации связи с абонентскими пунктами, содержащими более быстродействующую аппаратуру ввода-вывода (см. 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 полудуплексным режимом обмена информацией, то последние два — в звеньях с дуплексными каналами.

Аппаратура передачи

Устройство	Шифр	Краткая техническая характеристика
Устройство преобразования сигналов телеграфного типа (УПС Тлг)	ЕС-8030	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 2-проводным ЛС возможны полудуплексный и дуплексный режимы передачи Используется в ЛС с АП-1, АП-2, АП-70
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 — 600, 1200, 2400 бит/с Тлф ЛС — некоммутируемые 4-проводные Модуляция — фазовая Используется в ЛС с АП-61, АП-63 и между машинами
Модем-4800	ЕС-8015	Возможные скорости передачи данных — 2400 и 4800 бит/с Тлф ЛС — некоммутируемые 4-проводные Модуляция — фазовая Используется в ЛС между машинами Режим обмена — дуплексный
Модем-48 000	ЕС-8019	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 — 200, 600, 1200, 2400 и 4800 бит/с
Устройство защиты от ошибок с автозапросом (УЗО-4800)	ЕС-8135	Скорость передачи данных — 50—4800 бит/с
Устройство защиты от ошибок с автозапросом (УЗО-48 000)	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»

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 (см. рис. 15.1). Этим 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 серии 100. Пример такого стыка представлен в табл. 15.6. В случаях применения устройств защиты от ошибок УЗО соединения с ними выполняются 34-проводными стандартными стыками C_3.

Стык C_3 содержит девять цепей параллельной передачи к УЗО и девять цепей параллельного приема от него байт данных с контрольными (девятыми) битами. Остальные из 34 цепей используются для управляющих и идентифицирующих сигналов, таких как УПРАВЛЕНИЕ ПЕРЕДАТЧИКА АПД, УПРАВЛЕНИЕ ИСТОЧНИКА ОУД, УПРАВЛЕНИЕ ПОЛУЧАТЕЛЯ ОУД, УПРАВЛЕНИЕ ПРИЕМНИКА АПД, ОУД ГОТОВО, АПД ГОТОВА и т. п.

Устройства защиты от ошибок (табл. 15.8) 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 скоростей обмена данными. При использовании УЗО для помехозащитного кодирования применяют циклические коды, образующие полиномы которых выбирают 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 временно́е 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 следующих состояниях:

дежурном;

управления;

передачи данных.

В дежурном состоянии она следит за каналом связи с целью обнаружения управляющих служебных символов (см. табл. 4 приложения). 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 направ-

ления 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 как часть блока или промежуточный блок, не требующие 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, не требуя ответа о результате ее приема.

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,

где символы АР1 указывают начало и конец данного режима передачи.

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 («Разъединяюсь»). 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.

Глава 16

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 режима работы должно осуществляться постоянное слежение за техническим состоянием машины и правильностью передачи и преобразования в ней информации.

При обнаружении ошибки должны автоматически немедленно анализироваться. Если причиной их являются сбои, обусловленные кратковременным нарушением нормальной работы отдельных устройств машины, 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 элементов, нарушении 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 восстановления машин.

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 микрооперации (передачу и прием информации, дешифрирование, сдвиг, сложение и т. п.), так и их определенные последовательности, из которых слагаются микропрограммы операций. Ошибки в выполнении операций обнаруживаются в процессе работы устройств в темпе следования синхронизирующих сигналов. В процессорах реакцию на сигнал ошибки

стараются обеспечивать такой, чтобы время распространения последней ограничивалось 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 контролировать все схемы машины, включающие некоторые цепи управления и синхронизации, приема и выдачи информации и сигнализации, практически трудно. Для более полного решения этой задачи, наряду с аппаратными, применяют программные средства контроля.

Программный контроль осуществляется с помощью специальных тест-программ, имеющихся в комплекте программ технического обслуживания, входящем в состав общего программного 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-1033 и ЕС-1050(52) 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 локализировать неисправности с определенной разрешающей способностью, например, с точностью до типового элемента замены.

Программы технического обслуживания дают возможность машинам гибко реагировать на все обнаруживаемые ошибки и сигналы прерываний от системы аппаратного контроля, управлять автоматически диагностикой с помощью соответствующих тестов, накапливать и анализировать информацию о состоянии самой машины и особенностях функционирования устройств в различные периоды и в различных условиях эксплуатации, оперативно выдавать об этом необходимую информацию. Тест-программы 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 останова 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 продолжать

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 общем случае в систему обслуживания ЭВМ входят:

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;

управляющая программа (программа-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. Эта аппаратура нужна для предоставления дополнительной информации, без которой невозможен контроль. В устройствах машин Единой системы используются различные способы получения такой информации: либо повторным запросом или преобразованием, либо

дублированием аппаратуры, либо дополнением основной 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, специальных кодов и т. п., находят применение в качестве дополнительных средств контроля во внешних устройствах, в устройствах ввода-вывода и в аппаратуре передачи данных. На основе дублирования аппаратуры обычно организуется контроль АЛУ.

Для сравнительного анализа дополнительной информации также нужна специальная 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 памятями;
цепи передачи информации между блоками и 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.

Всего в процессоре и основной памяти этой модели 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, зафиксировавшему сигнал вида ошибки, и по триггеру, соответствующему номеру байта, в котором она обнаружена.

В процессоре ЕС-2433 значительно больше точек контроля. Узлы контроля, логически входящие в состав блока контроля БК, распределены по всей магистральной структуре процессора. Контролем охвачены все операционные блоки. В результате реализуется непрерывный сквозной аппаратный контроль преобразования и передачи информации внутри центрального процессора и обмена данными между ЦП и ОП. Не контролируются лишь схемы блоков выработки управляющих сигналов, формирования адресов и синхронизации.

Сигналы с выходов большинства схем контроля ЦП подаются на соответствующие триггеры 79-разрядного регистра машинных ошибок (РМО). По этим триггерам сразу определяются диагностирующие микропрограммы. Для ряда схем контроля существуют обобщенные сигналы ошибок, которые не имеют непосредственных выходов на 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 в ЭВМ, например при считывании информации с магнитных лент, когда для большей надежности дублируется запись либо производится контрольное считывание при записи; он также может быть применен при проверках правильности ввода с перфокарт, перфолент и т. п. При воспроизведении магнитной или перфорационной записи дублирующая информация в виде электрических сигналов одновременно с основной существует непродолжительное время, не превышающее периода следования считываемых кодов.

Контроль путем дублирования информации на всех этапах ее передачи и преобразования требует больших аппаратных затрат. Поэтому в современных ЭВМ обычно 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 какое-то число ошибок (одиночных, двойных, тройных и т. д.) или должна еще и исправлять некоторые из них.

Одним из простейших и экономических 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 $X = x_0 x_1 \dots x_{n-1}$ 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*.

Обозначив модуль при числовом контроле q, а при цифровом d, эквивалентность числа его остатку можно записать как

$$X = r_X \pmod{q},$$

а суммы цифр

$$\sum_{i=0}^{n-1} x_i = r_x (\mathrm{mod}\, d).$$

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 арифметических операций основан на известных из теории чисел тождествах:

$$\text{для суммы } X_1 + X_2 \equiv (r_{X_1} + r_{X_2})(\mathrm{mod}\, q);$$

$$\text{для произведения } X_1 X_2 \equiv (r_{X_1} r_{X_2})(\mathrm{mod}\, q);$$

$$\text{для деления } r_{X_2} r_{X_3} \equiv (r_{X_1} - r_{X_4})(\mathrm{mod}\, q),$$

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 mod 2. Однако в данном случае при основании системы счисления $N = 2$ числовой контроль при $q = 2$ не имеет смысла, так как он охватывает только младший разряд контролируемого числа (наименьший остаток от деления двоичного числа на два всегда равен значению младшего разряда делимого). Поэтому для двоичных кодов применяется цифровой контроль. В этом случае при $d = 2$ цифра контрольного разряда

$$r_x (\mathrm{mod}\, 2) = \left(\sum_{i=0}^{n-1} x_i \right) (\mathrm{mod}\, 2),$$

т. е. одноразрядный контрольный код определяется просто как сумма по mod 2 цифр числа X. Такой контроль, позволяющий обнаруживать все одиночные ошибки и групповые, число которых нечетно, широко используется в ЭВМ Единой системы. Каждый байт информации в них автоматически сопровождается девятым, контрольным разрядом.

Контроль по mod 2 может осуществляться проверкой либо четности, либо нечетности. В первом случае значение контрольного разряда формируется таким образом, чтобы сумма единиц в информационном слове, включая контрольный разряд, была всегда четной, а во втором случае — нечетной. Выбор вида проверки — на четность или нечетность зависит от способа представления контролируемой информации. При этом используется такая проверка, которая позволяет различать случаи нулевой информации и случаи отсутствия информации.

Пусть, например, отсутствию информации соответствуют сигналы кода 0 во всех разрядах, включая контрольный. В этом случае при использовании проверки нечетности в разрядах слова вместе с контрольным разрядом всегда будет хотя бы одна единица. Появление во всех разрядах, в том числе и контрольном, нулей будет свидетельствовать либо об ошибке, либо об отсутствии информации.

Если же отсутствию информации соответствуют сигналы кода 1 во всех разрядах, включая контрольный разряд, то целесообразной оказывается проверка на четность. При этом наличие сигналов кода 1 во всех

разрядах, кроме контрольного, будет свидетельствовать о нулевой информации. Такой случай контроля, например, имеет место в центральном процессоре ЕС-2433. В магистралях процессора значения информационных разрядов передаются в инверсной форме, т. е. нулевым значениям соответствуют сигналы кода 1 и наоборот. Когда на магистралях нет информации, во всех разрядах, включая контрольные, должны быть единицы.

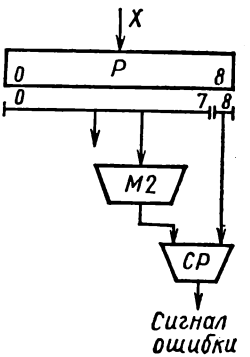


Рис. 16.1. Структурная схема контроля содержимого регистра

На рис. 16.1 показана структурная схема контроля двоичных кодов X, поступающих в 9-разрядный регистр Р. 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 mod 2 (М2) определяется сумма по mod 2 содержимого 0 – 7-го разрядов регистра, которая сравнивается схемой сравнения СР с содержимым 8-го разряда. При их совпадении схема сравнения вырабатывает сигнал ошибки.

Схема суммирования по mod 2 8-разрядных кодов с включенной на выходе схемой сравнения (рис. 16.2) выполнена трехкаскадной. Составляющие ее двухразрядные схемы М2 имеют прямые и инверсные выходы. На прямом выходе сигнал высокого уровня формируется тогда,

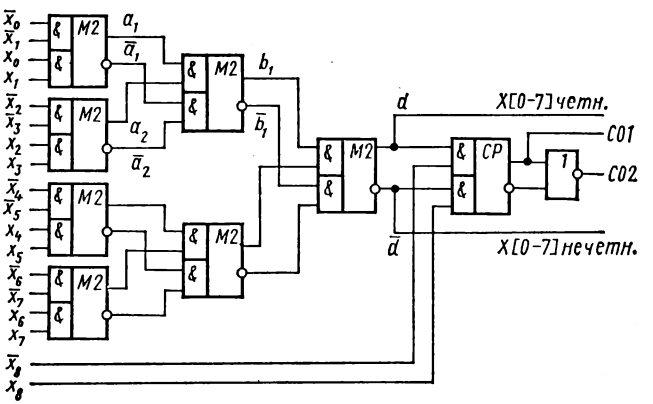


Рис. 16.2. Схема суммирования по mod2 8-разрядных кодов со схемой сравнения на выходе

когда сумма по mod 2 двух входных переменных – четная. Если схема М2 строится на элементах *И – ИЛИ – НЕ*, то выходные переменные a_1 и $\bar{a}_1$ вырабатываются, например, согласно формулам

$$a_1 = \overline{x_0\bar{x}_1 \vee \bar{x}_0 x_1};$$

$$\bar{a}_1 = \overline{\bar{x}_0\bar{x}_1 \vee x_0 x_1}.$$

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 СО1. Для выявления отказов в основном и контрольном оборудовании, в результате которых на выходах схемы СР могут появиться низкие уровни потенциала, на выходе имеется дополнительная схема *ИЛИ – НЕ*. Она вырабатывает сигнал ошибки СО2, когда на обоих выводах схемы сравнения устанавливаются сигналы низкого уровня. Возникновение сигналов высокого уровня на обоих выходах обнаруживается сигналом СО1

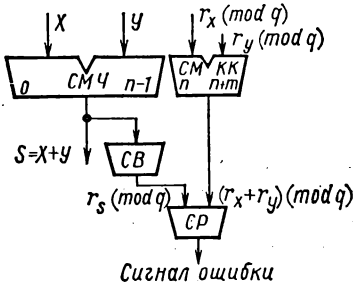


Рис. 16.3. Схема организации контроля по mod q процесса сложения

Арифметические и логические операции, выполняемые цифровыми машинами в процессе преобразования информации, слагаются из последовательностей элементарных действий – микроопераций – 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 применим для проверки правильности выполнения и таких действий. Правда, способы построения схем контроля в каждом конкретном случае имеют свои специфические особенности. Это обусловливается 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 числового контроля сложения требуются дополнительные схемы, показанные на рис. 16.3. Одновременно со сложением чисел X и Y в специальном сумматоре (СМКК) по mod q складываются их контрольные коды r_X и r_Y. Сумма $(r_X + r_Y) \times (\mathrm{mod}\, q)$ в схеме сравнения СР сравнивается с контрольным кодом результата сложения в сумматоре числа СМЧ. Этот контрольный код формируется схемой свертки СВ по mod q. При несовпадении суммы контрольных кодов с контрольным кодом результата сложения выдается сигнал ошибки.

Как было указано выше, в ЭВМ Единой системы применяется цифровой контроль по mod 2. Особенность проверки процесса суммирования при таком модуле состоит в том, что для полноты контроля процессов, происходящих в сумматоре СМЧ, недостаточно только сравнения значения контрольного кода суммы r_s с суммой $(r_x + r_y)(\mathrm{mod}\, 2)$. Необходимо еще учитывать и анализировать процессы, происходящие в цепях переносов.

Действительно, значение контрольного разряда r_s суммы двух байт определяется следующим соотношением:

$$r_s(\mathrm{mod}\,2) = s_0 \oplus s_1 \oplus \ldots \oplus s_7 = (x_0 \oplus y_0 \oplus p_1) \oplus (x_1 \oplus y_1 \oplus p_2) \oplus \ldots$$
$$\ldots \oplus (x_7 \oplus y_7 \oplus p_{вх}) = (x_0 \oplus x_1 \oplus \ldots \oplus x_7) \oplus (y_0 \oplus y_1 \oplus \ldots \oplus y_7) \oplus$$
$$\oplus (p_1 \oplus p_2 \oplus \ldots \oplus p_{вх}) = r_x(\mathrm{mod}\,2) \oplus r_y(\mathrm{mod}\,2) \oplus r_p(\mathrm{mod}\,2), \quad (16.1)$$

где $x_0, \ldots, x_7, y_0, \ldots, y_7$ — значения разрядов слагаемых; $s_0, \ldots, s_7$ — значения разрядов суммы; $p_1, \ldots, p_{вх}$ — переносы между разрядами и входной перенос в младший (7-й) разряд; r_x, r_y, r_s — символы в контрольных разрядах слагаемых X, Y и суммы S; r_p — сумма по mod 2 возникающих при сложении переносов.

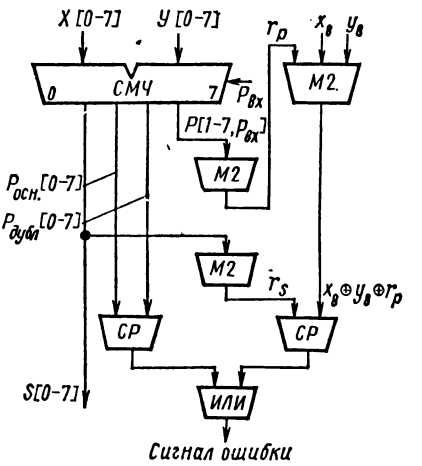


Рис. 16.4. Схема организации контроля по mod2 сложения кодов двух байтов

Из этого соотношения следует, что при данном контроле необходимо регистрировать возникающие в процессе сложения переносы, суммировать их по mod 2 и определять значение r_p. Это значение должно суммироваться по mod 2 с x_8 (r_x) и y_8 (r_y) в сумматоре М2, как показано на структурной схеме рис. 16.4.

Помимо одиночных ошибок, контроль по mod 2 позволяет обнаруживать нечетные количества групповых ошибок. Четные количества ошибок при таком контроле не улавливаются. Последнее обстоятельство приводит к тому, что при контроле на четность могут пропускаться ошибки, возникающие в цепях переносов сумматоров. Каждая такая ошибка в переносе p_{i+1}, даже не распространяясь далее одного разряда, вызывает как минимум еще одну ошибку в разряде суммы s_i. Это равносильно прибавлению по mod 2 к левой и правой частям (16.1) по единице, от чего равенство не нарушается.

Чтобы исключить возможность пропуска таких ошибок, в цепи сумматоров вводят дополнительные элементы, позволяющие контролировать процессы возникновения и распространения переносов. Одним из простейших способов осуществления дополнительного контроля является дублирование цепей формирования переносов. Дополнительные компоненты аппаратуры контроля, предназначенные для слежения за правильностью работы цепей переносов, показаны на структурной схеме рис. 16.4.

При построении сумматоров на интегральных схемах, подобных рассмотренной на рис. 4.43, затруднительно дублирование цепей межразрядных переносов. Поэтому в АЛУ, строящихся на таких интегральных схемах, применяют полное или погрупповое дублирование АЛС. Погрупповое

дублирование имеет место в арифметико-логическом блоке процессора ЕС-2433 (см. рис. 6.12).

В этом блоке все АЛСО – АЛС7 задублированы, как показано на примере i-й группы на рис. 16.5. Вспомогательные АЛС соединены цепями параллельного переноса аналогично основным (см. рис. 6.13). Они образуют вспомогательный БАЛ, работающий параллельно основному. Сигналы, соответствующие значениям разрядов операндов X и Y, подаются

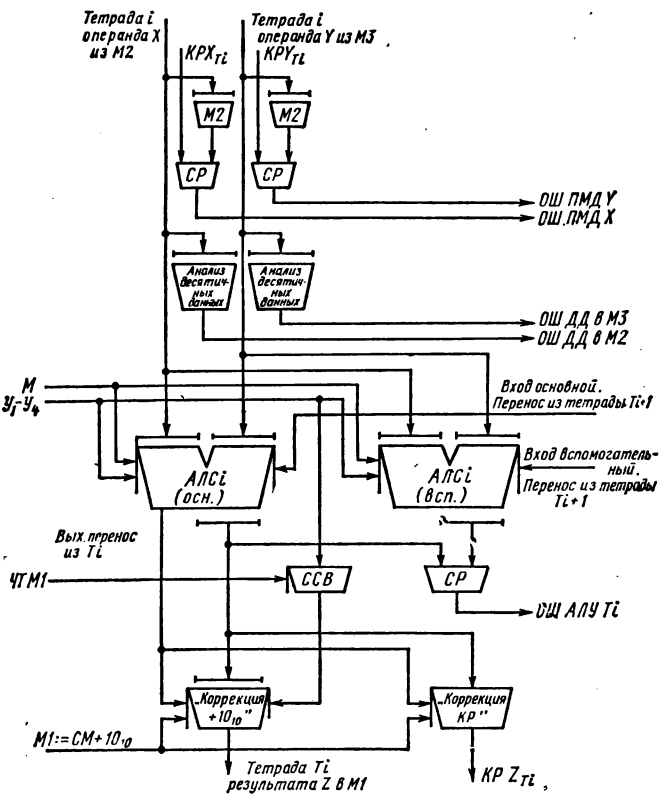


Рис. 16.5. Структурная схема 4-разрядной части БАЛ с дополнительными цепями дублирования и контроля

параллельно на входы основных и вспомогательных АЛС. Сигналы управления M и $У_1 - У_4$ поступают на управляющие входы также параллельно. Тетрады результатов по сигналу чтения ЧТ М1 выдаются в магистраль 1 через схему «Коррекция + 10» с выходов основной АЛС. При этом схемами сравнения СР осуществляется поразрядное сравнение результатов основных и вспомогательных арифметико-логических схем. При несовпадении результатов вырабатывается обобщенный для каждой i-й тетрады T_i сигнал ошибки ОШ АЛУTi. Этот сигнал заносится в регистр машинных ошибок РМО.

Кроме поразрядного сравнения результатов, в БАЛ применены и другие виды контроля. Для всех поступающих на входы тетрад T_i операндов X и Y сумматорами М2 определяются значения контрольных разрядов.

Они затем сравниваются схемами СР. При обнаружении ошибки в принимаемых данных для каждой тетрады вырабатываются сигналы ОШ ПМД *X* и ОШ ПМД *Y*. Кроме этого, специальными схемами анализа на входах проверяются и десятичные данные. При обнаружении двоичных тетрад, значения которых превышают 10 (1010_2), вырабатываются сигналы ошибок в десятичных данных, принимаемых из магистралей 2 и 3, – ОШ ДД магистрали 2 и ОШ ДД магистрали 3. В отличие от предыдущих потетрадных сигналов ошибок последние сигналы вырабатываются обобщенными, для 32-разрядных слов. На выходах БАЛ схемами коррекции КР для каждой тетрады вырабатывается контрольный разряд, например, при сложении двоичных чисел согласно (16.1). Эти значения, затем формируемые для байтов, позволяют следить за четностью результатов, передаваемых в регистры через магистраль 1.

Аппаратный контроль передачи и преобразования данных обеспечивает возможность осуществления быстрой реакции через систему прерывания на возникающие сигналы ошибок. Большое число точек контроля, слежение за четностью не только байт, но в ряде случаев и тетрад, в свою очередь способствует построению системы диагностики отказов, обладающей высокой разрешающей способностью.

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 обеспечивается 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 предназначаются 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. Вначале вручную проверяется минимум аппаратуры, позволяющей вводить управляющую программу, с помощью которой затем загружаются и выполняются тесты второго уровня.

В модели ЕС-1050(52) начальным является базовый тест, представляющий собой набор наладочных тестов процессора, оперативной памяти, селекторных и мультиплексного каналов, накопителя на магнитной ленте, либо устройства ввода с перфокарт и пишущей машины. После предварительной наладки этими тестами дальнейшая наладка модели осуществляется с помощью проверочных программ — тест-секций, работающих под управлением тест-монитора. Тест-монитор загружается в оперативную память после успешного прохождения базового теста. 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, зацикливание отдельных тестов или примеров. При этом обеспечивается возможность гибкой оперативной связи с оператором либо через приказы, отдаваемые им с пишущей машинки, либо с пульта управления.

Тест-секция состоит из заглавия и тестовых примеров, количество которых может меняться. Тестовый пример представляет собой независимый программный блок, содержащий префикс и собственно проверочную часть. Связь тест-монитора с тестовыми примерами осуществляется через заглавия секций и префиксы примеров. Тест-монитор и тест-секции ЕС-1050(52) занимают оперативную память объемом не менее 20К байт.

В модели ЕС-1033 для проверки работы устройств используются аппаратные средства: блок контроля, распределенный по всему процессору и объединенный с большинством его блоков, блок диагностики и программные средства для центрального процессора, выполненные в виде микропрограмм анализа ошибок и диагностических микропрограмм. Достаточно большое число точек контроля и высокая эффективность процесса контроля и диагностики в этой модели не требуют предварительного прогона базовых тестов.

Различают следующие фазы работы системы обслуживания: рабочую; профилактическую и ремонтную.

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

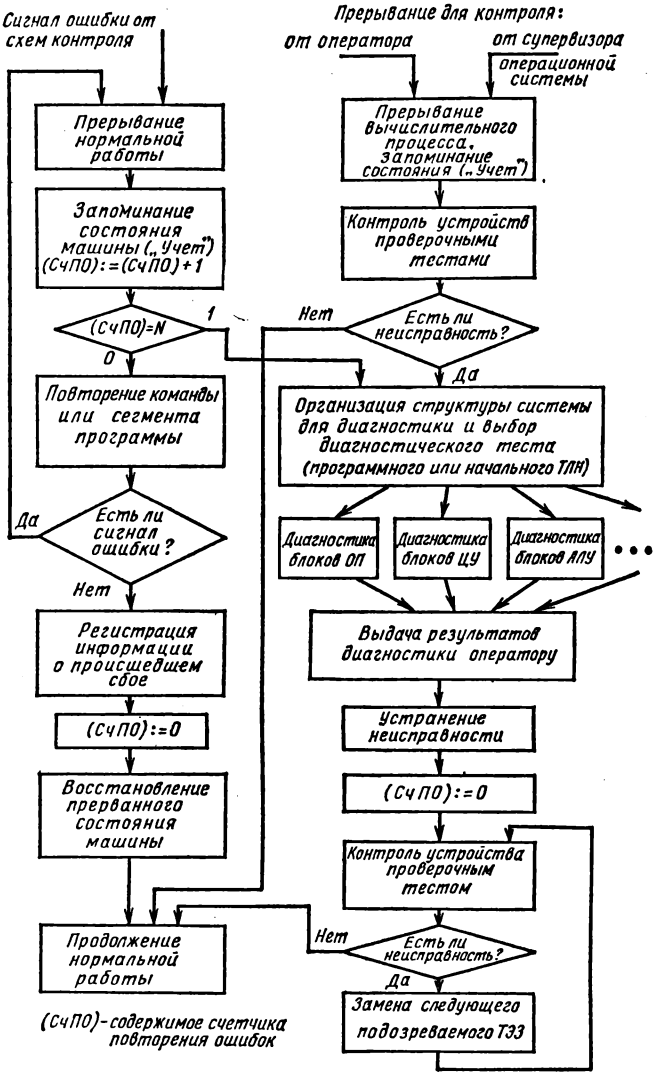


Рис. 16.6. Схема процесса автоматической проверки и диагностики машины

программы немедленно прекращается (рис. 16.6). 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(52) 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 отведенной для этого области оперативной памяти. В модели ЕС-1050(52) в ОП, начиная со 128-й ячейки, записывается информация объемом в 39 двоичных слов, из которых 34 содержат информацию основных регистров и управляющих триггеров, а пять – 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, 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. 16.6). 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 оказывается пройденным, например, начинается 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. В отличие от модели ЕС-1050(52) в модели ЕС-1033 все рассмотренные выше действия выполняются соответствующими микропрограммами.

Учет всей информации о состоянии машины в условиях, в которых происходили прерывания по сигналам ошибок, позволяет 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, индицированная 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. 16.6). В зависимости от развитости системы контроля сразу может диагностироваться конкретный блок устройства.

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 функциональных 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 определенный 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 и схемы контроля могут проверяться и диагностироваться, для чего с помощью УСКД принудительно можно нарушать условие четности кодов в различных схемах, где имеется аппаратный контроль. Кроме сигналов ошибок и команды диагностики, в режиме «Учет» использу-

ются также и сигналы 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 или микропрограмм и специальных диагностических процедур, реализуемых аппаратурой встроенного в процессор блока контроля и диагностики (БКД) или отдельного блока диагностики (БД). В модели ЕС-1050(52) диагностические программы в определенной последовательности вводятся с магнитной ленты в две специально выделяемые буферные области оперативной памяти. Вводом управляет специальная циклическая канальная программа. Она, будучи загруженной в ОП по программе начальной загрузки с ленты, содержащей тесты, все время находится там. Буферные области заполняются попеременно. Пока идет загрузка в одну из областей, блоком контроля и диагностики исполняется диагностическая программа, находящаяся в другой области. В модели ЕС-1033 диагностические микропрограммы всегда находятся в блоке постоянной памяти. Большинство программ выбирается для исполнения непосредственно по номеру возбужденного триггера РМО. Их выбором и исполнением управляет блок диагностики.

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(52) диагностические тесты дают возможность локализовать неисправность во всех устройствах с точностью до двух-трех 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 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.

Если при обнаружении ошибки в модели ЕС-1033 не пройден порог возможности повторения команды, то команда повторяется семикратно. Если при этом ошибка подтверждается или был пройден порог повторения, то начинается диагностирование с помощью ТЛН. Первый ТЛН выбирается в результате приоритетного анализа РМО.

ТЛН имеют бо́льшую разрешающую способность по сравнению с диагностическими тестами. Применение этих непрограммных тестов напоминает стендовую проверку приборов дискретного действия. Для различных блоков центрального управления и арифметико-логических 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).

Блоки процессора проверяются тестами локализации неисправностей при использовании блока контроля и диагностики или специального блока диагностики. Проверка 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(52) все ТЛН имеют фиксированный формат, содержащий 14 двойных слов (112 байт).

В модели ЕС-1050(52) ТЛН постоянно хранятся на магнитной ленте и режим ТЛН задается с пульта управления. Первый ТЛН определенного тестового «дерева» программой начальной загрузки через селекторный канал записывается в фиксированную буферную область оперативной памяти. Блок контроля выбирает из соответствующего поля ТЛН коды-стимулы и последовательно заполняет ими определенные регистры и триггеры проверяемого устройства. После этого он выбирает 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. В модели ЕС-1033 ТЛН состоят из последовательностей специальных диагностических микрокоманд, хранящихся в постоянной памяти. Тесты также содержат микрокоманды, с помощью которых осуществляется автоматическое переключение процессора в режим диагностики или возвращение в режим исполнения прерванной программы.

Для ЕС-1033 по результату сравнения устанавливается, какой следующий из двух возможных тестов должен быть вызван для исполнения постоянной памяти, т. е. устанавливается, какому тесту передать управление. Последний ТЛН в той или иной ветви диагностического дерева

указывает место неисправности. Информация об этом выдается оператору с помощью печатающей машинки.

Если последний тест не обнаружил ошибки, 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.

Глава 17

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, 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 применяется при наладке ЭВМ или во время профилактических работ.

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. Автоматически 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 $T_{зп}$ — 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, могут использоваться системы статических и динамических приоритетов. В первом случае приоритет программы является абсолютным, не связанным с динамикой процесса реализации пакета, во втором — 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, повторить счет из-за возникновения ошибок и т. д.) и процесса защиты данных, относящихся к прер-

ванным программам, от воздействия программ, выполняемых в данный момент.

Режим разделения времени. При работе в этом режиме 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 программным обеспечением, работающая 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 и 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 $T_{оа}$ — 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 $T_{оа}$, полученного при максимальном числе одновременно работающих абонентов, ко времени $T_{оа}$, 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 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. 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 быть представлен своей математической моделью. Тогда объединенная модель абонентов будет представлять собой сложную модель, состоящую из отдельных моделей абонентов, работающих с набором различных задач.

Применение режима разделения времени возможно и в том случае, когда ЭВМ выполняет функции вычислительного и управляющего органа некоторой замкнутой системы управления, т. е. когда работа ЭВМ протекает в реальном масштабе времени.

Работа ЭВМ в реальном масштабе времени характеризуется следующими особенностями:

машина должна быть в постоянной готовности принять данные от источников информации (эти данные управляют работой ЭВМ);

потоки входных данных носят случайный характер; машина должна работать как система массового обслуживания без потерь, так как потери поступающих данных не допускаются (как правило, они не могут быть переданы повторно);

время между моментами получения информации о состоянии управляе-

мого объекта или процесса и выдачи управляющего воздействия должно удовлетворять определенным требованиям.

В системах с разделением времени обычно предусматривается работа в режиме запрос-ответ, а также диалоговый режим работы.

Режим запрос-ответ (обработка запросов). Это вид телеобработки, при котором в соответствии с запросами от абонентских пунктов центральный вычислительный орган системы посылает на пункты данные, содержащиеся в одном или нескольких основных файлах данных. Обновление основных файлов производится в оперативном режиме в соответствии с данными, поступающими от абонентов. Такой вид телеобработки может применяться для управления различными процессами в реальном масштабе времени.

Разновидностью режима запрос-ответ является такой способ организации решения задач, при котором их программы постоянно хранятся в памяти системы, а запросы на их выполнение и необходимые исходные данные поступают от абонентских пунктов. Каждому запросу соответствует своя программа, которая формирует ответ на запрос. Число отвечающих программ ограничено емкостью памяти системы. Следовательно, ограничен и набор допустимых запросов, поэтому режим 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 диспетчеризации, определения порядка обслуживания запросов. Диспетчеризация определяет некоторую стратегию обслуживания, с помощью которой из множества поступивших запросов выбирается совокупность запросов для обслуживания в первую очередь. Правила выбора запросов из этой совокупности определяют дисциплину обслуживания.

Оценка качества функционирования системы в режиме запрос-ответ может осуществляться с помощью тех же показателей, которые были указаны для режима разделения времени.

Диалоговый режим организуется в форме прямого двустороннего взаимодействия системы и абонента в реальном масштабе времени. Он возможен и без режима разделения времени. Применяется диалоговый режим для обеспечения непосредственного и эффективного взаимодействия системы и человека в процессе решения конкретных задач, при отладке программ и компонентов программного обеспечения, при управлении функционированием системы (человек включается в контур управления функционированием системы, что приводит, как правило, к увеличению ее выходного эффекта).

Для осуществления диалогового режима необходимо удовлетворить следующие требования:

технические и программные средства, участвующие в реализации диалогового режима, должны обеспечивать необходимое быстродействие в обмене информацией между ЭВМ и абонентом, достаточное для работы в реальном масштабе времени;

абонентам должна быть предоставлена возможность подробного смыслового описания своих сообщений на достаточно высоком уровне формализации;

в составе аппаратуры абонентских пунктов должны быть средства информации, манипулирования и анализа для многомерных структур данных;

в системе должна быть предусмотрена возможность вычисления апостериорных и априорных оценок качества и эффективности ее функционирования.

В состав средств, обеспечивающих диалоговый режим, входят:

а. Технические средства: терминальная аппаратура для ввода и вывода информации, устройства отображения символьно-графической информации, средства исследования для многомерных структур данных, средства, обеспечивающие непосредственную связь абонента с ЭВМ системы. В ЭВМ должны быть заложены возможности работы в реальном масштабе времени и в мультипрограммных режимах, а также возможности по организации непосредственной многоканальной дистанционной связи.

б. Математические средства, включающие программное обеспечение диалогового режима, алгоритмическое обеспечение этого режима и языковое обеспечение.

Программное обеспечение диалогового режима должно быть совместимо с операционной системой ЭВМ; оно в сущности является расширением этой системы.

Алгоритмическое обеспечение диалогового режима представляет собой совокупность алгоритмов, реализующих используемые методы качественного и количественного анализа информационных сообщений и всего процесса функционирования ЭВМ в этом режиме. Сюда входят алгоритмы грамматического и семантического анализа сообщений, алгоритмы графического представления информации, распознавания образов, статистического анализа, алгоритмы оценки эффективности процессов при работе ЭВМ в диалоговом режиме и др.

Языковое обеспечение диалогового режима – это совокупность функционально-ориентированных языковых средств для записи (описания) входных сообщений и представления выводимой информации. Входной язык диалогового режима должен быть доступным, гибким и универсальным. Ему должны быть присущи однозначность и лаконичность выражения смыслового содержания.

в. Информационное обеспечение – совокупность информационных массивов, необходимых при работе ЭВМ в диалоговом режиме: различные тексты и данные для абонентов, служебная информация для ведения грамматического и семантического анализа и т. д.

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 обще-

ние 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.

17.2. Дисциплины обслуживания требований

В машинах Единой системы обслуживание требований, поступающих от абонентов, может осуществляться либо с учетом их приоритетов, либо без учета. Приоритет требования (заявки, запроса) — это его характеристика, определяющая место требования в очереди на обслуживание. Он назначается либо в соответствии с характером задачи, решаемой по этому требованию, либо в соответствии с той ролью, которую играет в обслуживающей системе источник требования (абонент). В связи с этим может оказаться, что два требования на решение одной и той же задачи относятся к различным уровням приоритета, если они исходят от разных абонентов. В то же время и требования на решение различных задач, поступающие от одного и того же абонента, могут иметь различный приоритет в зависимости от характера этих задач.

Дисциплина обслуживания — это правила, согласно которым требования выбираются из очереди для обслуживания, т. е. это порядок рассасывания очереди. В литературе, кроме термина «дисциплина обслуживания», имеет хождение и его синоним — «алгоритм диспетчеризации». В дальнейшем будут использоваться оба эти термина. Вопрос о выборе дисциплины обслуживания возникает в тех случаях, когда требования не идентичны, они различаются по времени на обслуживание, по допустимому времени ожидания обслуживания, по размерам штрафа за каждую единицу времени пребывания в очереди и т. д.

При выборе необходимого алгоритма диспетчеризации следует удовлетворить ряд требований, некоторые из которых являются противоречивыми:

способность обслуживать требования высшего приоритета в кратчайшее время;

способность обслуживать требования низшего приоритета в приемлемые для абонентов сроки (во всяком случае в такие сроки, которые не дали бы повода абонентам отказаться от услуг обслуживающей системы);

стремление полнее загрузить ЭВМ полезной работой, т. е. выполнением программ абонентов (от того, какой принят алгоритм диспетчеризации, зависит частота переключения ЭВМ от выполнения одной программы к другой, а значит и суммарная потеря времени на эти переключения);

стремление уменьшить среднее время реакции ЭВМ на запрос и среднее число требований, ожидающих обслуживания;

обеспечение относительной простоты реализации выбранного алгоритма диспетчеризации.

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 ЭВМ на запрос и среднего числа запросов, ожидающих обслуживания, с одной стороны, и требование учета степени важности и срочности обслуживания отдельных запросов, с другой стороны. В связи с этим при разработке алгоритма диспетчеризации возникает задача нахождения компромиссного решения, в той или иной степени удовлетворяющего указанные требования. Разработка оптимального алгоритма диспетчеризации является задачей исследования операций, требующей 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 обслуживания.

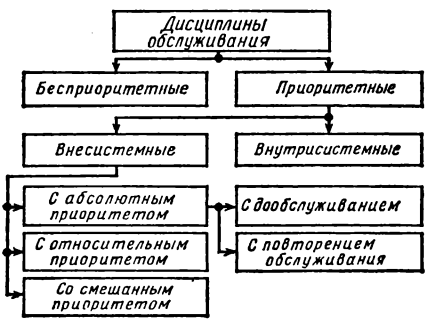


Рис. 17.1. Классификационная схема дисциплин обслуживания требований

Типы дисциплин обслуживания требований. Все дисциплины обслуживания требований можно разбить на две группы (рис. 17.1):

а) дисциплины обслуживания без учета приоритетов (бесприоритетное обслуживание, или бесприоритетная диспетчеризация);

б) дисциплины обслуживания с учетом приоритетов.

В первой группе объединены простейшие дисциплины, обычно не учитывающие ценности поступающих на обслуживание требований и их временных характеристик. Для требований не устанавливается никакой дифференциации по степени их важности и срочности.

К числу дисциплин первой группы относятся:

круговое циклическое обслуживание, или обслуживание требований в порядке их поступления;

обслуживание в инверсном порядке по принципу «последним пришел — первым обслужен»;

обслуживание со случайным и равновероятным выбором требований из числа ожидающих в очереди.

Круговое циклическое обслуживание представляет собой наиболее простую и довольно широко распространенную дисциплину обслуживания. При круговом обслуживании предполагается, что для всех абонентов стоимость ожидания результатов выполнения их программ по заявкам одинакова. Все поступающие в машину заявки формируют одну общую очередь. В случае работы ЭВМ в режиме разделения времени (с квантованием времени, необходимого на выполнение программ по запросам абонентов) выполняемые программы по очереди получают одинаковый квант времени $\tau_{кв}$. Если за время одного кванта выполнение программы по заявке i-го абонента не заканчивается, производится ее прерывание и переход к выполнению программы очередного абонента. При этом программа i-го абонента устанавливается в конец очереди. Формирование очереди осуществляется в порядке поступления заявок, т. е. по принципу «первым пришел — первым обслужен». Если в процессе выполнения программ абонентов длина кванта времени не изменяется, говорят, что обслуживание заявок осуществляется в соответствии с алгоритмом круговой циклической диспетчеризации с постоянным квантом времени. Если же длина кванта изменяется, например, в зависимости от длины очереди заявок на обслуживание, имеет место алгоритм круговой циклической диспетчеризации с переменной длиной кванта времени.

В случае, когда обслуживающая система работает в режиме запрос-ответ или в режиме разделения времени, но без его квантования для выполнения программ абонентов, алгоритм круговой циклической диспетчеризации оказывается проще. Кроме того, меньше затрачивается времени на переключение программ.

В отличие от кругового циклического обслуживания, при обслуживании в инверсном порядке учитывается ценность заявки. Наиболее ценной считается та заявка, которая поступила позже других, находящихся в очереди и ожидающих обслуживания. Ценность этой заявки определяется тем, что в ней содержится самая «свежая» информация о состоянии некоторого объекта или процесса.

При обслуживании в инверсном порядке можно рекомендовать алгоритмы диспетчеризации с выбыванием из очереди устаревших заявок. Очередь имеет ограничения по длине, т. е. по числу заявок, которые могут в ней находиться в ожидании обслуживания. Если с приходом новой, i-й заявки очередь переполняется, ее покидает та из заявок, которая поступила в систему раньше других, находящихся в очереди. Следовательно, i-я заявка замещает в очереди наиболее устаревшую заявку, после чего отправляется на обслуживание. При таком алгоритме диспетчеризации суммарная ценность заявок, находящихся в очереди, увеличивается, и убытки из-за задержки обслуживания заявок и потери заявок уменьшаются. Средняя длина очереди не изменяется, но уменьшается среднее время ожидания обслуживания заявок, поскольку покидает очередь и теряется часть наиболее долгоожидавших и поэтому наиболее обесцененных заявок.

В любой дисциплине обслуживания с учетом приоритетов (приоритетной дисциплине) должны быть заложены правила, согласно которым принимаются два решения:

а) выбрать требование из числа ожидающих в очереди (одной или нескольких) для обслуживания в момент готовности ЭВМ;

б) прервать или продолжить обслуживание (поскольку оно начато) требования, находящегося на обслуживании, до его завершения или до окончания интервала времени, выделенного этому требованию.

В зависимости от того, как принимается первое из указанных решений, приоритетные дисциплины могут быть внесистемными или внутрисистемными. При использовании внесистемной приоритетной дисциплины решение о выборе следующего требования для обслуживания принимается внесистемно, оно зависит лишь от номера приоритета, соответствующего классу, к которому принадлежит требование. Если в системе обслуживается q различных классов требований, то каждому классу приписывается свой приоритетный признак (номер) i, причем $1 \leqslant i \leqslant q$.

Классу требований с высшим приоритетом присваивается признак 1, а классу требований с низшим приоритетом – признак q. Чем выше уровень приоритета требования, тем меньше его приоритетный номер. Каждому уровню приоритета соответствует своя очередь, упорядоченная в зависимости от поступления требований этого уровня. Требование, которое должно обслуживаться следующим, выбирается из очереди требований наивысшего приоритета.

Решение о выборе следующего требования для обслуживания при использовании внутрисистемной дисциплины принимается внутрисистемно, т. е. оно полностью или частично базируется на заключениях, касающихся текущего состояния ЭВМ, например времени ожидания требований в текущий момент.

В зависимости от того, как решается вопрос с прерыванием обслуживания требования j-го приоритета, если в систему поступило требование i-го приоритета ($i < j$), все внесистемные приоритетные дисциплины разделяются на три группы:

а) дисциплины обслуживания с относительным приоритетом;

б) дисциплины обслуживания с абсолютным приоритетом;

в) дисциплины обслуживания со смешанным приоритетом.

При использовании алгоритма диспетчеризации с относительным приоритетом с приходом требования i-го приоритета обслуживание требования j-го приоритета не прерывается. Оно продолжается до его полного завершения, если время, необходимое для обслуживания требования j-го приоритета, не квантуется, или до истечения кванта времени, выделенного этому требованию, в случае квантования времени на обслуживание требований. Следовательно, при таком алгоритме диспетчеризации возможна ситуация, когда в данный момент обслуживается требование не самого высокого приоритета из числа имеющихся в очереди. Алгоритм диспетчеризации с относительным приоритетом иначе называется алгоритмом диспетчеризации без прерывания обслуживания.

Для алгоритма диспетчеризации с абсолютным приоритетом, иначе называемого алгоритмом диспетчеризации с прерыванием обслуживания, характерно немедленное прерывание обслуживания требования j-го приори-

тета, как только в системе появляется требование i-го приоритета ($i < j$), которое и начинает обслуживаться. При обслуживании с прерыванием в любой момент времени осуществляется обслуживание требования с высшим приоритетом из числа имеющихся в очереди.

Дисциплина обслуживания с абсолютным приоритетом имеет следующие модификации:

б) абсолютный приоритет с дообслуживанием; при такой дисциплине возобновление выполнения программы по требованию j-го приоритета начинается с точки прерывания;

б) абсолютный приоритет с повторением обслуживания; в этом случае выполнение прерванной программы начинается не с точки прерывания, а либо с той точки, которая соответствует началу прерванного кванта времени, если время на выполнение программ по заявкам абонентов квантуется, либо с самого начала программы, если это время не квантуется.

В алгоритме диспетчеризации со смешанным приоритетом заложены возможности обслуживания требований в соответствии либо с абсолютным приоритетом, либо с относительным приоритетом в зависимости от длительности времени непрерывного обслуживания требования j-го приоритета. Обслуживание требования j-го приоритета немедленно прерывается с приходом требования более высокого, i-го приоритета, если оно непрерывно обслуживалось в течение времени, меньшего $\tau_{\text{п}}$ (точка переключения), и продолжается до полного завершения или до окончания выделенного кванта времени, если оно обслуживалось в течение времени, большего или равного $\tau_{\text{п}}$.

Внесистемные приоритетные дисциплины обслуживания требований. Рассмотрим ряд дисциплин обслуживания требований, которые в соответствии с приведенной ранее классификационной схемой относятся к числу внесистемных приоритетных дисциплин.

1. Приоритетное обслуживание по принципу «чем короче программа, тем выше ее приоритет». Очередность обслуживания устанавливается в порядке возрастания величины τ_{ok}, представляющей собой чистое время, затрачиваемое на обслуживание k-го требования, без учета потерь времени на ожидание обслуживания в очереди и на прерывания в выполнении соответствующей программы (одной или нескольких) по этому требованию. В первую очередь обслуживается самая короткая программа, т. е. требование, на обслуживание которого затрачивается наименьший объем вычислений. Требования с одинаковыми значениями τ_{ok} обслуживаются в порядке поступления. Значения τ_{ok} должны быть известны априори. Приоритетное обслуживание по указанному принципу применяется в случае, когда стоимость ожидания обслуживания для всех требований одинакова.

2. Приоритетное обслуживание по так называемому правилу τ/C в порядке возрастания отношения τ_{ok}/C_k, где через C_k обозначена величина, определяющая стоимость ожидания обслуживания k-го требования, отнесенная к единице времени, а через τ_{ok} — как и в предыдущем случае, чистое время обслуживания этого требования. Как показано в работе [9], обслуживание в соответствии с такой дисциплиной приводит к минимизации функции суммарного штрафа за ожидание обслуживания.

3. Организация очередей в соответствии с алгоритмом Корбато [15],

который предназначался для применения в экспериментальной вычислительной системе, построенной на базе машины IBM-7094.

Согласно алгоритму Корбато программы абонентов, ожидающих обслуживания, распределяются по очередям, число которых равно N. Сначала программа, состоящая из $n_{\text{п}}$ слов, поступает в очередь с номером i, причем

$$i = \left[\log_2\left(\left[\frac{n_{\text{п}}}{n_{\text{кв}}}\right] + 1\right)\right],$$

где $n_{\text{кв}}$ – количество слов, передаваемых из оперативной памяти в ВЗУ и обратно в течение одного кванта времени $\tau_{\text{кв}}$ при выполнении программы, а символ [ ] обозначает целую часть числа.

Будучи в очереди i, программа получает квант обслуживания, равный $2^i\tau_{\text{кв}}$. Если этого кванта окажется недостаточно, программа для дообслуживания отправляется в конец очереди с номером $i+1$.

Обслуживание программы из очереди i прерывается, и она ставится в начало этой очереди, если во время ее обслуживания появляется программа в очереди с меньшим номером. Этой последней программе предоставляется квант обслуживания.

Номер последней очереди N определяется из условия

$$N = \left[\log_2\left(\left[\frac{n_{\text{п max}}}{k_{\text{о}} n_{\text{кв}}}\right] + 1\right)\right],$$

где $n_{\text{п max}}$ – максимально допустимый размер программы; $k_{\text{о}}$ – доля времени, расходуемого на обмен информацией между ВЗУ и оперативной памятью в течение кванта времени $\tau_{\text{кв}}$.

В алгоритме Корбато реализован такой же порядок обслуживания очередей и перехода программ из одной очереди в другую, как и в алгоритме FB_N, описанном в работе [15]. Программы из очереди N получают квант обслуживания, равный $2^N\tau_{\text{кв}}$. Если его окажется недостаточно, программа поступает на дообслуживание в конец этой же очереди.

4. Обслуживание в порядке, определяемом ролью абонентов в системе. Приоритет требования зависит не от характера решаемой задачи по этому требованию, а от того, откуда оно поступило. Приоритеты абонентам присваиваются волевым решением должностных лиц.

Кроме рассмотренных, возможны и другие внесистемные приоритетные дисциплины обслуживания требований или различные модификации этих дисциплин. Одной из таких модификаций является алгоритм диспетчеризации с двумя приоритетными очередями № 1 и № 2. Вновь поступающие требования ставятся в конец очереди № 1. Обслуживание требований из этой очереди осуществляется по принципу «первым пришел – первым обслужен». Каждому требованию предоставляется один квант времени. Если одного кванта оказывается достаточно, требование покидает очередь, в противном случае оно отправляется на дообслуживание в конец очереди № 2. Требования из очереди № 2 обслуживаются только при пустой очереди № 1. Каждому из них также предоставляется один квант времени. Недообслуженные требования из очереди № 2 устанавливаются в конец этой же очереди.

Необходимо отметить, что каждый из рассмотренных алгоритмов

диспетчеризации может быть реализован как алгоритм с абсолютным приоритетом, с относительным приоритетом или со смешанным приоритетом.

Внутрисистемные приоритетные дисциплины обслуживания требований. Внесистемные приоритетные дисциплины обслуживания, хотя и являются в гораздо большей степени изученными и распространенными по сравнению с внутрисистемными, во многих практических ситуациях могут оказаться неприемлемыми, поскольку при их использовании выбор очередного требования на обслуживание определяется только уровнем приоритета и не принимаются во внимание состояние обслуживающей системы и другие факторы. В системе определенного назначения может быть поставлено условие не задерживать требование любого класса свыше некоторого, заранее установленного предела. Следовательно, при выборе очередного требования на обслуживание необходимо учитывать время пребывания в системе каждого требования, находящегося в очереди. Это приводит к использованию внутрисистемных дисциплин обслуживания. К числу таких дисциплин относятся дисциплина обслуживания с динамическим приоритетом, дисциплина обслуживания с чередующимся приоритетом и др.

В алгоритме диспетчеризации с динамическим приоритетом предусматривается такой порядок выбора требований на обслуживание, который определяется не только их приоритетной классификацией, но и временем ожидания каждого требования, стоящего в очереди, или допустимым временем пребывания в системе. Согласно этому алгоритму на обслуживание выбирают требование, у которого оставшееся допустимое время пребывания в системе минимально. Приоритет каждого требования растет как некоторая функция времени ожидания, т. е. чем дольше требование находится в очереди, тем больше шансов у него получить обслуживание.

Обозначим $I_j(t)$ мгновенный приоритетный индекс (номер), который в момент времени t получает требование j-го класса, обслуживаемое вместе с другими требованиями в соответствии с динамическим приоритетом. Для внесистемных алгоритмов диспетчеризации этот индекс является величиной постоянной, для внутрисистемных он изменяется во времени.

Пусть u_j – допустимое время пребывания требования j-го класса в системе, которое отражает порядок срочности и, следовательно, начальный уровень приоритета этого требования, а $W_j(t)$ – время, проведенное в системе к моменту t требованием j-го класса. Тогда в случае динамического приоритета

$$I_j(t) = u_j - W_j(t).$$

По мере возрастания $W_j(t)$ приоритетный индекс $I_j(t)$ убывает, поэтому для любого требования j-го класса наступит момент, когда его индекс станет меньше, чем у всех других требований, находящихся в очереди. При одновременном поступлении двух требований предпочтение отдается требованию с меньшим приоритетным индексом, т. е. с меньшим u_j. Порядок обслуживания требований с одинаковым приоритетным индексом в момент t может устанавливаться либо в соответствии с круговым циклическим алгоритмом, либо на основе случайной равновероятной выборки.

Таким образом, при использовании алгоритма диспетчеризации с

динамическим приоритетом выбор очередного требования на обслуживание начинается с просмотра всех требований в очереди и выявления требования с наименьшим приоритетным индексом.

Легко заметить, что для алгоритма диспетчеризации с динамическим приоритетом предельными случаями являются алгоритм с относительным приоритетом и алгоритм круговой циклической диспетчеризации. Действительно, если d представляет собой разность $u_i - u_j$, где u_i – допустимое время пребывания в системе требования i-го класса, то при $d \to \infty$ динамический приоритет вырождается в относительный приоритет, так как значение приоритетного индекса будет полностью определяться начальным уровнем приоритета (влияние времени ожидания пренебрежимо мало), а при $d \to 0$ он соответствует круговому циклическому обслуживанию, так как разница в значениях мгновенных приоритетных индексов требований i-го и j-го классов определяется только временем пребывания их в системе.

Дисциплина обслуживания с динамическим приоритетом принципиально возможна в сочетании с прерыванием обслуживания требования с бо́льшим приоритетным индексом, если в очередь поступило требование с меньшим индексом, или без прерывания обслуживания такого требования.

Другим примером внутрисистемной дисциплины обслуживания является обслуживание с чередующимся приоритетом [15]. В соответствии с этой дисциплиной после завершения обслуживания требования i-го класса ($1 \leqslant i \leqslant q$) следующим на обслуживание принимается также требование i-го класса, если оно имеется в очереди. В противном случае из очереди выбирают требование с наименьшим приоритетным индексом. Например, при $q = 2$ требования 1-го класса получают преимущество в обслуживании над требованиями 2-го класса, если первым в систему поступило требование 1-го класса. После обслуживания всех требований 1-го класса, находящихся в очереди, преимущество в обслуживании получают требования 2-го класса. Таким образом, происходит чередование в приоритете обслуживания требований 1-го и 2-го классов, вследствие чего дисциплина обслуживания и называется дисциплиной с чередующимся приоритетом. Эта дисциплина является внутрисистемной, так как решение о выборе очередного требования на обслуживание определяется не только приоритетным индексом этого требования, но также и классом последнего обслуженного требования.

Сравнение внесистемных и внутрисистемных дисциплин обслуживания показывает [15], что первые из них обладают одним преимуществом. Оно заключается в том, что при использовании внесистемных дисциплин для классов требований с более высоким приоритетом может устанавливаться статистическое равновесие даже при отсутствии установившегося режима работы системы. Иначе говоря, такие требования образуют очереди конечной длины, и времена их пребывания в системе конечны. В то же время, требования с низким приоритетом образуют бесконечные очереди и времена их пребывания в системе бесконечны. Недостаток внесистемных дисциплин заключается в отсутствии таких управляемых параметров, с помощью которых можно было бы регулировать соответствующие характеристики каждого класса требований, например длину очереди, среднее время ожидания обслуживания, среднее время пребывания

в системе и др. Исключением является дисциплина обслуживания со смешанным приоритетом, при которой для обслуживания требований q классов имеется $q-1$ управляемых параметров. Большинство внутрисистемных дисциплин имеют управляемые параметры. Так, в алгоритме диспетчеризации с динамическим приоритетом таким параметром является $u_j (1 \leqslant j \leqslant q)$, где q — число различных классов требований, обслуживаемых системой. В случае использования внутрисистемных дисциплин не существует частичного равновесия: если бесконечно растет очередь из требований какого-либо класса, то бесконечно растут и очереди из требований всех других классов. В этом недостаток внутрисистемных дисциплин, которые, тем не менее, в практическом отношении являются более перспективными с точки зрения возможности регулирования поведения ЭВМ с целью изменения характеристик обслуживаемых требований.

Выполнение в ЭВМ программ абонентов по частям с предоставлением каждой из них определенного количества квантов времени связано с заметным снижением производительности процессора за счет частых передач информации между ОЗУ 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 передач информации между ОЗУ и ВЗУ состоит в том, что при приоритетном обслуживании требований каждому уровню приоритета ставится в соответствие своя величина $\tau_{кв}$, причем с увеличением номера уровня эта величина 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 бóльшим номером. Следовательно, требования с бóльшим временем обслуживания получают бóльшие значения $\tau_{кв}$. 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, то это требование перемещается в конец очереди, принадлежащей следующему уровню приоритета с меньшим номером.

Можно добиться уменьшения времени обслуживания за счет сокращения простоев процессора при обмене информацией между ступенями памяти ЭВМ путем регулировки величины $\tau_{кп}$. При уменьшении числа требований $\tau_{кв}$ увеличивается, следовательно, уменьшается число непроизводительных передач информации и сокращается время на обслуживание требований.

Показатели эффективности дисциплин обслуживания требований. В вычислительных системах, построенных на базе ЕС ЭВМ, принципиально возможны любые алгоритмы диспетчеризации. Однако целесообразность использования того или иного алгоритма будет определяться целым рядом факторов, в частности назначением системы, характером решаемых задач, допустимостью потерь требований на обслуживание, числом

уровней приоритетов, интенсивностью входящих потоков требований, степенью загрузки системы, режимом работы системы и т. д.

Выбор целесообразной дисциплины обслуживания иногда вполне очевиден и однозначен. Например, при выборе системы в режиме запрос-ответ для обслуживания идентичных (по важности и продолжительности обслуживания) требований очевидной является простейшая дисциплина — бесприоритетное круговое циклическое обслуживание. Влияние загрузки системы на выбор дисциплины обслуживания требований характеризуется тем, что при большом уровне загрузки эффективность функционирования системы существенно зависит от дисциплин обслуживания. При малых загрузках влияние дисциплин обслуживания на эффективность системы резко уменьшается.

Выбор оптимальной дисциплины связан с необходимостью оценки эффективности различных дисциплин. Оценка проводится с помощью показателей эффективности. Есть и другие задачи анализа дисциплин обслуживания, решаемые с участием этих показателей. К их числу относятся такие задачи, как оценка относительного выигрыша от применения различных вариантов приоритетного обслуживания по сравнению с бесприоритетными дисциплинами, если они допустимы, оценка относительной сложности программной и технической реализации алгоритмов диспетчеризации. Анализ алгоритмов диспетчеризации предполагает решение вопросов определения вероятности потери заявок различных приоритетов в системах с потерями, определение распределения времени ожидания или времени обслуживания и его численных характеристик (среднего значения, дисперсии, коэффициента вариации), объединение некоторых потоков заявок в один поток и др.

Оценка эффективности алгоритмов диспетчеризации обычно производится с помощью нескольких показателей, каждый из которых характеризует какую-то одну сторону процесса обслуживания требований. Применение не одного, а нескольких показателей эффективности тем более необходимо, что изучение этого процесса может производиться с различных позиций: с позиции разработчика, для которого наиболее важно знание распределения длин очередей, с позиции пользователей, которых прежде всего интересует распределение времени ожидания ответа, с позиции самой системы, когда важным оказывается распределение времени занятости системы обслуживанием требований.

Поскольку эффективность алгоритмов диспетчеризации прежде всего определяется возможностью потери заявок или задержки начала их обслуживания, в ряде работ [25] для оценки алгоритмов предлагается использовать функцию полного (суммарного) штрафа. Задержка заявки учитывается либо по средней величине времени задержки (при этом предполагается, что заявка обесценивается пропорционально времени задержки), либо по величине допустимого времени ожидания, когда заявка обесценивается скачком при превышении этого времени (в пределах допустимого времени ожидания ценность заявки не снижается). В первом случае функция полного штрафа имеет вид

$$F_1 = \sum_{i=1}^{m} C_i \lambda_i T_i, \qquad (17.1)$$

а во втором случае

$$F_2 = \sum_{i=1}^{m} b_i \lambda_i P(> T_{\text{д}i}), \tag{17.2}$$

где C_i – штраф за единицу времени ожидания обслуживания требования i-го класса; λ_i – интенсивность потока требований i-го класса; T_i – среднее время ожидания обслуживания требований i-го класса; b_i – штраф за потерю одного требования i-го класса; $P(> T_{\text{д}i})$ – вероятность того, что требование i-го класса будет ожидать обслуживания свыше допустимого времени $T_{\text{д}i}$; m – число классов требований.

Выражения (17.1) и (17.2) справедливы, если потери для требований различных классов аддитивны, а для выражения (17.1) эти потери, кроме того, линейно зависят от времени ожидания. В ряде случаев стоимость ожидания нелинейна. Однако рассмотрение этих случаев связано с необходимостью вычисления высших моментов распределения времени ожидания, что резко усложняет вычисление функции штрафа. При определении функции штрафа F_1 в выражении (17.1) может фигурировать не среднее время ожидания обслуживания, а среднее время пребывания в системе требований i-го класса.

С помощью функций штрафа F_1 и F_2 можно сравнивать различные алгоритмы диспетчеризации, так как для различных алгоритмов будут различны величины T_i и $P(> T_{\text{д}i})$. Это сравнение можно производить и по значениям показателей эффективности функционирования системы, которые, естественно, критичны к дисциплинам обслуживания требований.

В качестве частных показателей эффективности алгоритмов диспетчеризации могут быть использованы:

T_i – среднее время ожидания обслуживания требований i-го класса;

H_i – среднее время пребывания в системе требований i-го класса;

L_i – средняя длина очереди требований i-го класса;

D_T, D_H, D_L – дисперсии соответственно времени ожидания обслуживания, времени пребывания в системе и длины очереди требований i-го класса;

σ_T, σ_H, σ_L – среднеквадратические отклонения этих же величин;

K_T, K_H, K_L – коэффициенты вариации этих величин.

Все эти частные показатели определяются аналитически, если известны функции распределения соответствующих величин.

Целесообразность организации той или иной дисциплины обслуживания может определяться с помощью относительных показателей [15]:

а) относительной длительности ожидания обслуживания требований i-го класса

$$\alpha_{T_i} = \frac{T_{i\text{и}}}{T_i};$$

б) относительной длительности пребывания в системе требований i-го класса

$$\alpha_{H_i} = \frac{H_{i\text{и}}}{H_i};$$

в) относительной длины очереди требований i-го класса

$$\alpha_{L_i} = \frac{L_{iп}}{L_i},$$

где $T_{iп}$, $H_{iп}$, $L_{iп}$ – соответственно те же величины, что и T_i, H_i, L_i, но определенные в условиях обслуживания только одной, изолированной очереди требований i-го класса.

Относительные показатели изменяются на отрезке $[0, 1]$. Чем больше их значения, тем меньше влияние требований других классов на качество обслуживания требований i-го класса. Например, для требований высшего приоритета значения этих показателей приближаются к единице или равны единице, если используется дисциплина обслуживания с абсолютным приоритетом. Следовательно, наличие в системе требований с более низким приоритетом не оказывает или почти не оказывает влияния на качество обслуживания требований высшего приоритета. Малые значения относительных показателей отражают тот факт, что время ожидания, время пребывания в системе, длина очереди значительно возрастают, если требования данного класса обслуживаются совместно с требованиями других классов.

17.3. Организация функционирования ЭВМ 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 обработки, подготовка рабочих программ к пакетной реализации, планирование загрузки устройств системы. Это характерно как для обычной, так и для дистанционной пакетной обработки.

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 могут использоваться системы статических и 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 $T_{\text{знп}}$, затрачиваемое на выполнение заданного набора рабочих программ. Это время зависит не только от качества 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 $T_{\text{знп}}$. Оперативное планирование в сущности представляет собой диспетчеризацию, одно из основных требований к которой состоит в том, чтобы время на составление плана (время на диспетчеризацию) 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.

Методы мультипрограммного планирования разделяются:

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, применяемые 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 приближенных решений не обеспечивают получения минимального значения $T_{ЗНП}$, но по своей трудоемкости 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		
*F*2	Пакетирование		Одиночная программа
*F*1	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 оператором. Выполнение программ в режиме пакетной обработки осуществляется с помощью управляющей программы ДОС, причем одновременно могут быть выполнены 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 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 программами 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 быть использовано 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 быть использованы 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 осуществляется динамическое распределение пакетированных заданий между разделами. 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 заданий, 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. Максимальное число одновременно выполняемых задач и подзадач не должно превышать 255. В это число входят как системные задачи и подзадачи, так и задачи и подзадачи абонентов.

В однопроцессорной ВС в каждый данный момент времени 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, монопольно используя 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.

17.4.

Организация функционирования ЭВМ Единой системы в режиме разделения времени

В СРВ 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 обменом 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 режиме 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 обра-

зуемых 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.

Операционная система ОС ЕС в настоящее время обеспечивает простейший вариант режима разделения времени, называемый квантованием времени [37]. Задачи, выполняемые в этом режиме, объединяются в группу квантования, которая размещается в оперативной памяти. Каждая из таких задач получает управление на определенный интервал времени, называемый квантом. По истечении этого времени управление получает следующая задача из группы квантования и т. д. Величина кванта и признак принадлежности задачи к группе квантования устанавливаются при генерации ОС. Одновременно могут выполняться несколько групп квантования, различающихся по уровню приоритета. Задачи, принадлежащие одной группе квантования, имеют один и тот же уровень приоритета. Приоритет квантующихся задач может быть как выше, так и ниже приоритета неквантующихся задач. Последние могут находиться в оперативной памяти вместе с квантующимися задачами.

Квантующиеся задачи находятся в оперативной памяти до полного выполнения. Они могут выполняться как в режиме мультипрограммирования с фиксированным числом задач, так и в режиме мультипрограммирования с переменным числом задач.

На базе старших моделей ЕС ЭВМ, имеющих оперативную память емкостью не менее 512К байт, можно организовать систему разделения времени для одновременного обслуживания более ста удаленных абонентов. При этом, кроме работы в обычном режиме разделения времени, возможно выполнение таких действий, как работа с наборами данных (ввод с АП, запоминание, поиск, модификация, редактирование, вывод и т. д.), разработка или выполнение программ в режиме диалога. Выполнение этих действий производится с помощью набора команд, подаваемых пользователями с абонентских пунктов.

Программные средства телеобработки ЕС ЭВМ представляют совокупность макрокоманд, ориентированных на выполнение следующих функций:

установление контакта и управление передачей сообщений между центральными вычислительными средствами (ЦВС) системы телеобработки и АП;

анализ и исправление различных ошибок и ошибочных ситуаций, возникающих во время передачи сообщений, регистрация этих ошибок;

диагностика неисправностей в оборудовании и каналах передачи данных;

получение, использование и освобождение буферной памяти по мере необходимости во время работы.

Программные средства телеобработки ЕС ЭВМ являются расширением ОС ЕС. Они представляют собой совокупность программ, включающих базисный телекоммуникационный метод доступа (БТМД) и общий телекоммуникационный метод доступа (ОТМД).

Методы доступа БТМД и ОТМД – это программные средства, обслуживающие наборы данных с телекоммуникационной организацией. Они представляются на уровне макрокоманд языка АССЕМБЛЕР. Метод доступа БТМД включает средства в основном для организации приема и передачи данных.

Метод ОТМД является методом доступа с очередями. Программные наборы данных, получаемые от АП или передаваемые АП и используемые программами абонента, рассматриваются как очереди сообщений. Для управления сообщениями метод доступа представляет гибкий язык высокого уровня.

Метод доступа ОТМД обеспечивает сопряжение с прикладными программами, позволяющее абоненту обрабатывать сообщения, поступающие по телекоммуникационной сети, такими же простыми макрокомандами, как и при обработке для местных устройств ввода-вывода. Использование в системах телеобработки метода доступа ОТМД позволяет не только эффективно управлять потоком сообщений, поступающих от АП, и программ, генерирующих сообщения, но и параллельно производить любую требуемую их обработку. Наличие этих средств наряду с дополнительно включенным сервисом способствует построению высокопроизводительных и удобных для абонента систем обработки данных на расстоянии.

Программные средства телеобработки, реализуемые в виде указанных телекоммуникационных методов доступа, работают под управлением ОС ЕС. Чтобы позволить операционной системе сконцентрировать усилия на обработке данных, ей необходимо сообщить информацию о выбранном методе доступа и его конкретных ресурсах (количестве и размерах буферов ввода-вывода, способе управления ими и т. д.) и характеристики каналов передачи данных и устройств, составляющих систему телеобработки. Используя эту информацию, операционная система устанавливает программное сопряжение между программой пользователя и программами ввода-вывода телекоммуникационных методов доступа. Для подбора программных средств, соответствующих используемому составу технических средств и решаемым задачам, обеспечивается возможность генерации в составе ОС ЕС средств программного обеспечения обработки данных на расстоянии.

Глава 18

ПОСТРОЕНИЕ МНОГОПРОЦЕССОРНЫХ И МНОГОМАШИННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА БАЗЕ ЕС ЭВМ

18.1. Общие принципы построения многопроцессорных и многомашинных вычислительных систем

Многопроцессорная (многомашинная) вычислительная система представляет собой совокупность взаимосвязанных и согласованно действующих однотипных или неоднотипных процессоров (машин) и других устройств, а также программных средств (программного 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 в целом системы, 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, превалирующим в зависимости от назначения, характера решаемых задач и состояния системы.

Современные ВС строят 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 главные 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 рабочих программ.

Классификация ВС. Укажем прежде всего на различия между многопроцессорными и многомашинными вычислительными системами. Многопроцессорная система строится либо на базе одной многопроцессорной машины, либо на базе нескольких однотипных или неоднотипных процессоров. Многомашинная ВС строится на базе нескольких машин, каждая из которых может быть как однопроцессорной, так и многопроцессорной. В связи с этим многомашинные 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.

По назначению вычислительные системы 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 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 системы 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 строятся совмещенного типа.

По постоянству структуры системы разделяются на ВС с постоянной и ВС с переменной структурой. Под структурой ВС понимается состав системы и схема 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 раздельными.

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 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 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 строят 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 программном 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 подсоединения к одному или к нескольким процессорам ВС.

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 с помощью двухканального переключателя;

на уровне устройств управления 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 многовходовых 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 программному 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 ЭВМ 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 — сверхоперативная память небольшого объема (регистровая память);

основная оперативная память — быстродействующая память модульной структуры с независимым обращением к модулям;

большая оперативная память — сравнительно недорогая память модульной структуры на ферритовых сердечниках со средним быстродействием и с независимым обращением к модулям;

внешняя память — ВЗУ большого объема на магнитных дисках (реже барабанах) с достаточно малым временем выборки информации;

архивная память — ВЗУ практически неограниченного объема на магнитных лентах, магнитных картах.

При такой иерархии памяти может быть следующее размещение информации по ее уровням;

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: 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 многопроцессорным ВС. При построении и организации функционирования многопроцессорных систем необходимо удовлетворить 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 только занятость процессоров, их возможности по производительности.

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: 1) 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; 2) каждый процессор должен иметь возможность начать операцию ввода-вывода через любой из каналов; 3) в каждом процессоре должна быть заложена возможность реакции на всевозможные прерывания; прерывания, появляющиеся внутри данного процессора, обычно обрабатываются этим же процессором; 4) 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 два и более процессоров одновременно могли выполнить одну и ту 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 оперативной памяти ВС несколько процессоров одновременно не должны изменять одну и ту же величину. Для этого предусматривается временная блокировка обращений процессоров к избранным ячейкам памяти или блокировка входов нескольких процессоров 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 программным 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 та же, что 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 программного обеспечения. В вычислительных системах предусматривается реализация всех видов телеобработки, указанных в гл. 1.

Выбор того или иного способа организации вычислительного процесса ВС (режима работы) зависит от степени однородности центральных вычислительных средств системы, способа закрепления функций за ЭВМ ЕС, 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 быть выполнено заблаговременно. С бо́льшими 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 уровней 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 раз-

работке программного 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 [35].

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 выбирают такие методы, которые не требуют сколько-нибудь значительных затрат машинного времени системы на составление плана ее работы, хотя при этом план не является оптимальным в смысле минимума времени на реализацию заданного набора задач.

18.2. Структуры многомашинных и многопроцессорных ВС

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 подобных режимов сопряжена с необходимостью разветвленной сети внешних абонентских пунктов, аппаратуры передачи данных, мультиплексоров передачи данных, линий связи большой протяженности;

способ объединения машин (процессоров) в системе или способ обмена информацией между машинами (уровень комплексирования 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. 18.1). В таких системах программы решаемых задач исходная информация и промежуточные данные хранятся в общей системной памяти (ООП и ВЗУ), которая доступна всем ЭВМ ЕС.

Обмен информацией между машинами осуществляется через ООП и ВЗУ. Вычислительные системы с общей памятью могут быть как централизованными, так и децентрализованными. Обычно они являются системами совмещенного типа.

На рис. 18.2 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.

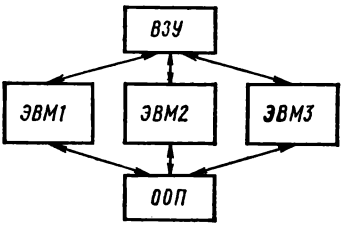


Рис. 18.1. Вычислительная система с общей памятью

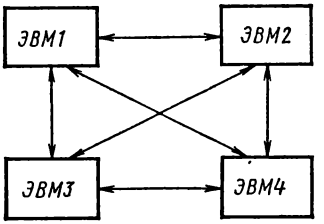
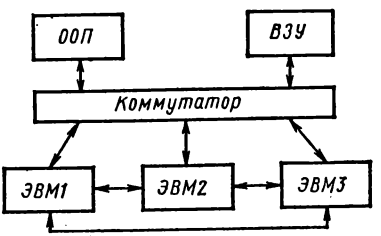


Рис. 18.2. Децентрализованная ВС

Более совершенная структура системы приведена на рис. 18.3. 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 ООП с помощью коммутатора. По степени централизации управления такие системы могут быть централизованными, со смешанным управлением.

Рис. 18.3. Вычислительная система с общей памятью и непосредственной связью между машинами



В децентрализованной ВС с иерархической структурой средств управления (рис. 18.4) можно выделить несколько уровней управления, в данном случае два. 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. 18.3. 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. УВС обычно строятся

из универсальных ЭВМ. Однако в них зачастую просматривается целевая направленность системы, обусловленная выполняемыми ею функциями, используемыми устройствами ввода-вывода и отображения информации, особенностями организации взаимодействия с коллективами людей в человеко-машинных комплексах и т. д.

Имеются и полностью специализированные вычислительные системы (СВС), в которых не только общая организация вычислительных процессов, но и реализация структур, являются прямым следствием особенностей решаемых задач. Такие СВС еще называют проблемно-ориентированными, так как они предназначаются для решения задач определенного класса или ориентированы на решение определенных проблем. При реализации СВС в ряде случаев достигается упрощение конструкций основных модулей СВС и связей между ними.

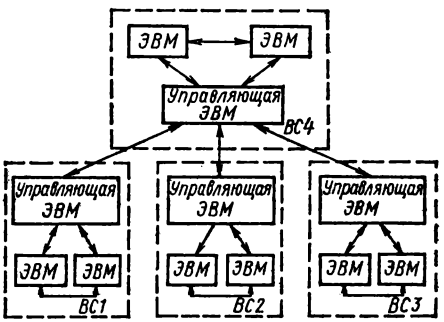


Рис. 18.4. Децентрализованная ВС с иерархической структурой средств управления.

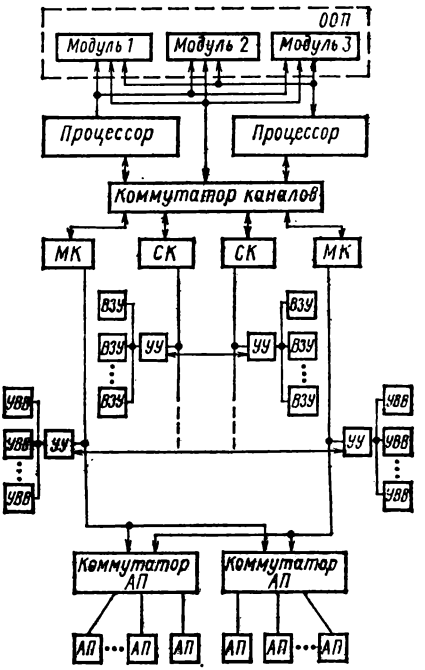


Рис. 18.5. Двухпроцессорная СРВ

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 непосредственное взаимодействие удаленных абонентов с ВС. Ниже

приводятся варианты структур многопроцессорных систем с разделением времени, в которых такое взаимодействие имеет место.

Структура многопроцессорной СРВ определяется рядом факторов, наиболее существенными из которых 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

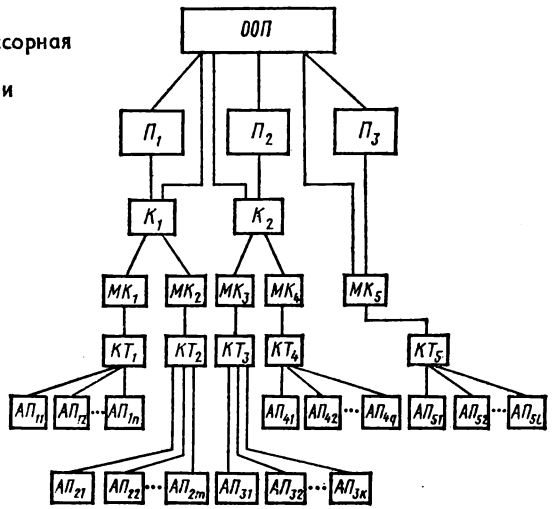


Рис. 18.6. Многопроцессорная СРВ с симплексными связями

связей и включению дополнительных элементов, обеспечивающих повышение надежности функционирования системы.

Структура многопроцессорной СРВ зависит от организации симплексной или мультиплексной связи между абонентскими пунктами и процессорами системы. Такие связи характерны для многопроцессорных СРВ с иерархической структурой.

Один из возможных вариантов структуры двухпроцессорной СРВ приведен на рис. 18.5. В системе имеется общее поле оперативной памяти. Кроме того, устройства управления ВЗУ и УВВ связаны между собой с помощью двойных переключателей. Абонентские пункты связаны с мультиплексными каналами посредством коммутаторов абонентских пунктов, функции которых могут выполнять устройства управления АП.

Иерархические структуры многопроцессорных СРВ изображены на рис. 18.6 и 18.7, где $П_1$, $П_2$... — процессоры; $К_1$, $К_2$ — коммутаторы каналов; $МК_1$, $МК_2$... — мультиплексные каналы; $КТ_1$, $КТ_2$... — коммутаторы терминалов (абонентских пунктов). Система с симплексными связями (рис. 18.6) характеризуется тем, что в ней связь между элементами из различных уровней (например, между абонентскими пунктами

$АП_{11}$, $АП_{12}$, ..., $АП_{1n}$, с одной стороны, и $МК_1$, с другой, 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.

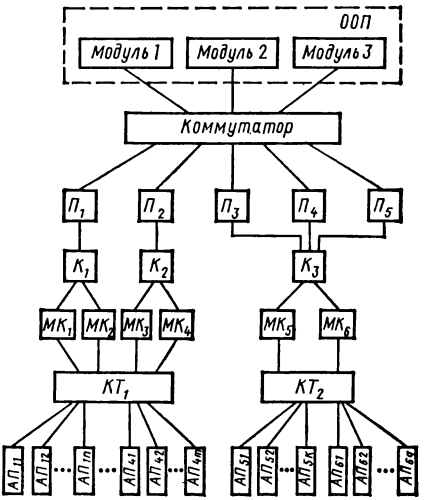


Рис. 18.7. Многопроцессорная СРВ с мультиплексными связями

В системе с мультиплексными связями (рис. 18.7) 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.

18.3. Многопроцессорная система с общим полем оперативной памяти

Организация общего поля оперативной памяти: Общее поле оперативной памяти для двух и большего числа 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 процессоров или более 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. 18.8).

Вариант 1 (рис. 18.8, *а*): очереди запросов на обращение к оперативной памяти организуются на входе каждого блока памяти. Каждый очередной запрос анализируется процессором и направляется в магазин запросов соответствующего блока памяти. Удовлетворение запросов производится в порядке их приоритетов.

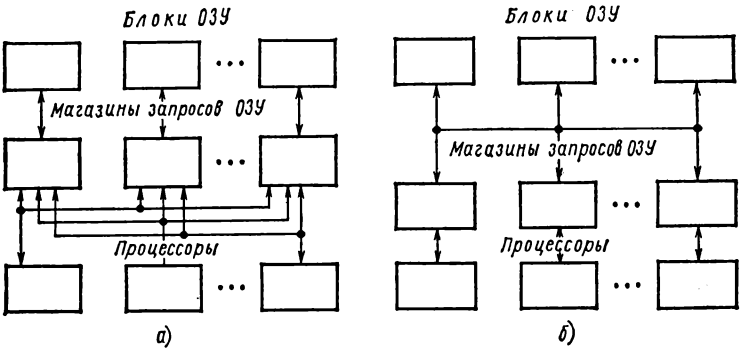


Рис. 18.8. Схема организации обращения к общей оперативной памяти в многопроцессорной системе: *а* — очереди запросов на обращение организуются на входе каждого блока памяти; *б* — очереди запросов организуются на выходе каждого процессора

Вариант 2 (рис. 18.8, *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 времени в вычислительной системе, как правило, решаются задачи одного уровня приоритетов или близких уровней 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 в 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.

Структура оперативной памяти двухпроцессорной системы. Рассмотрим возможный вариант структуры основной оперативной памяти двухпроцессорной системы, построенной на базе средних или старших моделей ЕС ЭВМ.

Основная оперативная память 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 памяти сгруппированы по четыре блока в группе (рис. 18.9), причем каждая группа связана со своей шиной обращения. За-

просы к каждой группе блоков ООП от модулей УУП производятся практически одновременно. Запросы же внутри группы чередуются со сдвигом во времени. Право на обращение получает тот модуль УУП, который содержит наиболее срочный запрос в данный блок памяти (уровень позиционного приоритета модулей УУП, приведенных на рис. 18.9, 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

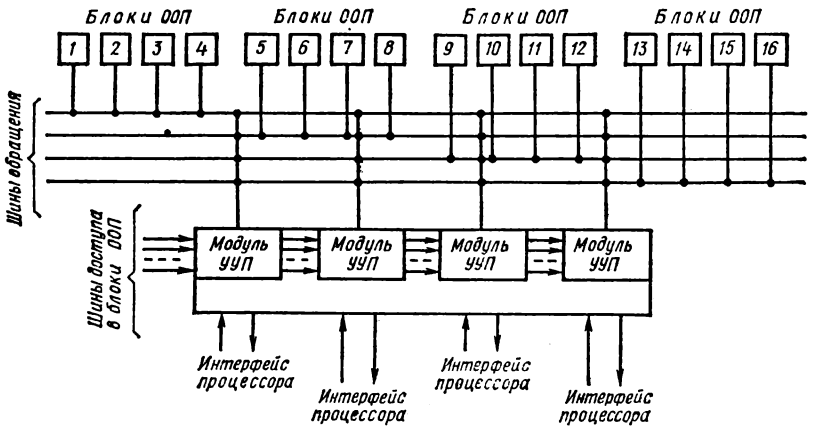


Рис. 18.9. Структура основной оперативной памяти многопроцессорной системы

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 обращения 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.

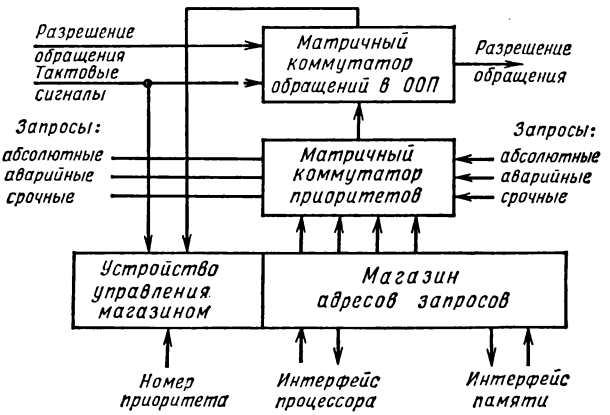


Рис. 18.10. Схема модуля устройства управления памятью

Схема одного модуля УУП приведена на рис. 18.10. Она включает магазин адресов запросов со своим устройством управления, матричный коммутатор приоритетных шин и матричный коммутатор обращений.

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 выбирают 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 влево, к модулю УУП более высокого позиционного приоритета.

Матричный коммутатор обращений обеспечивает выполнение следующих операций:

а) сравнение разрешаемых запросов данного магазина с сигналами

разрешения обращения, поступающими слева, от модулей УУП более высокого позиционного приоритета; наличие запроса и сигнала разрешения означает возможность обращения к блокам памяти для данного магазина;

б) выдачу сигнала возможности обращения в магазин запросов, если имеет место ситуация, описанная выше, и, кроме того, появится сигнал окончания цикла обращения;

в) объединение сигналов обращения данного магазина с сигналами разрешения, поступившими слева, и передачу сигналов разрешения вправо, 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.

Режимы работы двухпроцессорной системы на базе моделей ЕС ЭВМ. Для такой системы предусматриваются 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 многопроцессорный 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 для этого процессора, 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 выдать 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 операторной памяти прямоадресуемая область памяти для одного или 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 от 0-го до 4095-го.

Организация функционирования ВС в режимах «Мультисистема» 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), сигнала 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 многовходовыми, 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.

18.4. Операционные системы многопроцессорных ВС

Для эффективного функционирования многопроцессорной ВС принципиально необходимо развитое программное обеспечение. Особенно это касается операционной системы, которая должна обеспечивать эффектив-

ное использование всех ресурсов системы (в том числе машинного времени процессоров) во всех режимах работы, в частности в режиме мультипрограммирования, когда в каждом процессоре ВС осуществляется выполнение нескольких программ пользователей в режиме совпадения.

Операционная система многопроцессорной ВС (иначе – многопроцессорная операционная система – МОС) во многом сходна с ОС мультипрограммной однопроцессорной ВС, однако, как правило, является более сложной по структуре основных компонентов и выполняемым ими функциям. Это прежде всего относится к таким функциям, как обеспечение эффективного использования ресурсов системы, синхронизация работы процессоров, обнаружение и исправление ошибок без прекращения работы системы и др. С другой стороны, наличие дополнительных, избыточных функциональных блоков в МОС часто позволяет найти лучшие (по сравнению с тем, что дает возможность сделать ОС однопроцессорной системы) решения целого ряда проблем: адаптируемости к изменению параметров входящих потоков требований, обеспечения заданного времени на обслуживание требований, расширения возможностей ОС без изменения имеющихся общесистемных программ и т. д.

Имеющиеся в настоящее время многопроцессорные операционные системы по своей глобальной организации и способу функционирования могут быть разделены на три типа [33]:

системы централизованного типа, иначе называемые МОС типа «ведущий – ведомый»;

системы децентрализованного типа, или системы с раздельным выполнением заданий в каждом процессоре;

симметричные МОС, или системы с однородной обработкой во всех процессорах.

Система централизованного типа обязательно имеет в своем составе один ведущий процессор и один или несколько ведомых процессоров. Ведущий процессор реализует супервизорные функции, т. е. выполняет программы МОС. Он может иметь либо специализированную конфигурацию, приспособленную исключительно лишь для выполнения супервизорных функций (супервизорный процессор), либо конфигурацию процессора универсального типа. В последнем случае ведущий процессор может быть использован и для расчетов по прикладным программам пользователей. В рассматриваемой системе супервизорные программы не обязательно должны обладать повторной входимостью, так как они выполняются только одним процессором. Доступ к системным таблицам имеет только ведущий процессор, поэтому в системе не возникают конфликтные ситуации или блокировки, связанные с таким доступом (разумеется, что для прикладных программ при обращении к системным таблицам конфликты возможны).

Эффективность использования системы централизованного типа может быть достаточно высокой только при условии, что ведущий процессор выполняет свои супервизорные функции достаточно быстро, своевременно загружая процессоры новыми задачами. В противном случае время простоя, накапливаемое в ведомых процессорах, делает такой способ функционирования системы пригодным лишь для специальных применений, где эти простои допускаются. Он целесообразен и эффективен

также для неоднородных систем с существенно различающимися по своей производительности процессорами. При прочих равных условиях эффективность систем централизованного типа тем ниже, чем больше преобладание коротких задач в потоке заданий, так как такие задачи предъявляют более высокие требования к супервизорным функциям ведущего процессора.

Системы централизованного типа отличаются сравнительно простым аппаратным и программным обеспечением, малой гибкостью и недостаточной надежностью, так как отказы и сбои в ведущем процессоре могут привести к полной потере работоспособности системы.

Системы с раздельным выполнением заданий в каждом процессоре отличаются определенной децентрализацией в выполнении общесистемных супервизорных функций. Для них нередки ситуации, когда каждый процессор должен работать автономно и выполнять относящиеся к нему супервизорные, исполнительные и вспомогательные функции подобно тому, как это имеет место в однопроцессорной системе. Характерной особенностью такого типа организации МОС является то, что каждый процессор выполняет супервизорные функции только в соответствии с его собственными потребностями, которые определяются требованиями программ, предназначенных для этого процессора. Оперативная память мультипроцессорной системы используется в режиме разделения времени, поэтому нет необходимости для каждого процессора иметь полный набор копий модулей операционной системы. Супервизорные программы, поскольку они выполняются несколькими процессорами, должны обладать повторной входимостью или же их копии должны загружаться в каждый процессор (может использоваться и комбинация этих двух процедур). Для каждого процессора имеется свой собственный набор системных таблиц, вследствие чего число конфликтов, связанных с блокировкой этих таблиц, сводится к минимуму. Устройства ввода-вывода, файлы и т. д. распределяются между процессорами, так что каждый процессор имеет свой набор этих устройств. Реконфигурация системы ввода-вывода (перераспределение устройств ввода-вывода между процессорами) обычно требует ручного переключения.

Рассматриваемая система обладает большей надежностью по сравнению с предыдущей, так как одиночные отказы не приводят к полной потере работоспособности системы. Однако повторный запуск отказавшего процессора и восстановление программ, которые им выполнялись, обычно связаны с большими трудностями. Высокая эффективность функционирования системы достигается при условии равномерной загрузки всех процессоров.

Наиболее совершенной, но зато и наиболее сложной в реализации, является система с симметричной, или однородной обработкой во всех процессорах. Такая система состоит из однотипных процессоров, которые по своим функциональным возможностям абсолютно взаимозаменяемы. Функции ведущего, по необходимости, могут передаваться с одного процессора на другой. Каждый процессор может в полном объеме выполнять общесистемные супервизорные функции. Однако в ходе реализации потоков заданий каждый процессор выполняет только те супервизорные функции, которые неразрывно связаны с решаемой им задачей,

а также функции, необходимые для очередной задачи в случае, когда текущая задача будет прервана или полностью завершена. Любая задача может быть распределена на любой процессор, а для выполнения громоздких задач могут использоваться различные наборы процессоров. Естественно, что при указанной организации вычислительного процесса необходимо общесистемное управление, реализующее такие функции, как планирование мультипрограммной работы системы, управление системными таблицами, разрешение конфликтных ситуаций и др. Процессор, обеспечивающий выполнение общесистемных функций, называется исполнительным. В каждый момент времени исполнительным может быть только один процессор.

В симметричной системе такие средства, как память, каналы устройства ввода-вывода, модули ОС, системные таблицы, наборы данных используются совместно всеми процессорами. Поэтому возможны конфликтные ситуации, когда к одному и тому же средству одновременно обращаются несколько процессоров. Конфликтные ситуации при обслуживании запросов разрешаются с помощью приоритетов. Каждому процессору устанавливается определенный приоритет не только с целью разрешения конфликтов при запросах общесистемных ресурсов, но и для ранжирования управляющих функций. Модули ОС должны обладать повторной входимостью, тогда несколько процессоров могут одновременно выполнять одни и те же сервисные программы. Здесь, как и в случае систем с раздельным выполнением заданий, допускается и другое решение: наличие необходимого числа копий модулей ОС для выполнения супервизорных процедур.

Симметричная система обладает наибольшей эффективностью по сравнению с другими типами МОС. Это достигается большей возможностью оптимального распределения нагрузки между процессорами, а также тем, что такие неспецифические функции, как прерывание для ввода-вывода, могут выполняться наименее загруженными в данный момент блоками. Надежность симметричной системы выше других МОС, в ней реализован единственный способ эффективного резервирования. При выходе из строя отдельных устройств система частично сохраняет работоспособность.

Ранее отмечалось сходство программного обеспечения многопроцессорных ВС и мультипрограммных однопроцессорных систем. Это сходство особенно заметно, если иметь в виду общие цели, проблемы и трудности реализации программного обеспечения и необходимость оставаться в рамках общих положений и определений. Можно сказать, что сходство между программным обеспечением этих двух систем (особенно в части ОС) является поверхностным. Различия же между ними определяются не только разной степенью сложности. Они значительно глубже. Основное различие обусловлено тем, что в многопроцессорной системе несколько программ выполняются одновременно и параллельно, в то время как в мультипрограммной однопроцессорной системе они выполняются в режиме совпадения. Следовательно, в многопроцессорной системе при неудачных планировании и диспетчеризации снижение эффективности использования общесистемных ресурсов будет заметнее. Кроме того, в такой системе ошибка в программном обеспечении или временный отказ одного из процессоров приводят, как правило, к более тяжелым

последствиям, так как они быстро распространяются по всей системе и создают сложные ситуации, разобраться в которых и внести исправления оказывается весьма трудно.

Конфигурация аппаратных средств многопроцессорных вычислительных систем накладывает заметный отпечаток на перечень и характер функций, выполняемых МОС. Одна из главных функций — это распределение и управление ресурсами. Основные задачи, которые решаются при выполнении этой функции, те же, что и в случае однопроцессорной системы. Это — распределение памяти и управление ею, планирование работы системы, диспетчеризация.

Проблемы распределения памяти и управления ею, возникающие при работе многопроцессорной и однопроцессорной систем, аналогичны. Решение этих проблем также одинаково. Оно основывается на разбиении памяти на участки, сегменты, страницы. Здесь одним из наиболее важных вопросов является техническое и особенно программное обеспечение разбиения памяти на страницы и управления доступом к командам или данным, которые могут использоваться несколькими процессорами. В зависимости от типа МОС для каждого процессора может потребоваться собственная память, физически недоступная для других процессоров.

В многопроцессорных системах при организации доступа к постраничной памяти ключевым вопросом является вопрос о том, где осуществляется преобразование адресов — в памяти или в процессоре. Это относится главным образом к логическим операциям по преобразованию адресов, хотя физическое расположение предназначенных для этого аппаратных средств также имеет значение. Различают три варианта расположения регистров для преобразования логических адресов в физические: в оперативной памяти, в процессорных блоках и в специальном устройстве преобразования адресов памяти с большой скоростью выполнения логических операций. Эти варианты различаются не столько сложностью аппаратной реализации, сколько временем на доступ к необходимой физической странице памяти.

При решении задач планирования и диспетчеризации мультипроцессор следует рассматривать как многоканальную систему массового обслуживания с очередями (мультипрограммная однопроцессорная система является более простой одноканальной обслуживающей системой). Планировщик и диспетчер многопроцессорной системы должны быть более совершенными по сравнению с такими же компонентами однопроцессорной системы. Перед составлением плана реализации заданного набора заданий и задач осуществляется сегментация соответствующих проблемных программ с целью определения перечня задач и их частей, которые могут выполняться параллельно. Это необходимо для более полного использования возможностей многопроцессорной системы. СУПЕРВИЗОР МОС управляет реализацией составленного плана работы системы или алгоритмов диспетчеризации. Управление осуществляется как сравнительно простыми операциями, связанными с решением независимых задач, так и более сложными операциями по координации выполнения комплекса взаимосвязанных задач. Следует отметить, что планирование и диспетчеризация в многопроцессорной системе являются наиболее «слабо оптимизируемыми» процедурами. Они требуют наибольших затрат времени

на работу МОС и оказывают наибольшее влияние на эффективность использования процессоров.

В процессе функционирования системы обеспечивается постоянная взаимосвязь между процессорами, которая может быть как запланированной (например, при выполнении несколькими процессорами комплекса взаимосвязанных задач), так и незапланированной, возникающей как реакция на сигнал о возникновении ошибки в одном из процессоров или при обслуживании прерывания для ввода-вывода. Способность процессоров к осуществлению взаимной связи существенно влияет на выполнение функции балансировки загрузки процессоров, которая является ключевым фактором для достижения максимальной производительности многопроцессорной системы. Важное значение имеет также обеспечение балансировки загрузки ввода-вывода. В случае, если каналы и аппаратура ввода-вывода образуют однородный комплекс, целесообразно использовать простой приоритетный принцип для формирования запросов на ввод-вывод, подобно тому как это делается в однопроцессорной системе. Труднее решается вопрос балансировки, когда некоторые тракты или оборудование ввода-вывода жестко закреплены за каким-то одним процессором.

Выполнение функции защиты системных таблиц и наборов данных от неправомерного доступа в многопроцессорной системе требует использования очень сложных процедур. Для обеспечения уверенной защиты обычно необходимо предусматривать некоторые аппаратные средства. В противном случае может возникнуть ситуация, когда неправомерный доступ к системной таблице или набору данных происходит в пределах цикла работы памяти, следующего непосредственно за правомерным доступом, и соответствующие программы не успевают заблокировать таблицу.

Одним из требований, которые предъявляются к ОС многопроцессорной системы, является способность автоматически распознавать изменения в оборудовании, готовом в данный момент к использованию, и вносить необходимые коррекции во все системные таблицы и программы. Эта способность наиболее просто реализуется в системах с симметричной обработкой во всех процессорах.

18.5. Показатели качества и эффективности функционирования вычислительных систем

Принадлежность вычислительной системы к тому или иному типу в соответствии с указанными выше классификационными признаками во многом определяет ее свойства.

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 работы системы.

Показатель качества — это некоторая численная характеристика, являющаяся функционалом от процесса работы системы и определяющая одну из сторон качества этого процесса. Показатели качества ВС можно разбить на две группы. В первую группу входят вероятностные показатели, во вторую — статистические оценки 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 работоспособность ВС оценивается с помощью показателей ее эффективности (W_i). Эффективность отражает степень соответствия ВС своему назначению, ее техническое совершенство и экономическую целесообразность. Количественно эффективность системы оценивается через показатели эффективности, являющиеся функциями множества параметров входящих потоков требований (или параметров реализуемых алгоритмов) $L_п$, множества параметров системы и ее элементов $L_с$ и множества параметров, характеризующих условия функционирования ВС, влияние внешних факторов на процесс функционирования $L_у$. Показатель эффективности определяется процессом функционирования системы; он является функционалом от этого процесса. В качестве показателей эффективности принимаются те показатели качества ВС, которые в наибольшей мере отражают степень соответствия системы своему назначению.

Основные требования, предъявляемые к показателям эффективности ВС: их следует выбирать исходя из интересов тех задач, которые решаются системой, т. е. показатели эффективности действительно должны количественно отражать степень соответствия системы своему назначению; должны быть критичны к параметрам множеств $L_п$, $L_с$, $L_у$; по возможности быть простыми, имеющими четкий физический смысл; эффективными в статистическом смысле (должны иметь сравнительно небольшую дисперсию). Если ВС является подсистемой более сложной системы, то ее показатель эффективности следует рассматривать 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 $L_п$, $L_с$, $L_у$, 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 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 за основной показатель ее эффективности, является вероятность выполнения поставленных перед ней задач на требуемом уровне за определенное время τ и при заданных условиях функционирования $P_{в.з}(\tau)$. На величину этого показателя влияют параметры всех множеств $L_п$, $L_с$, $L_у$. Однако вычисление значений $P_{в.з}(\tau)$ 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 $P_{в.з}(\tau)$ 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 — отношение среднего числа обслуженных требований за некоторый интервал времени к среднему числу поступивших требований за тот же интервал; коэффициент задержки в обслуживании требований — отношение среднего времени пребывания в системе к среднему времени, затрачиваемому на обслуживание требований (на решение задачи, соответствующей требованию).

Важное значение имеют и другие показатели, которые могут быть использованы 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 $T_и$ к разности $T_и - T_{п.в}$, где $T_{п.в}$ — 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 оцениваться 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 представлением чисел. Кроме того, при последовательной информационной связи между машинами системы выходные ошибки предыдущей машины становятся входными ошибками последующей.

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: 1) $P_c(\tau)$ — вероятность безотказной работы системы за время τ при заданных условиях эксплуатации; 2) T_c — наработка на отказ системы; 3) $T_{в.с}$ — 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 [6]:

а) коэффициент снижения эффективности системы

$$K_W = \frac{W}{W_0},$$

показывающий, какую часть эффективность реальной системы (W) составляет от эффективности идеальной (в смысле надежности) системы (W_0);

б) значение разности

$$\Delta W_0 = W_0 - W,$$

показывающее, 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_W достаточно велико или разность ΔW_0 достаточно мала, отказы элементов системы слабо влияют на эффективность, и поэтому могут оказаться нецелесообразными меры повышения надеж-

ности элементов ВС, поскольку они не оправдывают произведенных затрат. При малом значении K_W или недостаточно 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 заданными характеристиками.

Количественная оценка помехозащищенности ВС может осуществляться с помощью следующих показателей:

а) коэффициента помехозащищенности системы

$$K_{\text{п.з}} = \frac{W_{\text{п}}}{W_0},$$

показывающего, какую часть значение показателя эффективности системы, функционирующей в условиях действия помех с заданными характеристиками ($W_{\text{п}}$), составляет от значения показателя эффективности системы, функционирующей в идеальных условиях (W_0), когда помехи отсутствуют;

б) абсолютной величины разности

$$\Delta W_{\text{п.з}} = |W_0 - W_{\text{п}}|,$$

показывающей, насколько изменяется эффективность ВС под влиянием помех с заданными 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 ($\tau_{\text{г}}$), не превышающее допустимое τ,

$$P_{\text{г}}(\tau) = P\{\tau_{\text{г}} < \tau\};$$

б) среднее время подготовки системы к работе.

Экономичность системы — это затраты на ее разработку, создание и эксплуатацию. Количественно экономичность ВС может оцениваться с помощью показателя экономической эффективности $C_{\text{э}}$ — стоимости одной вычислительной операции, определяемой с учетом затрат на разработку, создание и эксплуатацию системы.

Глава 19

КОМПЛЕКСНОЕ ЦЕНТРАЛИЗОВАННОЕ ОБСЛУЖИВАНИЕ СРЕДСТВ ЕС ЭВМ

19.1. Функции и принципы построения системы комплексного централизованного обслуживания средств ЕС ЭВМ

Комплексное централизованное обслуживание (КЦО) средств ЕС ЭВМ представляет собой совокупность организационных, научных и технических мероприятий и работ по обслуживанию технических средств ЕС ЭВМ, фондированию, поставке и сопровождению программных средств, подготовке специалистов, занимающихся эксплуатацией этих средств. Эти мероприятия и работы осуществляются централизованно и направлены на повышение эффективности использования ЭВМ.

Система комплексного централизованного обслуживания (СКЦО) средств ЕС ЭВМ включает производственные, научные, учебные предприятия и подразделения, располагающие необходимой материально-технической базой, а также управляющие органы, объединенные между собой и с пользователями информационно-управляющими связями и выполняющие функции КЦО.

Основные функции СКЦО:

ввод в эксплуатацию и обслуживание технических средств ЕС ЭВМ в гарантийный и послегарантийный период, т. е. проведение пусконаладочных, ремонтно-восстановительных и профилактических работ;

фондирование базового и прикладного программного обеспечения ЕС ЭВМ, поставка его пользователям, ввод в эксплуатацию и сопровождение;

проектирование размещения технических средств ЕС ЭВМ, а также типового внутреннего оформления помещений (вычислительных центров), предназначенных для установки этих средств;

выпуск (тиражирование) систематически совершенствуемых операционных систем, пакетов прикладных программ общего назначения, а также технической и учебно-методической документации по ЕС ЭВМ;

подготовка и переподготовка для работы со средствами ЕС ЭВМ эксплуатационного персонала предприятий, учреждений и организаций министерств и ведомств, специалистов-инструкторов социалистических стран — участниц сотрудничества в области электронной вычислительной техники и, в необходимых случаях, специалистов других стран, в которые экспортируются средства ЕС ЭВМ;

взаимодействие по вопросам КЦО средств ЕС ЭВМ с соответствую-

щими национальными организациями технического обслуживания (НОТО) социалистических стран – участниц сотрудничества в области ЭВТ.

При создании и организации функционирования системы КЦО необходимо соблюсти следующие основные принципы, многие из которых являются общими для автоматизированных систем организационного управления [10].

1. Ориентация на перспективные, прогрессивные методы и средства эксплуатации и обслуживания электронной вычислительной техники. В СКЦО должны быть заложены возможности перехода от комплексного централизованного обслуживания отдельных автономно используемых ЭВМ и ВЦ к обслуживанию вычислительных центров коллективного пользования (ВЦКП), сетей ЭВМ, а также ЭВМ и вычислительных комплексов, являющихся управляющими и вычислительными органами различных АСУ.

2. Модульность структуры технических и программных средств системы на всех ее уровнях. Модульный принцип построения СКЦО необходим не только для облегчения решения вопросов по удовлетворению запросов пользователей, но и для более полного и оперативного обеспечения непрерывности развития системы, поэтапного наращивания ее возможностей путем добавления новых технических средств и программных компонентов.

3. Комплексный (системный) подход к проектированию СКЦО. Такой подход предполагает проведение проектирования на основе системного анализа, включающего целый комплекс вопросов технического, экономического и организационного характера, решение которых необходимо для того, чтобы создаваемая система наилучшим образом удовлетворяла предъявляемым требованиям. Системный анализ – совокупность методов и приемов для изучения сложных объектов, представляющих собой сочетание взаимосвязанных более простых объектов и изменяющихся во времени процессов. Такие сложные объекты можно назвать обобщенными динамическими системами. К их числу принадлежит и СКЦО. Принцип системного подхода к проектированию СКЦО проявляется, кроме всего прочего, в том, что разработка технических и программных средств системы должна проводиться в тесном взаимодействии, даже если эта работа выполняется разными коллективами. Это объясняется взаимозависимостью параметров аппаратной и программной частей системы.

4. Типизация и унификация технических и программных средств системы. Суть этого принципа заключается в том, что при разработке и создании таких дорогостоящих и трудоемких систем, как СКЦО, необходимо, чтобы предлагаемые разработчиками решения подходили возможно более широкому кругу пользователей. Типизация и унификация решений при разработке технических и программных средств СКЦО способствует концентрации квалифицированных сил разработчиков, а также ускорению разработки, поскольку максимально используются решения, полученные ранее в ходе проектирования других систем. Типизация проектных решений, если она проведена правильно, не только не препятствует, а наоборот, способствует увеличению возможностей учета индивидуальных особенностей различных подсистем СКЦО, связанных с конкретными условиями их применения.

5. Иерархия структуры звеньев управления процессом функционирования системы. Такая иерархия является естественным следствием иерархи-

ческой организации структуры всей системы комплексного централизованного обслуживания средств ЕС ЭВМ.

6. Согласованность пропускных способностей основных функциональных звеньев системы. Это прежде всего относится к определению оптимального числа производственных бригад, оснащенных необходимыми средствами и непосредственно занимающихся пусконаладочными работами и техническим обслуживанием ЕС ЭВМ.

7. Обеспечение максимальных удобств пользователей в общении с системой. Это достигается за счет совершенной организации диспетчерской службы, наличия современного сервисного оборудования, обеспечения оперативности в удовлетворении запросов пользователей (в частности, с помощью подвижных ремонтно-технических мастерских) и т. д.

8. Способность системы к самоорганизации, самонастройке, адаптивность к изменению условий функционирования. Система называется самоорганизующейся, если в ней на основании оценки воздействия внешней среды путем последовательного изменения свойств возможен переход в некоторое устойчивое состояние, при котором воздействие внешней среды оказывается в допустимых пределах.

Принцип системного подхода, о котором говорилось выше, распространяется также на саму организацию переработки информации в СКЦО. Можно сформулировать ряд принципов, характеризующих системный подход к процессу переработки информации. Основные из них следующие.

1. Единство информационной базы. Смысл этого принципа заключается в том, что в центральной памяти системы (в ВЦ основных звеньев управления СКЦО) накапливается и постоянно обновляется общесистемная информация, необходимая для решения не какой-то одной или нескольких задач, а всех задач, решаемых в системе. Следовательно, в основных (общесистемных) информационных массивах исключается неоправданное дублирование информации, неизбежное при создании первичных информационных массивов для каждой задачи отдельно.

2. Гибкость информационной базы. Реализация этого принципа означает, что помимо основных информационных массивов, содержащих сведения, необходимые при решении любой задачи, следует иметь так называемые рабочие массивы, непосредственно используемые при решении конкретных задач. Необходимость рабочих массивов обусловлена тем, что основные массивы, будучи источниками для воспроизведения любой нужной информации в системе, сами не всегда удобны для непосредственного использования при решении конкретных задач.

Соблюдение принципов единства и гибкости информационной базы СКЦО облегчает задачу ее дальнейшего развития, расширения сети пользователей, увеличения перечня решаемых задач.

3. Стандартизация систем программирования, существенно расширяющая возможности по взаимообмену программными средствами даже в том случае, когда СКЦО строится на базе ЭВМ различных типов.

4. Автоматизация документооборота. Для реализации этого принципа необходимо все в большей степени использовать документооборот на машинных носителях.

19.2.

Структура системы комплексного централизованного обслуживания средств ЕС ЭВМ

Система комплексного централизованного обслуживания средств ЕС ЭВМ может быть представлена как сложная система, в которой выделяются следующие подсистемы:

управления КЦО;

ввода в эксплуатацию и технического обслуживания ЭВМ;

фондирования базового и прикладного программного обеспечения;

поставки, ввода в эксплуатацию и сопровождения базового и прикладного программного обеспечения;

материально-технического обеспечения КЦО;

информационного обеспечения КЦО;

подготовки специалистов по эксплуатации и обслуживанию технических и программных средств ЕС ЭВМ;

совершенствования и развития СКЦО.

Подсистема управления КЦО предназначена для управления всеми другими подсистемами СКЦО, организации и обеспечения их взаимодействия, планирования всех видов работ, составления сводных отчетов, контроля за своевременностью и качеством выполняемых работ. Структурно она представляет собой иерархическую сеть органов управления, оснащенных необходимыми техническими и программными средствами и методами управления. Основным средством управления СКЦО является автоматизированная система управления, построенная на базе ВЦ (в том числе и ВЦКП), которые имеются в организациях и структурных единицах СКЦО.

Подсистема ввода в эксплуатацию и технического обслуживания ЭВМ построена в СССР по территориально-иерархическому принципу. Основным ее производственным звеном является лаборатория технического обслуживания (ЛТО), состоящая из бригад специалистов, обслуживающих ЭВМ в своей территориальной зоне. Лаборатории объединены по принципу расширения территориальной зоны обслуживания в региональные центры обслуживания (РЦО), специализированные территориальные управления (СТУ) и производственные объединения (ПО). В рамках этой подсистемы производятся пусконаладочные, ремонтно-восстановительные и профилактические работы как в гарантийный, так и в послегарантийный периоды. Более подробно эта подсистема, а также две последующие подсистемы рассматриваются ниже.

Основные функции подсистемы материально-технического обеспечения КЦО состоят в организации снабжения всех структурных и функциональных звеньев СКЦО всем необходимым для создания и совершенствования материально-технической базы, без которой невозможно выполнение работ по комплексному централизованному обслуживанию средств ЕС ЭВМ, а также в снабжении производственных звеньев системы (ПО, СТУ, РЦО) сервисным оборудованием и ЗИП, необходимыми в производственной деятельности. Подсистема включает иерархическую

сеть органов снабжения и органов управления материально-технического обеспечения, а также сеть центральных и периферийных баз и складов, в том числе центральную базу по экспорту и центральную базу по импорту ЗИП.

Современная ЭВТ, отличающаяся многообразием технических и программных средств, тенденцией увеличения их номенклатуры и сложности разработки принципиально новых технических решений, требует все большего информирования пользователей, поставщиков и разработчиков об имеющихся и проектируемых аппаратных и программных средствах ЭВТ, об эффективности их использования, о прогнозах развития. Информационно-справочное обеспечение пользователей в условиях стремительного развития и роста производства ЭВМ должно прежде всего способствовать повышению эффективности использования машин и нахождению новых областей их применения. В настоящее время информация о программных и аппаратных средствах ЭВТ, имеющая по сути дела комплексный характер, в значительной степени разобщена, а ее сбор по всей совокупности и анализ связаны с определенными трудностями. Концентрация этой информации, ее анализ и оперативное доведение до всех заинтересованных организаций позволят резко сократить время на внедрение новых средств ЭВТ.

Решение вопросов по существенному улучшению информационно-справочного обеспечения пользователей осуществляется в рамках подсистемы информационного обеспечения КЦО и включает такие составляющие, как создание банка данных об аппаратных и программных средствах ЭВТ, ведение и обновление этого банка, информирование о нем пользователей, изучение и анализ информации о потребностях пользователей, согласование с возможностями промышленной базы и формирование банка данных о потребностях в аппаратно-программных средствах.

Создание и ведение указанных банков данных возможно потребует разработки новых принципов информационного обеспечения, определения интегральных характеристик информации и оценки информационных потоков, разработки унифицированных структур и форм входных и выходных документов.

Подготовка и переподготовка специалистов по эксплуатации технических и программных средств ЕС ЭВМ, как одна из функций СКЦО, осуществляется по трем направлениям: обслуживание технических средств, эксплуатация базового и прикладного программного обеспечения, организация использования ЭВМ.

По каждому из этих направлений введены три уровня квалификации.

При подготовке специалистов по обслуживанию технических средств предусматриваются следующие уровни квалификации:

а) электромеханик — специализируется по установке и наладке отдельных функциональных устройств ЭВМ;

б) инженер-эксплуатационник — специализируется по установке, наладке и эксплуатации групп устройств ЭВМ или однопроцессорной (одномашинной) системы;

в) системный инженер — специализируется по установке, комплексной наладке и эксплуатации машины в целом или однопроцессорной (одномашинной) системы.

Уровни квалификации при подготовке специалистов по эксплуатации систем программного обеспечения:

а) программист – специализируется по использованию одного из алгоритмических языков программирования и пакетов прикладных программ. Он должен получить сведения об операционной системе, необходимые для выполнения одиночных программ под управлением ОС;

б) проблемный программист – специализируется по комплексному использованию средств одного или нескольких алгоритмических языков программирования и операционной системы. Он должен иметь сведения об операционной системе, необходимые для компоновки сложных программ и выполнения их под управлением ОС;

в) системный программист – специализируется по обслуживанию операционных систем и разработке сложных программных систем. Он должен знать все средства операционной системы, иметь навыки обслуживания, запуска, восстановления и генерации ОС.

При подготовке специалистов по организации использования ЕС ЭВМ предусматриваются такие уровни квалификации:

а) оператор – специализируется на обеспечении запросов ОС по установке носителей информации на устройства ввода-вывода и внешней памяти, а также на подготовке данных. Он должен знать организацию ЭВМ, правила установки носителей информации и простейшего диалога с ОС, иметь навыки практической работы на устройствах подготовки данных;

б) диспетчер – специализируется на обеспечении диалога с ОС и руководит группой операторов. Он должен знать организацию и принципы функционирования ЕС ЭВМ, правила установки носителей информации, правила загрузки и обслуживания запросов операционной системы;

в) инженер-системотехник – специализируется по вопросам обеспечения эффективного использования вычислительных систем. Он обязан знать принцип организации и функционирования вычислительных систем и вычислительных центров, а также структуру и функционирование операционной системы.

Существует довольно большое количество специализаций при подготовке специалистов как по техническим, так и по программным средствам ЭВМ. Эти специализации непрерывно обновляются (особенно по техническим средствам) по мере развития и совершенствования электронной вычислительной техники.

Большие и актуальные научно-технические задачи решаются в рамках подсистемы совершенствования и развития СКТО, которая в СССР представлена научными предприятиями НОТО страны. К числу основных научно-технических проблем, решаемых коллективами этой подсистемы, относятся: разработка АСУ СКТО средств ЕС ЭВМ, разработка и совершенствование нормативно-технологической базы КТО, разработка новых приборов и сервисного оборудования для подвижных ремонтно-технических мастерских и ремонтно-восстановительного производства, разработка и обоснование новых принципов, методов и форм КТО технических и программных средств ЕС ЭВМ (включая вопросы комплексной поставки ЭВМ), разработка проектов типовых модулей ВЦ, решение вопросов автоматизации работы фондов алгоритмов и программ базового и при-

кладного программного обеспечения ЕС ЭВМ, построение автоматизированных систем подготовки специалистов и т. д. Некоторые из этих проблем более подробно обсуждаются ниже.

19.3. Организация ввода в эксплуатацию и технического обслуживания ЕС ЭВМ

Кроме проведения пусконаладочных, ремонтно-восстановительных и профилактических работ, в рамках подсистемы ввода в эксплуатацию и технического обслуживания ЕС ЭВМ предоставляются услуги пользователям по подготовке помещений, предназначенных для размещения ЭВМ или оборудования ВЦ.

Обеспечение полной готовности помещений для установки в них ЭВМ является обязанностью пользователей, т. е. предприятий и организаций, куда поставляются ЭВМ. Для облегчения и унификации этих работ предоставляется новый вид услуг пользователям: проектирование размещения технических средств ЭВМ Единой системы и типового внутреннего оформления помещений ВЦ для них.

Проектирование для ВЦ любой структуры и конфигурации технических средств осуществляется путем подбора необходимых проектов типовых модулей ВЦ. Типовой модуль ВЦ это конструктивно законченный узел, выполненный на базе специальных конструкций ВЦ. Типовые модули ВЦ предназначены для размещения и компоновки технических средств ЕС ЭВМ и технологического оборудования. Состав технических средств и технологического оборудования каждого типового ВЦ определяется в зависимости от его функционального назначения. В состав типового модуля могут входить элементы систем, обеспечивающих необходимые режимы и условия работы технических средств ЭВМ и технологического оборудования. Например, в состав типового модуля машинного зала кроме технических средств ЭВМ входят конструкции воздуховодов системы кондиционирования, радиаторы и трубопровод системы отопления ВЦ, щиты и внутренняя разводка системы электроснабжения, светильники и внутренняя разводка системы освещения, шины схемного и защитного заземления, часы и соответствующая проводка, аппаратура для системы внутренней связи ВЦ и ее разводка внутри машинного зала, транспортные средства и элементы автоматики. Типовые модули ВЦ соединяются между собой посредством межмодульных коммутационных сетей.

По подготовке ВЦ пользователям предоставляются в процессе КЦО средств ЕС ЭВМ следующие услуги:

проведение предпроектного обследования помещений, предлагаемых под ВЦ, и поставка пользователю типового проекта ВЦ, составленного из набора необходимых проектов типовых модулей;

привязка проектов типовых модулей к имеющимся у пользователя архитектурно-строительным решениям. Она заключается в корректировке этих модулей в соответствии с конкретными помещениями, составом технических средств ЭВМ и технологического оборудования, а также в проектировании межмодульных коммутационных сетей;

сопровождение типового проекта, которое заключается в постоянном контроле за ходом строительства и оборудования ВЦ.

Основные объемы работ, которые выполняются производственными звеньями рассматриваемой подсистемы, включают следующие виды технического обслуживания:

ввод в эксплуатацию технических средств ЭВМ (пусконаладочные работы);

проведение комплекса профилактических работ по техническим средствам (ежедневное, еженедельное, месячное, квартальное, полугодовое и годовое профилактическое обслуживание);

ремонтно-восстановительные работы в гарантийный и послегарантийный периоды.

Взаимоотношения между СКЦО (исполнителем) и пользователями в процессе технического обслуживания ЭВМ строятся на договорной основе. Постановка ЭВМ пользователя на КЦО осуществляется с момента удовлетворения его заявки на приобретение ЭВМ или с любого другого момента эксплуатации ЭВМ. При выполнении услуг по вводу ЭВМ в эксплуатацию пользователь обязан обеспечить полную строительную готовность помещений, предназначенных для размещения технических средств. Техническое обслуживание ЭВМ пользователя осуществляется исполнителем на основании «Договора на выполнение работ по техническому обслуживанию электронной вычислительной техники». Все расчеты между пользователем и исполнителем за выполненные профилактические и ремонтно-восстановительные работы производятся на основании акта и ведомости израсходованных запасных частей и материалов. В гарантийный период ремонт ЭВМ выполняется исполнителем за счет средств завода-изготовителя, в послегарантийный период – за счет средств пользователя.

Для повышения оперативности и качества работ по техническому обслуживанию ЭВМ производственные бригады ЛТО снабжаются подвижными ремонтно-техническими мастерскими (ПРТМ), построенными на базе автомашин типа РАФ, УАЗ, ГАЗ-66, оснащенных необходимыми контрольно-измерительными приборами, сервисным оборудованием и запасными частями. Имеются несколько вариантов таких ПРТМ. Работа по совершенствованию их оборудования и технологии применения продолжается.

Необходимо отметить, что, несмотря на наличие подвижных средств технического обслуживания (ПРТМ), производственные бригады, как правило, не в состоянии проводить ежедневное профилактическое обслуживание ЭВМ своего региона. Более того, при малой насыщенности ЭВМ в регионе и большой их территориальной разобщенности не всегда удается проводить и еженедельное профилактическое обслуживание. В этом случае такие виды работ выполняются силами обслуживающего персонала пользователей, о чем делается специальная оговорка в договоре.

Перспективы развития методов и средств технического обслуживания ЕС ЭВМ на ближайшие годы определяются главным образом совершенствованием самих машин и изменением методов и форм их использования (переход от автономно используемых ЭВМ и ВЦ к ВЦКП и сетям ЭВМ). Одной из первоочередных задач является переоснащение персонала

ЛТО более совершенными ПРТМ, а также миниатюрными средствами диагностики элементов и устройств ЭВМ, средствами дистанционной диагностики, новыми программными средствами проверки технического состояния ЭВМ.

Совершенствование технического обслуживания ЭВМ требует решения ряда научных проблем, таких как разработка и создание автоматизированной системы учета и распределения ЗИП, определение оптимального состава ЗИП по объему и номенклатуре, организация автоматизированной работы складов, разработка автоматизированной системы учета, планирования и управления профилактическими работами, разработка типовых технологических процессов обслуживания новых моделей ЭВМ, стандартизация этих процессов и т. д.

Принципиально важное значение имеют работы по созданию новой контрольно-измерительной аппаратуры, сервисного оборудования и инструментов. Имеющиеся средства такого назначения, а также проверочные и диагностические программы, не позволяют вести восстановительные работы достаточно быстро и эффективно, так как не обеспечивают быструю локализацию неисправностей с необходимой точностью.

Проверка функционирования логических схем и поиск неисправностей с помощью существующих обычных контрольно-измерительных приборов, основанных на частотно-временных принципах (вольтметры, осциллографы, частотомеры и т. д.), при широком использовании интегральных и больших интегральных схем в машинах третьего и четвертого поколений малоэффективны, поскольку несовместимы основополагающие начала — единицы измерения. В связи с этим для обеспечения эффективного поиска неисправностей в ЭВМ необходимы разработка и внедрение информационных контрольно-измерительных приборов принципиально нового класса, совместимых с ЭВМ по информационным принципам на основе общих единиц измерения — битов, байтов, слов. Такая информационная контрольно-измерительная аппаратура, оперирующая информационными единицами измерения, более удобна и эффективна для анализа цифровых устройств, так как представляет в удобной для восприятия форме наиболее важную для оценки работоспособности этих устройств информацию о последовательности выполнения программы и результатах операций над данными.

В течение последних лет ведущие зарубежные фирмы, занимающиеся исследованиями и разработкой новых средств для технического обслуживания ЭВМ, создали и внедрили ряд принципиально новых, высокоэффективных методов контроля и диагностики электронной вычислительной техники, таких как логический анализ, проверка схемных плат путем моделирования их работы, контроль избыточных переходов, внутрисхемная эмуляция, сигнатурный анализ и др. Эти методы реализованы в аппаратуре контроля и диагностики, которая включает портативные тестеры контроля передачи данных, логические анализаторы, тестеры схемных плат и микропроцессорного оборудования, универсальные тестеры для интерфейсных шин, тестеры периферийного оборудования и устройств связи, блоки диагностики для интерактивных систем, специальные приспособления для настройки отдельных узлов и др.

Аналогичные исследования по созданию новых методов и средств

контроля и диагностики проводятся в странах – участницах Соглашения в области электронной вычислительной техники. Эти исследования направлены прежде всего на обеспечение автоматизации и механизации работ по техническому обслуживанию ЭВМ, связанных с локализацией неисправностей машин, дистанционного контроля и проверки работоспособности ЭВМ через обычные коммутируемые каналы связи, сокращения номенклатуры средств и оснастки путем создания многофункциональной аппаратуры, приспособлений и инструментов. При этом предъявляются требования по значительному снижению стоимости, размеров и массы создаваемых средств.

19.4. Фонд программ и служба сопровождения

В процессе функционирования подсистема фондирования базового и прикладного программного обеспечения и подсистема поставки, ввода в эксплуатацию и сопровождения его настолько тесно взаимодействуют, что это послужило основанием объединить их под одним названием – фонд программ и служба сопровождения (ФПСС).

Фондирование базового и прикладного программного обеспечения осуществляется фондом алгоритмов и программ (ФАП), который представляет собой систему организаций, взаимодействующих между собой по вопросам фондодержательской деятельности, а также с организациями-разработчиками программного обеспечения и организациями-пользователями ЭВМ. Имеются головное отделение ФАП и региональные отделения, рассредоточенные по территории страны.

К фондодержательской деятельности ФАП относится его работа по сбору сведений о программном обеспечении, разработанном как в СССР, так и в других странах, работа по приему и хранению базового программного обеспечения (БПО) и пакетов прикладных программ (ППП) общего назначения, а также по организации и ведению информационной и консультационной службы по БПО и ППП. ФАП содержит пригодные для непосредственного использования на ЭВМ системы программного обеспечения на носителях данных с соответствующими комплектами документации. Кроме ДОС ЕС и ОС ЕС, в ФАП имеются следующие типы ППП: расширяющие возможности ОС, для научных приложений, для инженерных приложений, для информационно-поисковых систем и баз данных, для реализации различных математических методов решения задач.

Прием программных средств от организаций-разработчиков или из других фондов осуществляется в порядке совместного сотрудничества и технической помощи. Между ФАП и этими организациями заключается двусторонний договор, оговаривающий взаимные обязательства сторон, порядок, условия и сроки передачи программных средств в фонд. Пользователями фонда алгоритмов и программ являются все организации, применяющие в своей работе ЭВМ и эксплуатирующие программные средства фонда. Поставка программных средств пользователям производится через подразделения службы сопровождения этих средств.

Подсистема поставки, ввода и сопровождения программных средств по своей организационной структуре аналогична подсистеме ввода и технического обслуживания ЭВМ. Она тоже построена по территориально-иерархическому принципу, в основу которого положены иерархичность функционально-производственных связей, территориальная распределенность производственных подразделений, хозрасчетная система взаимодействия с пользователями, оперативность обслуживания.

Первый уровень этой подсистемы представляют Головная служба сопровождения (ГСС) и ФАП, второй уровень — отделы и лаборатории программного обеспечения при СТУ и ПО, третий уровень — лаборатории и бюро программного обеспечения при РЦО.

Основные задачи подразделений первого уровня состоят в следующем: плановое накопление и тиражирование программных средств (БПО и ППП), ведение подлинников документации и носителей данных фондируемого и поставляемого программного обеспечения, устранение ошибок и совершенствование сопровождаемых программных средств, внесение изменений в подлинники программных средств по результатам исправления ошибок и совершенствования, выпуск информационных, рекламных и справочных материалов по программному обеспечению, взаимодействие с организациями-разработчиками программных средств и заводами-изготовителями ЭВМ.

Задачи подразделений второго и третьего уровней: обеспечение (поставка и ввод в эксплуатацию) пользователей программными средствами; генерация этих средств (настройка программных средств под конфигурацию технических средств ЭВМ и задачи пользователей); сопровождение БПО и ППП; прием от пользователей информации и материалов об ошибках в программных средствах и передача их для обобщения и устранения в ГСС; предоставление пользователям изменений, связанных с устранением ошибок или совершенствованием сопровождаемых программных средств; информирование пользователей о новых программных средствах и проведение консультаций.

Для обеспечения работы подразделений трех указанных уровней необходима большая и объемная работа по тиражированию эксплуатационной документации и носителей данных, которая проводится специальным производственным предприятием, функционирующим в рамках подсистемы поставки, ввода и сопровождения программных средств.

Рассмотренные выше вопросы организации и совершенствования системы комплексного централизованного обслуживания средств электронной вычислительной техники имеют прямое отношение к решению задач по дальнейшему совершенствованию обслуживания взаимопоставляемых технических и программных средств ЭВМ в рамках национальных организаций технического обслуживания стран — участниц Соглашения.

В области технического обслуживания ЭВМ (и прежде всего ЕС ЭВМ) такими задачами являются:

разработка единых принципов и условий обслуживания технических средств ЕС ЭВМ в странах — участницах Соглашения, разработка единой нормативно-технологической базы по вопросам технического обслуживания;

разработка предложений по усовершенствованию измерительной аппаратуры для средств ЕС ЭВМ, а также комплекса диагностических

программ для обеспечения ввода в эксплуатацию и технического обслуживания этих средств;

организация единой системы сбора и анализа данных о надежности технических средств ЕС ЭВМ, обобщение их и разработка рекомендаций по повышению качества и надежности ЭВМ;

организация централизованной аварийной и диспетчерской службы для устранения аварийных ситуаций;

организация единой службы извещения об изменениях технических средств и документации на них.

В области программного обеспечения необходимо радикальное решение вопросов по внедрению единых принципов и нормативов при вводе и сопровождении БПО и ППО. В этой области важное значение имеет организация действенной международной информационно-справочной службы, а также взаимное участие специалистов по программным средствам при совместных испытаниях программ.

Для обеспечения эффективного использования ЕС ЭВМ, особенно новых моделей, поступающих в эксплуатацию, необходимо проведение в больших масштабах опережающей подготовки специалистов. Здесь важно добиться унификации программ, создания единой учебно-методической базы, координации работ по написанию и изданию учебной документации к средствам ЕС ЭВМ.

Решение рассмотренных вопросов по организации и совершенствованию комплексного централизованного обслуживания технических и программных средств электронной вычислительной техники будет способствовать повышению эффективности использования парка ЭВМ в странах – участницах Соглашения.

ПРИЛОЖЕНИЕ

Таблица 1

Основные характеристики ряда моделей ЕС ЭВМ

Характеристика	Модель ЕС ЭВМ и страна-разработчик						
	ЕС-1012, ВНР	ЕС-1022, СССР	ЕС-1032, ПНР	ЕС-1033, СССР	ЕС-1040, ГДР	ЕС-1050, СССР	ЕС-1052, СССР
Среднее быстродействие машины, тыс. операций в 1 с	6	80	180	150–200	380	500	700
Разрядность процессора, байт	2	2	4	4	8	8	8
Основная оперативная память:							
емкость, кбайт	До 64	128–512	128–1024	256–512	128–512	256–1024	1024
время цикла, мкс	0,8	2	1,2	1,2	1,35	1,25	1,25
время выборки, мкс	0,4	0,95	0,7	0,7	0,45	0,8	0,65
ширина выборки, байт	2	4	4	4	8	8	16
Мультиплексный канал:							
число мультиплексных подканалов	16–64	128	256	128	128–256	196	224
скорость в мультиплексном режиме, кбайт/с	20	80	145	70	20–25	110–670	425
скорость в монопольном режиме, кбайт/с	50	400	470	350	180–720	180–720	
число подключаемых блоков управления	8	18	8	8	10	40	8
Селекторные каналы:							
число каналов	1	2	3	3	6	6	2
скорость канала, кбайт/с	240	500	1100	800	1300	1300	1250
число подключаемых блоков управления	8	10	8	8	10	8	8
Операционная система	ОС-12	ДОС; ОС	ДОС; ОС	ДОС; ОС	ДОС; ОС	ОС	ОС
Потребляемая мощность, кВ·А	4	25	23	40	65	70	60
Занимаемая площадь, м^2	12	108	80	120	150	230	230

Таблица 2

КОДЫ ОПЕРАЦИЙ

1. Команды формата RR

Младшая тетрада кода операции	Операции			
	передачи управления, переключения состояния	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 XXXX	0001 XXXX	0010 XXXX	0011 XXXX
0000		Загрузка положительного — Load Positive*; LPR**	Загрузка положительного — Load Positive (Long); LPDR	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); LCER
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 возвратом — Branch and Link; BALR	Сравнение логическое — Compare Logical; CLR		
0110	Передача управления по счетчику — Branch on Count; BCTR	Или — Or; OR		

Продолжение табл. 2

Младшая тетрада кода операции	Операции			
	передачи управления, переключения состояния	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
0111	Передача управления по условию – Branch on Condition; BCR	Исключающее ИЛИ – Exclusive Or; XR		
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

Продолжение табл. 2

2. Команды формата RX

Младшая тетрада кода операции	Операции			
	с фиксированной запятой, производимые над полусловами (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	0100 XXXX	0101 XXXX	0110 XXXX	0111 XXXX
0000	Запись в память – Store Halfword; STH	Запись в память – Store; ST	Запись в память – Store (Long); STD	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		
0101	Передача управления с возвратом – Branch and Link; BAL	Сравнение логическое – Compare Logical; CL		
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

Продолжение табл. 2

Младшая тетрада кода операции	Операции			
	с фиксированной запятой, производимые над полусловами (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
1010	Сложение полуслова – Add Halfword; AH	Сложение – Add; A	Сложение (с нормализацией) – Add Normalized (Long); AD	Сложение (с нормализацией) – Add Normalized (Short); AE
1011	Вычитание полуслова – Subtract Halfword; SH	Вычитание – Subtract; S	Вычитание (с нормализацией) – Subtract Normalized (Long); SD	Вычитание (с нормализацией) – Subtract Normalized (Short); SE
1100	Умножение полуслова – Multiply Halfword; MH	Умножение – Multiply; M	Умножение – Multiply (Long); MD	Умножение – Multiply (Short); ME
1101		Деление – Divide; D	Деление – Divide (Long); DD	Деление – Divide (Short); DE
1110	Преобразование в десятичную – Convert to Decimal; CVD	Сложение логическое – Add Logical; AL	Сложение (без нормализации) – Add Unnormalized (Long); AW	Сложение (без нормализации) – Add Unnormalized (Short); SU
1111	Преобразование в двоичную Convert to Binary; CVB	Вычитание логическое – Subtract Logical; SL	Вычитание (без нормализации) – Subtract Unnormalized (Long); SW	Вычитание (без нормализации) – Subtract Unnormalized (Short); SU

Продолжение табл. 2

3. Команды формата RS, SI

Младшая тетрада кода операции	Операции			
	передачи управления, переключения состояния и сдвига	с фиксированной запятой, логические и ввода-вывода	–	–
^XXX	1000 XXXX	1001 XXXX	1010 XXXX	1011 XXXX
0000	Установка маски системы – Set Sistem Mask; SSM	Запись в память групповая – Store Multiply; STM		
0001		Проверка по маске – Test under Mask; TM		
0010	Загрузка ССП (слова состояния программы) – Load Program Status Word; LPSW	Пересылка непосредственная – Move Immediatle; MVI		
0011	Диагностика – Diagnose	Проверка и установка – Test and Set; TS		
0100	Прямая запись – Write Direct; WRD	И непосредственное – And Immediatle; NI		
0101	Прямое чтение – Read Direct; RDD	Сравнение непосредственное – Compare Logical Immediatle; CLI		
0110	Передача управления по индексу «больше» – Branch on Index High; BXH	ИЛИ непосредственное – Or Immediatle; OI		

Продолжение табл. 2

Младшая тетрада кода операции	Операции			
	передачи управления, переключения состояния и сдвига	с фиксированной запятой, логические и ввода-вывода	–	–
0111	Передача управления по индексу «меньше» или «равно» – Branch on Index Low or Equal; BXLE	Исключающее ИЛИ непосредственное – Exclusive Or Immediatle; XI		
1000	Логический сдвиг слова вправо – Shift Right Single Logical; SRL	Загрузка групповая – Load Multiply; LM		
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			
1100	Логический сдвиг двойного слова вправо – Shift Right Double Logical; SRDL	Пуск ввода-вывода – Start I/O; SIO		
1101	Логический сдвиг двойного слова влево – Shift Left Double Logical; SLDL	Опрос ввода-вывода – Test I/O; TIO		
1110	Арифметический сдвиг двойного слова вправо – Shift Right Double Arithmetic; SRDA	Останов ввода-вывода – Halt I/O; HIO		
1111	Арифметический сдвиг двойного слова влево – Shift Left Double Arithmetic; SLDA	Опрос канала – Test Channel; TCH		

Продолжение табл. 2

4. Команды формата SS

Младшая тетрада кода операции	Операции			
	—	логические и пересылочные над полями переменной длины	—	десятичные
XXXX	1100 XXXX	1101 XXXX	1110 XXXX	1111 XXXX
0000				
0001		Пересылка цифр — Move Numerics; MVN		Пересылка со сдвигом — Move with Offset; MVO
0010		Пересылка символов — Move Characters; MVC		Упаковка — Pack; PACK
0011		Пересылка зон — Move Zones; MVZ		Распаковка — Unpack; UNPK
0100		И — And; NC		
0101		Сравнение кодов — Compare Logical; CLC		
0110		ИЛИ — Or; OC		
0111		Исключающие ИЛИ — Exclusive Or; XC		
1000				Очистка и сложение — Zero and Add; ZAP
1001				Сравнение десятичных чисел — Compare Decimal; CP

Продолжение табл. 2

Младшая тетрада кода операции	Операции			
	–	логические и пересылочные над полями переменной длины	–	десятичные
1010				Десятичное сложение – Add Decimal; AP
1011				Десятичное вычитание – Subtract Decimal; SP
1100		Перекодирование – Translate; TP		Десятичное умножение – Multiply Decimal; MP
1101		Перекодирование и проверка – Translate and Test; TRT		Десятичное деление – Divide Decimal; DP
1110		Редактирование – Edit; ED		
1111		Редактирование и отметка – Edit and Mark; EDMK		

Примечание. Указанные в таблице мнемонические обозначения операций приняты и в ЕС ЭВМ.

* Наименование операции, принятое в американской системе IBM-360.
** Условное мнемоническое обозначение операции, применяемое при программировании на языке АССЕМБЛЕР.

Таблица 3

Графические знаки

Знак	Наименование	Знак	Наименование
[	Квадратная скобка (левая)	,	Запятая
.	Точка	%	Процент
<	Меньше	_	Подчеркивание
(	Круглая скобка (левая)	>	Больше
+	Плюс	?	Вопросительный знак
!	Восклицательный знак	^	Диакритический знак
&	Коммерческое *И*	:	Двоеточие
]	Квадратная скобка (правая)	#	Номер
¤	Знак денежной единицы	@	Коммерческое ЭТ
*	Звездочка	'	Апостроф
)	Круглая скобка (правая)	=	Равно
;	Точка с запятой	»	Кавычки
¬	Логическое *НЕ*	‾	Черта сверху
—	Минус	{	Фигурная скобка (левая)
/	Дробная черта	}	Фигурная скобка (правая)
|	Логическое *ИЛИ* (черта вертикальная)		

Таблица 4

Стандартные управляющие символы

Позиция в кодовой таблице ДКОИ	Сокращенное обозначение символа		Наименование символа	
	русское	международное	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/B	ВТ	VT	Вертикальная табуляция	Vertical Tabulation
0/C	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/C	РФ	FS	Разделитель файлов	File Separation
1/D	РГ	GS	Разделитель групп	Group Separation
1/E	РЗ	RS	Разделитель записей	Record Separation
1/F	РЭ	US	Разделитель элементов	Unit Separation
2/5	ПС	LF	Перевод строки	Line Feed
2/6	КБ	ETB	Конец блоков	End of Transmission Block

Продолжение табл. 4

Позиция в кодовой таблице ДКОИ	Сокращенное обозначение символа		Наименование символа	
	русское	международное	русское	международное
2/7	АР2	ESC	Авторегистр 2	Escape
2/D	КТМ	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

Таблица 5

Дополнительные управляющие символы

Позиция в кодовой таблице ДКОИ	Сокращенное обозначение символа		Наименование символа	
	русское	международное	русское	международное
0/4	ВЫП	PF	Выключение перфоратора	Punch off
0/6	НП	LC	Нижнее положение	Lower Case
0/A	НРВ	SMM	Начало ручного ввода	Start of Manual Massage
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
1/B	СП1	CU1	Символ пользователя 1	Customer Use 1
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
2/B	СП2	CU2	Символ пользователя 2	Customer Use 2
3/4	ВКП	PN	Включение перфоратора	Punch On
3/5	ОСУ	RS	Остановка считывающего устройства	Reader Stop
3/6	ВП	UC	Верхнее положение	Upper Case
3/B	СП3	CU3	Символ пользователя 3	Customer Use 3

СПИСОК ЛИТЕРАТУРЫ

1. **Абси О. И., Каган Я. А.** Математические модели сложных вычислительных систем.— Автоматика и телемеханика, 1971, № 1, с. 109—127.
2. **Альянах И. Н.** Внешние запоминающие устройства ЕС ЭВМ. М.: Сов. радио. 1979. 208 с.
3. **Белоногов Г. Г., Богатырев В. И.** Автоматизированные информационные системы. М.: Сов. радио, 1973, 328 с.
4. **Булей Г.** Микропрограммирование. М.: Мир, 1973. 128 с.
5. **Бурцев В. С.** Перспективы создания ЭВМ высокой производительности.— Вестник АН СССР, 1975, № 3, с. 25—31.
6. **Бусленко Н. П.** Моделирование сложных систем. М.: Наука, 1968. 336 с.
7. **Васильев Б. В., Козлов Б. А., Ткаченко Л. Г.** Надежность и эффективность радиоэлектронных устройств. М.: Сов. радио, 1964, 368 с.
8. **Вычислительная** система IBM 360. Принципы работы. М.: Сов. радио, 1969. 440 с.
9. **Вычислительные** машины с развитыми системами интерпретации/Под ред. В. М. Глушкова. Киев: Наукова думка, 1970. 260 с.
10. **Глушков В. М.** Введение в АСУ. Киев: Техніка, 1974, 320 с.
11. **Глушков В. М.** Проблемная ориентация и другие пути повышения эффективности ЭВМ.— Вестник АН СССР, 1975, № 3, с. 18—24.
12. **Глушков В. М., Кушнер Э. Ф., Стогний А. А.** Функциональная структура и элементы сетей ЭВМ.— Управляющие системы и машины, 1975, № 3 (17), с. 1—10.
13. **Голубев-Новожилов Ю. С.** Многомашинные комплексы вычислительных средств. М.: Сов. радио, 1967, 424 с.
14. **Данилочкин В. П., Пеледов Г. В.** Система обработки данных в реальном масштабе времени на базе ЕС ЭВМ.— Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 3, с. 33—42.
15. **Джейсуол Н.** Очереди с приоритетами. Пер. с англ./Под ред. В. В. Калашникова. М.: Мир, 1973. 280 с.
16. **Джеймейн К.** Программирование на IBM 360. М.: Мир, 1971, 870 с.
17. **Дроздов Е. А., Комарницкий В. А., Пятибратов А. П.** Многопрограммные цифровые вычислительные машины. М.: Воениздат, 1974, 408 с.
18. **Дроздов Е. А., Пятибратов А. П.** Основы построения и функционирования вычислительных систем. М.: Энергия, 1973. 368 с.
19. **Дроздов Е. А., Комарницкий В. А., Пятибратов А. П.** Электронные вычислительные машины Единой системы. М.: Машиностроение, 1976. 672 с.
20. **Единая** система ЭВМ/Под ред. А. М. Ларионова. М.: Статистика, 1974. 136 с.
21. **Каган Б. М.** Электронные вычислительные машины и системы. М.: Энергия, 1979, 528 с.
22. **Каналы** ввода-вывода ЭВМ ЕС-1020/Под ред. А. М. Ларионова. М.: Статистика, 1976. 272 с.
23. **Каталог** ЕС ЭВМ. Т. 1. Технические средства. М.: Стройиздат, 1979. 160 с.
24. **Каталог** ЕС ЭВМ. Т. 2. Программное обеспечение. М.: Стройиздат, 1978, 72 с.
25. **Клямко Э. И.** Схемный и тестовый контроль автоматических цифровых вычислительных машин. М.: Сов. радио, 1963. 192 с.
26. **Колин К. К., Липаев В. В.** Проектирование алгоритмов управляющих ЦВМ. М.: Сов. радио, 1970. 344 с.
27. **Королев Л. Н.** Структуры ЭВМ и их математическое обеспечение. М.: Наука, 1978. 352 с.
28. **Крайзмер Л. П.** Быстродействующие ферромагнитные запоминающие устройства. М.—Л.: Энергия, 1964, 372 с.
29. **Крайзмер Л. П.** Устройства хранения дискретной информации. М.: Энергия, 1969. 312 с.
30. **Майоров С. А., Новиков Г. И.** Принципы организации цифровых машин. М.: Машиностроение, 1974. 432 с.
31. **Макурочкин В. Г.** Внешние запоминающие устройства ЕС ЭВМ.— Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 3, с. 60—71.
32. **Мачулин В. В., Пятибратов А. П.** Эффективность систем обработки информации. М.: Сов. радио, 1972. 280 с.

33. **Мультипроцессорные** системы и параллельные вычисления. Пер. с англ./ Пер. Ю. С. Голубева-Новожилова и А. Л. Шерса;/Под ред. Ф. Г. Энслоу. М.: Мир, 1976. 384 с.
34. **Операционная** система ДОС ЕС. Общие положения/М. Р. Шура-Бура, Э. В. Ковалевич, М. С. Марголин и др. М.: Статистика, 1975. 120 с.
35. **Поспелов Д. А.** Введение в теорию вычислительных систем. М.: Сов. радио, 1972. 280 с.
36. **Принципы** системной организации ЭВМ Единой системы /А. М. Ларионов, В. К. Левин, С. П. Соловьев и др.— Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 1, с. 55—65.
37. **Проектирование** свойств эксплуатационной надежности для системы IBM 360.— Кибернетический сборник. Новая сер., вып. 1, 1965, с. 137—161.
38. **Процессор** ЭВМ ЕС-1020/Под ред. А. М. Ларионова. М.: Статистика, 1975. 160 с.
39. **Путинцев Н. Д.** Аппаратный контроль управляющих цифровых вычислительных машин. М.: Сов. радио, 1966. 420 с.
40. **Селлерс Ф.** Методы обнаружения ошибок в работе ЭЦВМ. М.: Мир, 1972. 312 с.
41. **Семенихин В. С., Ларионов А. М., Лапин В. С.** Средства и сети телеобработки данных ЕС ЭВМ.— Вычислительные средства в технике и системах связи. М.: Связь, 1978, вып. 3, с. 79—95.
42. **Сети** ЭВМ коллективного пользования.— Экспресс-информация. Сер. ВТ, 1975, № 16, с. 16—21.
43. **Система** ввода-вывода ЕС ЭВМ. Принципы организации/А. М. Ларионов, В. К. Левин, С. П. Соловьев и др.— Вопросы радиоэлектроники. Сер. ЭВТ. 1973, вып. 1, с. 30—39.
44. **Система** документации Единой системы ЭВМ/Под ред. А. М. Ларионова. М.: Статистика, 1976. 328 с.
45. **Система** математического обеспечения ЕС ЭВМ/Под ред. А. М. Ларионова. М.: Статистика. 1974. 216 с.
46. **Системы** передачи данных и сети ЭВМ/Под ред. П. Грина и Р. Лаки. М.: Мир, 1974. 216 с.
47. **Система** телеобработки данных ЕС ЭВМ/В. С. Лапин, В. С. Антонов, Ю. Г. Данилевский и др.— Вопросы радиоэлектроники. Сер. ЭВТ. 1973, вып. 1, с. 40—54.
48. **Справочник** по полупроводниковым диодам, транзисторам и интегральным схемам/Под общей ред. Н. Н. Горюнова. Изд. 4-е, перераб. и доп. М.: Энергия, 1977. 744 с.
49. **Средства** управления данными и перспективы их развития в ОС ЕС ЭВМ/ Е. Х. Еникеева, В. Г. Лесюк, Л. Д. Райков и др.— Вопросы радиоэлектроники. Сер. ЭВТ. 1973, вып. 1, с. 23—32.
50. **Управляющая** программа ОС ЕС ЭВМ/Л. Б. Лебедев, В. В. Наумов, Г. В. Пеледов и др.— Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып, 3, с. 15—22.
51. **Флорес А.** Внешние устройства ЭВМ. М.: Мир, 1977. 550 с.
52. **Флорес А.** Организация вычислительных машин. М.: Мир, 1972. 428 с.
53. **Флорес А.** Программное обеспечение. М.: Мир, 1971. 520 с.
54. **Шаруненко Н. М.** Запоминающие устройства ЕС ЭВМ.— Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 3, с. 72—95.
55. **Шелихов А. А., Селиванов Ю. П.** Вычислительные машины. Справочник. М.: Энергия, 1978. 224 с.
56. **Шигин А. Г., Дерюгин А. А.** Цифровые вычислительные машины (память ЦВМ). М.: Энергия, 1975, 536 с.
57. **Электронная** вычислительная машина ЕС-1020/Под ред. А. М. Ларионова. М.: Статистика, 1975, 128 с.
58. **Электронная** вычислительная машина ЕС-1030/Под ред. А. М. Ларионова. М.: Статистика. 1977. 256 с.
59. **Электронная** вычислительная машина ЕС-1050/Под ред. А. М. Ларионова. М.: Статистика, 1976. 304 с.