

Электронная
вычислительная
машина

ЕС-1033

Электронная вычислительная машина

ЕС-1033

Под редакцией

В. А. КОМАРНИЦКОГО, Г. П. СОРОКИНА



ББК 32.972

Э45

УДК 681.3

Г. П. Сорокин, В. Н. Иванов, А. У. Ярмухаметов,
В. Ф. Гусев, Г. Н. Иванов, В. В. Фадеев, А. Х. Абдрахманов,
Г. М. Персов, Л. И. Нейман, Ю. А. Каршев, В. И. Кельдышер,
В. И. Коротышкин, Д. А. Касимова, Т. А. Петрова

Редактор Е. Е. СКВОРЦОВ

Рецензенты: Е. А. ДРОЗДОВ, А. Г. ШЕВЧЕНКО

Э45 **Электронная вычислительная машина ЕС-1033/Под ред.
В. А. Комарницкого, Г. П. Сорокина. — М.: Машиностроение,
1982. 383 с., ил.**

В пер.: 1 р. 60 к.

Книга посвящена ЭВМ ЕС-1033, предназначенной для решения широкого круга научно-технических и экономических задач. Приведены общие сведения о модели, рассмотрены ее структурные и технические особенности, элементная база, принципы построения и функционирования узлов и блоков центрального процессора, каналов ввода-вывода и системы диагностики, описаны алгоритмы выполнения операций, структура и возможности программного обеспечения.

Книга предназначена для разработчиков технических средств вычислительной техники, микропрограммного и программного обеспечения и для работников вычислительных центров. Она будет полезной для научных работников, преподавателей, студентов вузов и аспирантов, специализирующихся в названных областях.

Э $\frac{2405000000-062}{038(01)-82}$ 62-82

ББК 32.972

6Ф7.3

Основной задачей развития социалистического общества, которая изложена в решениях XXVI съезда КПСС, является подъем материального и культурного уровня жизни народа на основе динамического и пропорционального развития общественного производства и повышения его эффективности, роста производительности труда, всемерного улучшения качества работы. Решение этой задачи тесно связано с ускорением темпов научно-технического прогресса, характерной особенностью которого является широкое применение математических методов и средств вычислительной техники во всех областях народного хозяйства.

Постоянно возрастающие потребности народного хозяйства обуславливают необходимость создания и применения все более современных электронных вычислительных машин. В частности, это послужило причиной модернизации первых базовых моделей ЭВМ Единой системы: ЕС-1020, ЕС-1030, ЕС-1050. На смену им были разработаны и запущены в серийное производство новые модели: ЕС-1022, ЕС-1033, ЕС-1052. Они превосходят своих предшественниц по всем основным показателям.

На фоне прочих разработок выделяется модель ЕС-1033, которая относится к машинам средней производительности семейства «Ряд 1» ЕС ЭВМ. Анализ проблем, стоящих перед отечественными и зарубежными пользователями ЭВМ этого семейства, позволил определить требования, которым должна удовлетворять ЭВМ средней производительности: увеличение основного показателя — отношения производительности к стоимости, сокращение объема и номенклатуры компонентов основного электронного оборудования в сравнении с существующими моделями рассматриваемого класса и повышение достоверности результатов обработки информации. Планируемые характеристики должны базироваться на освоенной и сравнительно недорогой технологии, а ЭВМ — обладать достаточной патентной чистотой и конкурентоспособностью на внешнем рынке.

Проведенные исследования показали, что традиционные структуры ЭВМ средней производительности не позволяют добиться значительного улучшения их характеристик. Для достижения этой цели требуются разработка и применение новых принципов структурной организации вычислительных средств. Отмеченное обстоятельство обусловило необходимость поиска пути построения модели ЕС-1033.

В основу построения ЭВМ ЕС-1033 была положена магистральная структура, которая ранее не применялась для машин такого класса [А. с. 486711 (СССР)].

Структура вычислительной машины в значительной степени определяется теми техническими средствами, которые используются при ее построении. Поэтому реализация нетрадиционной структуры вызвала необходимость поиска и применения ряда оригинальных технических решений и приемов микропрограммирования. Они защищены 17 авторскими свидетельствами и 65 зарубежными патентами.

Наряду с широко известными интегральными микросхемами, составившими основу элементной базы ЕС-1033, были использованы микросхемы повышенной степени интеграции, разработка которых велась в рамках проекта ЕС-1033 параллельно с проектированием машины. Следует отметить, что в процессе проектирования определились дополнительные требования к элементам ЭВМ, реализация которых способствовала расширению возможностей их применения.

Совершенство средств вычислительной машины определяется также и степенью развития программного обеспечения. В вычислительной технике программные средства давно уже используются не только для решения задач пользователем, но и для обеспечения самого процесса функционирования ЭВМ. Этот класс программ представляет собой непосредственное развитие технических средств ЭВМ и является ее составной частью. Поэтому одновременно с созданием технических средств ЕС-1033 велись работы по развитию операционной системы (ОС ЕС), ее модели-зависимых и моделиориентированных компонентов, пакетов программ, расширяющих возможности ОС ЕС, а также программ технического обслуживания, ориентированных на модель ЕС-1033.

Машина ЕС-1033 имеет развитую систему контроля процессов обработки и пересылки информации и микропрограммную систему диагностики. В сочетании с программными средствами восстановления вычислительного процесса это обеспечивает высокую достоверность результатов обработки информации.

По совокупности оригинальных технических решений ЭВМ ЕС-1033 является самостоятельной моделью ЕС ЭВМ, не имеющей аналогов среди отечественных машин своего класса. Эта модель завершает программу разработки машин третьего поколения семейства «Ряд 1» ЕС ЭВМ. В то же время структурная организация модели и принятые в ней технические решения во многом уже имеют признаки машин четвертого поколения. По производительности и своим возможностям ЕС-1033 приближается к старшим моделям, а по объему оборудования и энергопотреблению — к младшим моделям семейства ЕС «Ряд 1».

Названные выше особенности ЕС-1033 и определили в значительной степени содержание этой книги, посвященной общим структурным принципам, построению и функционированию ЭВМ ЕС-1033. В книге рассмотрены структурные и технические особенности машины, элементная и конструктивная база, основные

структурные и функциональные схемы блоков и узлов центрального процессора и каналов ввода-вывода, машинные алгоритмы и микропрограммы операций, возможности программного обеспечения

Основное внимание уделено изложению вопросов организации структуры и работы оборудования центрального процессора, каналов ввода-вывода и системы диагностики, микропрограммирования и особенностей математического обеспечения, которые характерны именно для этой модели. Авторы стремились ввести читателя в круг тех идей и технических решений, которые явились основополагающими при создании машины. Поэтому работа оборудования центрального процессора и каналов ввода-вывода рассмотрена на самых разных уровнях — от структурных до функциональных и в ряде случаев до принципиальных схем узлов и блоков. При описании алгоритмов операций авторы придерживались того же принципа — в необходимых случаях алгоритмы рассматриваются на уровне микрокоманд. Чтобы показать аппаратную реализацию алгоритма выполнения операций, в тексте даны перекрестные ссылки. Так, при описании аппаратной части указано, в каких микропрограммах, приведенных в книге, можно найти пример работы того или иного оборудования (аппарата маскирования, анализов, байтной записи, переадресации операционных регистров и др.). В свою очередь, при описании микропрограмм отмечается, какая при этом используется аппаратура. Тем самым авторы стремились показать, для каких целей принято то или иное техническое решение, воплощенное в конструкции узлов и блоков, почему выбран тот или иной алгоритм, его отличие от традиционных решений, как и с помощью какого оборудования он реализуется.

В книге также освещены особенности программного обеспечения ЭВМ ЕС-1033, что очень важно для правильной эксплуатации машины, так как от того, насколько умело будет использовано программное обеспечение, зависит эффективность работы машины в целом. Основное внимание сосредоточено на описании модельно-зависимых компонентов операционной системы и, в частности, программных средств восстановления вычислительного процесса после обнаружения ошибок в работе центрального процессора и каналов ввода-вывода, а также тестового обеспечения и пакетов программ, расширяющих возможности ОС ЕС.

Книга рассчитана на подготовленного читателя, имеющего определенные знания в вычислительной технике и знакомого с работой и применением интегральных микросхем, с основами построения и применения цифровых вычислительных машин и систем, а также с Единой системой электронных вычислительных машин и принципами ее работы. Она адресована инженерно-техническим работникам — разработчикам технических средств вычислительной техники, микропрограммного и программного обеспечения, а также работникам вычислительных центров. Она

будет полезной для научных работников, преподавателей, студентов старших курсов и аспирантов вузов, специализирующихся в названных областях.

Книга написана группой авторов, разработчиков модели ЕС-1033 и ее программного обеспечения.

Предисловие и гл. 1 написали В. Н. Иванов, В. Ф. Гусев, А. У. Ярмухаметов, Г. П. Сорокин; гл. 2 и 3 — Г. П. Сорокин, Г. Н. Иванов, Г. М. Персов, А. У. Ярмухаметов, В. И. Коротышкин; гл. 4—9 — Г. П. Сорокин, Г. Н. Иванов, Г. М. Персов, А. У. Ярмухаметов; гл. 10 — Г. П. Сорокин; гл. 11—13 — В. В. Фадеев, Л. И. Нейман, Ю. А. Каршев, Г. П. Сорокин; гл. 14—16 — А. Х. Абдрахманов, В. И. Кельдышев, Д. А. Касимова, Т. А. Петрова.

ОБЩИЕ СВЕДЕНИЯ ОБ ЭВМ ЕС-1033 И ЕЕ СТРУКТУРНЫЕ ОСОБЕННОСТИ

1.1. СОСТАВ И ХАРАКТЕРИСТИКИ

ЭВМ ЕС-1033 (рис. 1) представляет собой дальнейшее развитие семейства «Ряд 1» программно-совместимых электронных вычислительных машин Единой системы (ЕС ЭВМ). Она использует единую систему программного обеспечения и номенклатуру технических средств ЕС ЭВМ.

Это универсальная ЭВМ, предназначенная для решения широкого круга инженерных, научно-технических, экономических, информационных задач в вычислительных центрах, системах сбора и обработки данных, информационно-поисковых системах, системах телеобработки, автоматизированных системах управления и т. п.

ЭВМ ЕС-1033 характеризуется модульностью построения, микропрограммным управлением, магистральной организацией связей узлов процессора и каналов ввода-вывода, а также наличием следующих устройств и средств:

общего канала (ОК), объединяющего мультиплексный канал (МК) и три селекторных канала (СК) ввода-вывода;

средств прямого управления, включающих средства мульти-системной работы;

автономных средств диагностики центрального процессора и каналов ввода-вывода, взаимодействующих с программными средствами восстановления операционной системы на уровне обработки ошибок центрального процессора, оперативной памяти и каналов ввода-вывода;

средств комплексирования, позволяющих строить на базе ЭВМ ЕС-1033 многомашинные вычислительные системы и подключать к ней мини-ЭВМ класса «СМ»;

стандартных средств расширения моделей Единой системы (ретрансляторами, двухканальными переключателями) и возможностями работы в системах телеобработки через мультиплексы и телепроцессоры;

специализированного процессора статистической обработки.

ЭВМ ЕС-1033 является стационарной вычислительной машиной. Она предназначена для установки в специально оборудованном помещении. Комплекты ЭВМ ЕС-1033 могут различаться емкостью оперативной памяти и составом внешних устройств.

В основной комплект ЭВМ ЕС-1033 входят (рис. 2):

| | |
|---|---|
| процессор ЕС-2433 | 1 |
| устройство запоминающее оперативное ЕС-3203 (ЕС-3207, ЕС-3208, ЕС-3263) | 1 |
| накопитель на магнитной ленте ЕС-5012-01 | 4 |

| | |
|--|---|
| устройство управления накопителями на магнитно ленте ЕС-5517 | 1 |
| накопитель на магнитных дисках ЕС-5061 (ЕС-5056М) | 3 |
| устройство управления накопителями на магнитных дисках ЕС-5561 (ЕС-5551М) | 1 |
| устройство ввода с перфокарт ЕС-6012 | 1 |
| устройство вывода с перфоленты ЕС-6022 | 1 |
| устройство вывода на перфокарты ЕС-7010 | 1 |
| устройство вывода на перфоленту ЕС-7022 | 1 |
| устройство печатающее ЕС-7032 | 1 |
| машинка пишущая с блоком управления ЕС-7077 | 1 |
| устройство подготовки данных на перфокартах ЕС-9011 | 2 |
| устройство подготовки данных на перфоленте ЕС-9024 | 1 |
| стойка питания ЕС-1033/С000 | 1 |
| щит распределительный ЕС-1030/С005 | 1 |

ЭВМ ЕС-1033 имеет следующие технические характеристики:

1. Производительность при решении научно-технических задач — 200 тыс. операций в 1 с, при решении планово-экономических задач — 150 тыс. операций в 1 с.

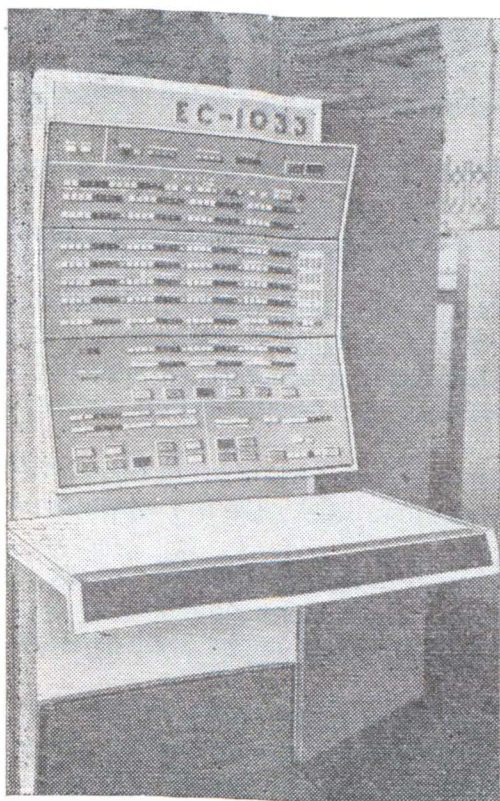


Рис. 1. Общий вид ЭВМ ЕС-1033 со стороны пульта управления

Эти показатели производительности получены путем исполнения программ, среднестатистический набор команд в которых соответственно удовлетворяет требованиям статистических смесей GIBSON и GPO — WU — II [8].

2. Емкость оперативной памяти (ОП) при комплектации модели одним устройством ЕС-3203 составляет 256К байт¹ и 512К байт при подключении двух устройств. Если используются одно или два устройства ЕС-3207 (ЕС-3208), то емкость ОП составляет соответственно 512К байт или 1024К байт. При комплектации устройством ЕС-3263 емкость ОП равна 1024К байт, а ЕС-3263.01 — 2048К байт.

3. ЭВМ имеет универсальный набор команд ЕС ЭВМ.

Она может оперировать числовыми данными, представленными в диапазоне от 10^{-78} до 10^{78} , а также логическими данными.

¹ Здесь и далее при определении емкости устройств памяти К=1024.

4. Модель имеет один мультиплексный и три селекторных канала. К мультиплексному каналу может быть подключено до 256 устройств ввода-вывода, а к каждому селекторному каналу — до 128 устройств. Каналы МК и СК сопрягаются с центральным процессором и оперативной памятью через общий канал. Нагрузочная способность интерфейсного выхода каждого канала — 10 устройств.

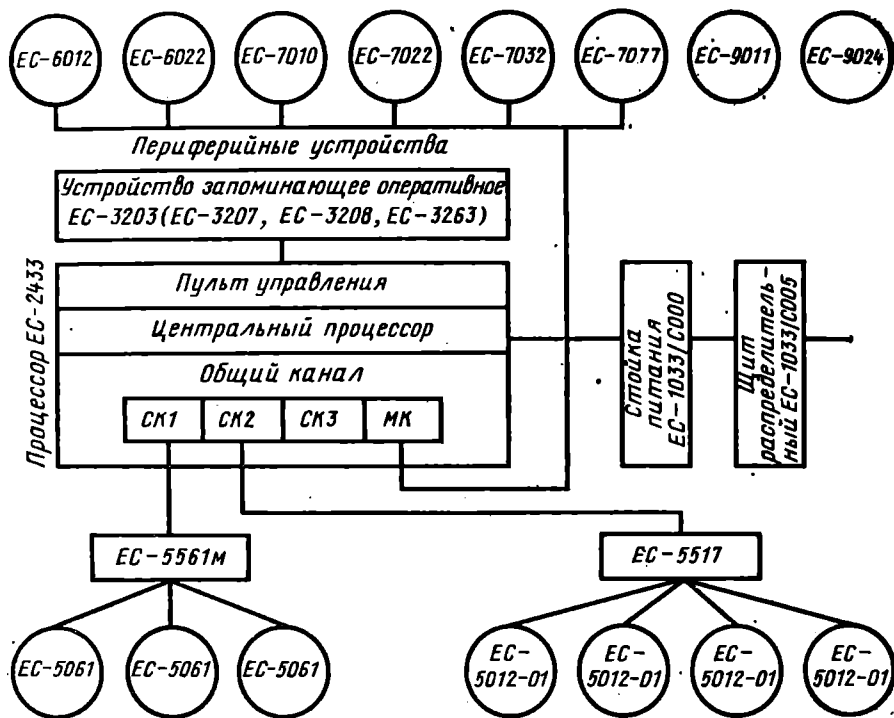


Рис. 2. Состав основного комплекта ЭВМ EC-1033

Скорость передачи данных: через мультиплексный канал в мультиплексном режиме — 50К байт в 1 с и в монопольном режиме — 300К байт в 1 с; через селекторный канал — 800К байт в 1 с.

Для максимального использования логических возможностей каналов по количеству подключаемых внешних устройств в составе ЭВМ EC-1033 предусмотрены логические ретрансляторы.

К каналам ввода-вывода могут быть подключены все типы внешних устройств EC ЭВМ.

5. В модели предусмотрены средства прямого управления с расширенными возможностями мультисистемной работы.

6. Машина имеет развитую систему контроля и диагностики, которая обеспечивает повторение команд центрального процессора при обнаружении сбоев, локализацию неисправностей центрального процессора и каналов, связь с программными средствами восстановления системы после сбоев в каналах ввода-вывода,

в центральном процессоре, в памяти ключей защиты и оперативной памяти.

7. Потребляемая мощность основного комплекта ЭВМ — 40 кВА.

8. Занимаемая площадь — 120 м².

Комплектность ЭВМ ЕС-1033 может быть расширена подключением дополнительных устройств к селекторным и мультиплексному каналам.

Использование логических ретрансляторов позволяет комплектовать машину набором внешних устройств, наиболее полно отвечающим ее конкретному назначению. Непосредственно через интерфейс ввода-вывода к ЭВМ ЕС-1033 подключаются устройства ввода-вывода, внешние магнитные запоминающие устройства, а также средства управления каналами связи. К последним относятся мультиплексоры передачи данных (МПД) и процессоры телеобработки данных (ПТД). К каналам также могут подключаться специальные устройства, такие, как процессор случайных чисел (ПСЧ) и процессор статистической обработки (ПСО).

ЭВМ ЕС-1033, в отличие от большинства моделей ЕС ЭВМ «Ряда 1», имеет дополнительные возможности по объединению в системы с использованием устройства комплексирования ВК1033.С000, позволяющего строить на его базе двух-, трех- и четырехмашинные системы. Возможная структура двухмашинной вычислительной системы представлена на рис. 3.

Устройство комплексирования ВК1033.С000 содержит шесть адаптеров канал-канал (АКК) и блок управления вычислительным комплексом (БУ ВК). Взаимное управление ЭВМ двухмашинного комплекса осуществляется с помощью мультисистемных средств центрального процессора каждой ЭВМ, связанных друг с другом через специальный интерфейс прямого управления. Управляющие сигналы, определяющие состояние и режим работы каждой ЭВМ ЕС-1033, передаются средствами прямого управления центрального процессора через интерфейс прямого управления и ВК1033.С000. Обмен информацией между ЭВМ производится через АКК.

Вычислительные системы на базе ЭВМ ЕС-1033 могут быть использованы для организации телеобработки данных. В настоящее время для Единой системы разработан и производится комплекс средств телеобработки, включающий как аппаратурные средства — мультиплексоры передачи данных, абонентские пункты (АП), аппаратуру передачи данных (АПД), так и программные [15, 29].

ЭВМ и абонентские пункты подключаются к каналам связи с помощью аппаратуры передачи данных, состав и технические данные которой приведены в прил. 1.

Сопряжение отдельных звеньев системы телеобработки осуществляется через специальные унифицированные интерфейсы, получившие название стыков. В качестве оконечных устройств в си-

стемах телеобработки могут быть использованы абонентские пункты, ориентированные на дистанционную работу в диалоговом режиме, на пакетную передачу и обработку данных, на сбор информации или на применение в управлении производством (прил. 2).

Передача данных осуществляется под управлением мультиплексоров передачи данных или процессоров телеобработки. Они

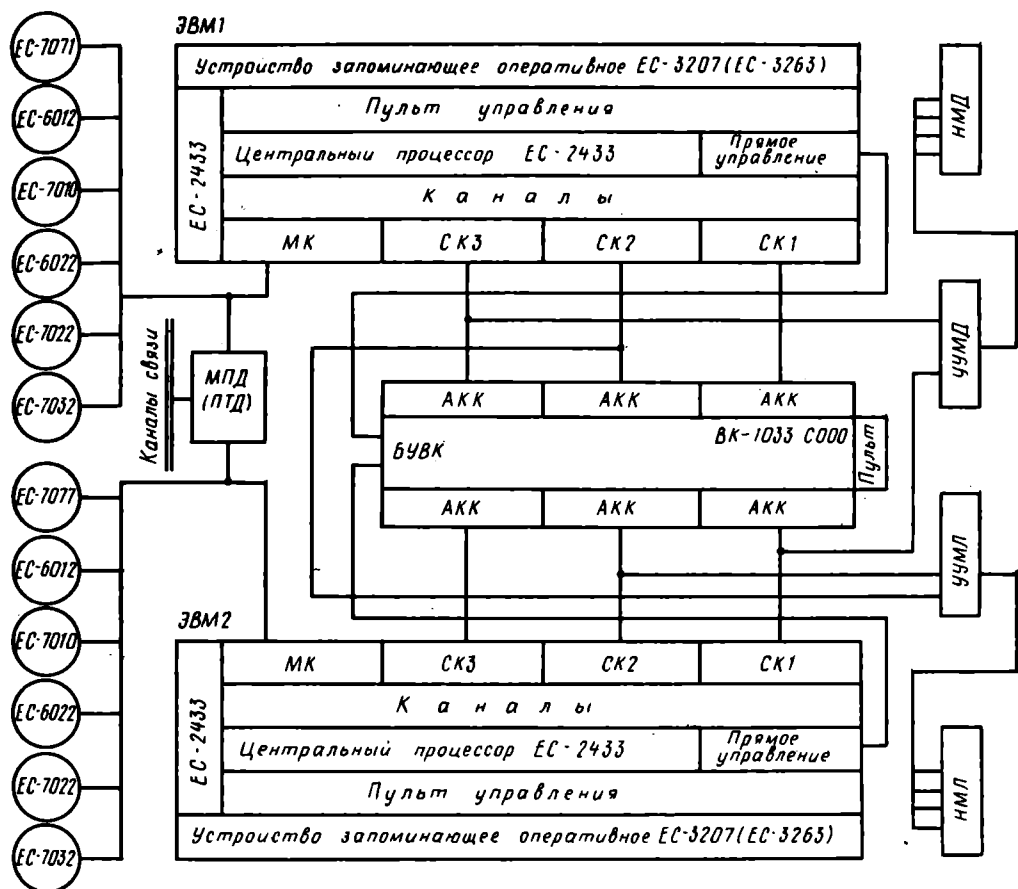


Рис. 3. Структура двухмашинной системы на базе ЭВМ EC-1033

обеспечивают сопряжение ЭВМ с АПД и управление каналами связи и АП. В ЕС ЭВМ имеются МПД с жесткой и программной конфигурацией (прил. 3), которые целесообразно использовать в простых системах телеобработки с небольшим количеством каналов. Создание сложных систем телеобработки связано с возможностями перераспределения функций управления между центральной ЭВМ и периферийными процессорами телеобработки данных.

К ЭВМ EC-1033 и строящимся на их основе многомашинным системам могут подключаться все средства телеобработки, перечисленные в прил. 1—3.

В качестве примера на рис. 4 показана структурная схема сети телеобработки с использованием ЭВМ ЕС-1033 и программируемого процессора телеобработки данных ЕС-8371.01 [36].

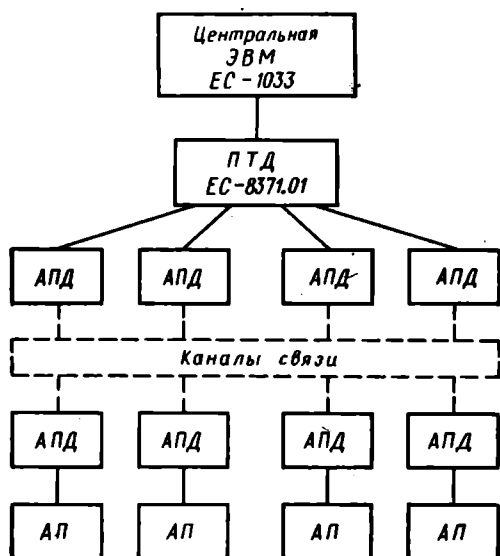


Рис. 4. Структурная схема сети телеобработки

Процессор телеобработки данных ЕС-8371.01 обеспечивает управление и обмен информацией между центральной ЭВМ и АП. Свои функции он выполняет благодаря управляющей программе, находящейся в его оперативной памяти. Программируемость ПТД позволяет значительно уменьшить нагрузку на центральную ЭВМ при управлении сетью телеобработки, что невозможно при использовании обычных МПД, а также образовывать сети различной конфигурации.

Быстродействие ПТД — более 500 000 операций в 1 с, что обеспечивает возможность совместной работы с 352 терминальными устройствами. Ско-

рость передачи данных может быть от 50 до нескольких тысяч бит в 1 с, в зависимости от типа применяемой аппаратуры передачи данных: для модема ЕС-8002 — 300, модема ЕС-8006 — (600—1200) и для модема ЕС-8013 — (1200—2400) бит в 1 с. Указанные модемы приспособлены к работе с телефонными каналами связи.

1.2. МАГИСТРАЛЬНАЯ ОРГАНИЗАЦИЯ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА И КАНАЛОВ ВВОДА-ВЫВОДА

Современная универсальная ЭВМ является сложным устройством, предназначенным для выполнения обширного набора операций, определяемых системой команд. При этом одинаково эффективно должны реализоваться алгоритмы таких существенно различных операций, как операции над операндами фиксированных и переменных форматов, операции побайтной обработки информации, операции команд переходов и команд преобразования чисел из одной системы счисления в другую и т. д. Каналы ввода-вывода должны обеспечивать как управление большим парком разнообразных внешних устройств, так и высокую пропускную способность при обмене информацией между внешними устройствами и оперативной памятью ЭВМ.

Эффективное выполнение разнообразных алгоритмов и высокое быстродействие машины может быть достигнуто использованием в универсальной ЭВМ специализированных процессоров, пред-

назначенных, например, для обработки чисел с плавающей запятой, десятичных чисел побайтной обработки и т. д. Однако при использовании интегральных схем малой и средней степени интеграции такой способ построения машин ведет к большим аппаратным затратам. Кроме того, из-за неизбежных простоев отдельных устройств оборудование такой машины используется недостаточно эффективно. С этой точки зрения более приемлемой является организация машины, основанная на использовании так называемых общих ресурсов, когда при выполнении любого алгоритма участвует большая часть оборудования, объединенная в существующую конфигурацию. При выполнении разных алгоритмов количество работающего оборудования и конфигурация соответствующим образом могут изменяться.

Цифровая вычислительная машина или ее устройства, например, центральный процессор (ЦП) или каналы ввода-вывода, построенные по типу общих ресурсов, в самом общем виде могут быть представлены в виде набора логических или функциональных модулей, соединенных информационными связями. Этими модулями, например, могут быть сумматоры, сдвигатели, регистры и т. п., предназначенные для обработки информации и хранения исходных данных, промежуточных или окончательных результатов. Управление работой каждого модуля осуществляется микропрограммным способом. Очевидно, что при прочих равных условиях наибольшей производительностью будет обладать такая система, все модули которой связаны друг с другом независимыми управляемыми связями. Тогда по любой микрокоманде можно осуществить пересылку информации из некоторого модуля в любой другой кратчайшим путем и, следовательно, за кратчайшее время. Кроме того, можно организовать ряд независимых передач внутри независимых групп модулей. Это позволяет реализовать быстродействующие и экономичные алгоритмы, ориентированные на максимальное распараллеливание микроопераций.

С увеличением количества модулей число связей быстро возрастает. Если учесть, что связь между двумя модулями фактически состоит из двух независимых связей — для чтения информации из модуля и для записи в него, то при N модулях потребуется

$$n = 2 \sum_{i=1}^{N-1} (N - i)$$

связей. Кроме того, для передачи содержимого отдельных разрядов или групп разрядов (так называемых дробных передач) требуется, как правило, применение дополнительных специальных связей либо усложнение аппаратуры. То же самое можно сказать и для случаев передач с «перекосом» и «перекрестных» передач, когда группа передаваемых бит меняет свое положение в передаваемом слове информации или две половины передаваемого слова меняют свои места.

Уменьшения количества связей между модулями можно достичь различными способами.

Один из них, широко используемый в большинстве современных машин, состоит в том, что некоторые модули не имеют непосредственной связи друг с другом. Если необходимо произвести обмен информацией между этими модулями, то ее пересылают через один или несколько других модулей, имеющих прямые связи как с первыми двумя, так и между собой. Такие холостые передачи, естественно, снижают производительность системы. Это снижение тем больше, чем больше модулей не имеют прямых связей при неизменном общем числе самих модулей.

Другой путь основывается на применении в логических модулях таких электронных схем, которые бы обеспечивали возможность производить как запись, так и чтение информации через один и тот же вход, который становится в этом случае и выходом модуля. Эта точка получает название вход-выход логического модуля. При такой организации обмена информацией количество связей сокращается в два раза.

В общем случае в машинах с микропрограммным управлением при выполнении каждой отдельной микрокоманды все многочисленные связи модулей друг с другом не используются одновременно. Используется лишь ограниченное их число. Для выполнения каждой микрокоманды образуется своя, каждый раз новая, конфигурация из ограниченного числа логических модулей, определяемая лишь теми связями, которые иницируются микрокомандой. На протяжении той или иной микропрограммы может потребоваться реализация всех связей между модулями. Таким образом, оптимальной, очевидно, является такая система, которая в принципе способна обеспечить прямую связь всех модулей друг с другом, но в каждый конкретный момент времени образует лишь только необходимое ограниченное число связей. Система, отвечающая этим требованиям, может быть реализована с помощью магистральной организации связей.

Рассмотрим группу логических и функциональных модулей, связанных несколькими (рис. 5) магистралями. Каждый из модулей своими входами-выходами подключен более чем к одной магистрали. Под воздействием сигналов управления при считывании информации из какого-либо модуля информация распространяется на всю магистраль. Этот процесс можно назвать чтением информации на магистраль или чтением на магистраль. Если одновременно с сигналом чтения (ЧТ) информации на магистраль, допустим, из первого модуля, подать на один или несколько других модулей сигналы записи (ЗП), то произойдет запись информации с магистрали в эти модули. Допустим, что в самом простейшем случае микрокоманда позволяет произвести чтение из любого одного модуля и запись в любой один модуль. Следующая микрокоманда организует связь между другой парой модулей на время одного такта работы и т. д. В целом на протяжении некоторого проме-

жутка времени такая система позволяет реализовать связь каждого модуля с каждым. В рассматриваемой системе можно в одном такте организовать связь для передачи информации, например, из первого модуля во второй через магистраль M_1 , из второго модуля в третий — через M_2 , из i -го в j -й — через M_3 . При этом для трехмагистральной организации из N модулей потребуется

$$n = 3N$$

связей модулей с магистралями. Сравним эти системы организации. Для группы из четырех модулей потребуется соответственно 12 и 12 связей, из восьми — 72 и 24, из 16 — 272 и 48, из 32 — 1048 и 96.

В системе с магистральной организацией возможна запись информации в тот же модуль, из которого производилось чтение. Например, в процессе обработки данных один из операндов извлекается из модуля 1 и подается на один из входов комбинационного блока k через магистраль M_1 , другой операнд извлекается из блока 2 и подается на второй вход блока k через магистраль M_2 , а результат обработки через магистраль M_3 записывается в модуль 1 на место первого операнда.

Чтобы обеспечить такую возможность, процессы чтения на магистраль и записи с магистрали необходимо разделить во времени. Для этого рабочий такт разбивается на два полутакта. В первом полутакте производится чтение, во втором — запись. Чтобы сохранить состояние магистрали в течение промежутка времени между чтением и записью, каждая магистраль имеет элементы памяти (ЭП). В конце рабочего такта ЭП магистрали (или просто магистраль) сбрасывается в исходное состояние специальным сигналом сброса. Таким образом, магистраль оказывается готовой принять в следующем такте по сигналам чтения новую информацию.

При чтении информации одновременно из двух модулей на одну и ту же магистраль со считанными кодами выполняется функция «логическое И», т. е. логическая обработка информации.

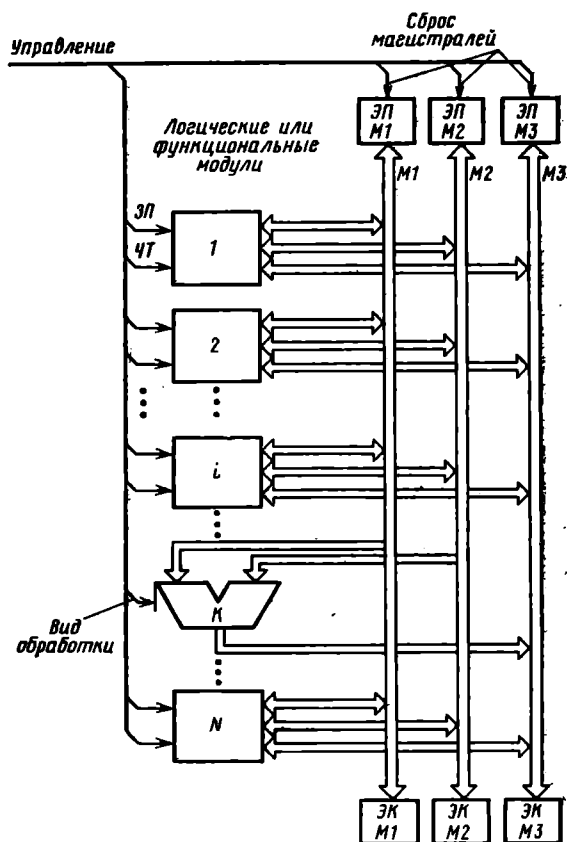


Рис. 5. Пример магистральной организации

Свойство магистрали производить логическую обработку передаваемой по ней информации используется для дробных передач. В этом случае из одного модуля считывается информация, а из другого — соответствующая выбранному формату передачи маска, содержащая единицы в тех битах, которые подлежат передаче, и нули — в остальных. Таким образом, и дробные, и полные передачи осуществляются одинаковым образом с применением тех же самых связей. При этом не требуется отдельного управления частью информационного тракта. Формат передачи программируется соответствующей маской. В общем случае маска может быть не только заданной в микропрограмме, а и быть результатом вычислений, целиком зависящим от обрабатываемых исходных данных (п. 2.4).

Таким образом, в каждом такте работы можно организовать несколько трактов передачи информации. Поскольку управление осуществляется микропрограммно, то в каждом такте создается структура связей, оптимально отвечающая выполняемому алгоритмическому действию. В следующем такте работы, при выполнении другого алгоритмического действия, создается другая структура, оптимальная уже для этого действия. Внешне это выглядит так, как будто в каждом очередном такте образуется новая конфигурация блоков. Эта конфигурация изменяется с тактовой частотой машины.

Как следует из сказанного выше, магистральная организация логических модулей в принципе требует такой их реализации, чтобы чтение и запись информации производилась через один и тот же вход-выход. А трехмагистральная организация требует, чтобы было три независимо управляемых входа-выхода, причем по каждому входу-выходу должно быть независимое управление как по записи, так и по чтению.

В ЕС-1033 принята магистральная организация центрального процессора и каналов ввода-вывода. Для ее реализации была создана специальная ячейка памяти — многофункциональный логический элемент (МФЭ), получившая свое конструктивное воплощение в микросхеме К155ХЛ1 [А. с. 486376 (СССР), а. с. 624295 (СССР)]. Эта микросхема не имеет зарубежных аналогов. Она представляет собой D-триггер с тремя информационными входами-выходами. Для записи информации в D-триггер имеются три входа для управляющих сигналов записи, каждый вход записи для «своего» входа-выхода. Аналогично для чтения информации из D-триггера имеются три входа для управляющих сигналов чтения. Микросхема К155ХЛ1 (п. 1.5) содержит два D-триггера с общими управляющими входами записи-чтения, и отдельными информационными входами-выходами. Два триггера позволяют реализовать два разряда регистра.

Магистрали следует рассматривать как устройства, имеющие определенную структуру и логику работы, а не как шины, обеспечивающие гальваническую связь между схемами. В этом заклю-

чается принципиальное отличие между «общими шинами» мини-ЭВМ и магистралями ЕС-1033. Если в первом случае шины представляют собой конструктивно систему проводов, процедура передачи информации по которым обусловлена сложной местной логикой и синхронизацией, то магистрали ЕС-1033 имеют относительно простое централизованное микропрограммное управление и довольно сложное конструктивное исполнение.

Кроме связей через магистрали для некоторых узлов и регистров процессора, а также каналов ввода-вывода необходимыми оказались и прямые связи. Их использование расширяет возможности распараллеливания микроопераций. Такие узлы могут иметь или не иметь связи с информационными магистралями.

Принцип магистральной организации ЕС-1033 не исчерпывается только наличием информационных магистралей, объединяющих функциональные и логические модули, а имеет многоуровневый, иерархический характер. В модулях имеются внутренние магистрали, объединяющие узлы и схемы модуля между собой. Обмен информацией между центральным процессором и каналами ввода-вывода также происходит через магистрали.

1.3. МИКРОПРОГРАММНОЕ УПРАВЛЕНИЕ

В ЭВМ ЕС-1033 применено микропрограммное управление (МУ), имеющее ряд преимуществ перед аппаратным — основным из них является простота и удобство проектирования. Принято считать, что МУ обеспечивает более низкое быстродействие по сравнению с аппаратным управлением вследствие расчленения выполняемого алгоритма на последовательные этапы. Это положение органически связано с построением модулей вычислительной машины и справедливо для традиционной их организации. Объединение логических модулей процессора и каналов ЕС-1033 через информационные магистрали предполагает применение МУ как неперменного компонента системы. В свою очередь, в магистральных системах МУ получило дальнейшее развитие и наиболее полно проявило свои потенциальные возможности. В частности, совместное применение так называемого кодированного управления ходом выполнения микропрограмм в сочетании с магистральной организацией позволило максимально распараллелить процесс обработки информации, значительно повысив тем самым производительность машины. Кроме того, широкие возможности такой системы позволили разработать и применить приемы микропрограммирования, специфичные именно для подобного рода систем. В отдельных частях модели ЕС-1033 — в центральном процессоре, в каналах ввода-вывода и в блоке диагностики центрального процессора — применено автономное микропрограммное управление. Если разделение управления процессора и каналов продиктовано стремлением избежать взаимных влияний и обеспечить их независимую работу, повысив тем самым

общее быстродействие машины, то автономность управления диагностикой процессора продиктовано соображениями обеспечения высокой эксплуатационной надежности.

В зависимости от степени распараллеливания процесса обработки информации и сложности выполняемых алгоритмов выбран различный формат микрокоманд. В процессоре микрокоманда имеет 128 информационных и четыре контрольных разряда, в каналах ввода-вывода — 64 информационных и два контрольных, в блоке диагностики процессора — 32 информационных разряда и один контрольный. Условно считается, что эти блоки состоят соответственно из четырех, двух и одного 32-разрядных слов; каждое слово имеет дополнительный контрольный разряд. Такое деление создает определенное удобство при индикации и описании содержимого микрокоманд, при контроле, а также на этапе проектирования. Кроме того, оно соответствует конструктивному исполнению управляющей памяти микропрограмм — постоянному запоминающему устройству (ПЗУ). Основной конструктивной единицей ПЗУ процессора, каналов ввода-вывода и блока диагностики ЦП является блок односторонней памяти БОП 2048-33 (п. 1.7). ПЗУ процессора содержит четыре таких блока: БОП0, БОП1, БОП2 и БОП3, ПЗУ каналов ввода-вывода — два блока БОП4 и БОП5, блок диагностики — один блок БОП6. Каждый из этих блоков содержит 2048 33-разрядных слов и отличается лишь информационным содержанием («прошивками»).

Микропрограммное управление в ЕС-1033 сочетается с элементами аппаратного управления. Это диктуется либо необходимостью повышения скорости передачи информации при относительно простом алгоритме управления (например, в селекторных каналах), либо стремлением ускорить выполнение операций с одновременным уменьшением объема микропрограмм. Последнее наиболее ярко проявляется в работе процессора. Ряд сложных анализов информации в нем реализован аппаратно. Выполнение такого анализа инициируется лишь одним микроприказом, например, анализ выборки (п. 4.3), анализ неуспешной передачи управления (п. 4.3), анализ циклов умножения и деления (пп. 7.3, 7.4) и целый ряд других. Используется также и такой прием, когда часть управляющей информации формируется непосредственно в процессе реализации алгоритма. Например, в микропрограмме в качестве компонента управляющей информации указывается содержимое регистра, которое, в свою очередь, зависит от обрабатываемой информации, ее расположения в ОП, длины операнда и т. п. (п. 9.4).

Высокая степень распараллеливания процесса выполнения операций подразумевает большое количество вариантов продолжения микропрограммы в зависимости от значения исходных операндов, их расположения в ОП и др., а также от промежуточных результатов обработки. Это потребовало создания развитой и гибкой системы подготовки адреса следующей микро-

команды. В ЕС-1033 использованы разнообразные приемы анализа ситуации и выбора пути продолжения микропрограммы. Многие из этих оригинальных приемов оказались достаточно эффективными [А. с. 591075 (СССР), а. с. 613401 (СССР), а. с. 615538 (СССР), а. с. 648984 (СССР)].

Еще одной характерной особенностью МУ в ЕС-1033 является использование в составе микрокоманды 32-разрядной константы — КОНСТ. Она может быть задана в любой микрокоманде. В процессе вычислений (обработки) константа подается на информационные магистрали и может служить операндом, маской, задавая фиксированные адреса ОП, служить для установки значений счетчиков, определять адреса команд. При выполнении в процессоре и каналах ввода-вывода диагностических процедур константа используется для целей тестирования трактов передачи информации, управления и проверки правильности функционирования обрабатывающих блоков.

Аппаратура МУ составляет около $\frac{1}{5}$ всего оборудования ЭВМ как по количеству ТЭЭ, так и по их номенклатуре.

1.4. СРЕДСТВА КОНТРОЛЯ И ДИАГНОСТИКИ

ЕС-1033 имеет развитые средства оперативного аппаратного контроля, которые функционируют непрерывно в процессе работы машины. Контролю подвергаются все процессы передачи информации между функциональными блоками в ЦП и каналах ввода-вывода, обработка информации в функциональных блоках, а также обмен информацией между ЦП, ОП, каналами ввода-вывода и внешними устройствами. Средства контроля позволяют обнаруживать ошибки в том же такте работы, в котором они возникают. Это определяет высокую достоверность результатов, так как предотвращает лавинообразное распространение ошибки в процессе дальнейшей работы и утрату исходных данных. Схемы контроля являются составными частями функциональных блоков ЦП и каналов ввода-вывода. Сигналы ошибок, вырабатываемые этими схемами, фиксируются в специальных триггерах и могут в дальнейшем использоваться для иницирования работы средств диагностики. Работа схем аппаратного контроля не влияет на производительность машины.

В ЕС-1033 используются контроль по mod2, контроль дублированием и контроль предсказанием значения контрольных бит по исходной информации и виду ее обработки.

По mod2 контролируются хранение информации, пересылка информации между функциональными блоками в ЦП и в каналах ввода-вывода, обмен информацией между ЦП и ОП, а также между ОП и каналами. Контроль производится как по четности, так и по нечетности.

Выбор четного и нечетного дополнения диктуется необходимостью отличить нулевую информацию от отсутствия информации. В трактах, связывающих ЦП и ОП, каналы с ОП и внешними

устройствами, а также память микропрограмм с другими функциональными блоками, отсутствие информации характеризуются нулевым значением всех разрядов, включая контрольные, а наличие нулевой информации — нулевым значением всех разрядов и единичным значением контрольного разряда. Для обеспечения возможности обнаружения случаев исчезновения информации в таких цепях используется дополнение до нечетности.

Во всех трактах связи функциональных блоков ЦП и каналов ввода-вывода со своими информационными магистралями, а также в трактах связи ЦП с каналами ввода-вывода отсутствие информации характеризуется единичным значением всех разрядов, включая контрольные. Это обусловлено тем, что сброшенная, т. е. находящаяся в исходном состоянии, магистраль содержит единицы во всех своих разрядах. Тогда наличие нулевой информации будет характеризоваться единичным значением всех информационных разрядов и нулевым значением контрольного разряда. Для контроля таких цепей используется дополнение до четности. Контрольным разрядом снабжается либо целое 32-разрядное слово, либо байт, либо тетрада информации. Контрольные биты хранятся в триггерах контрольных разрядов и пересылаются вместе с контролируемой информацией.

Дублирование используется при контроле работы арифметико-логического устройства (АЛУ) в ЦП и каналах ввода-вывода. Поскольку АЛУ может выполнять 16 арифметических и 16 логических операций, часть которых является достаточно сложной, другие способы контроля (например, предсказание контрольных бит) оказываются довольно громоздкими, требующими значительных затрат оборудования. При выполнении арифметических и логических операций операнды подаются одновременно на основное и дублирующее АЛУ, а результаты поразрядно сравниваются. В случае несовпадения значений хотя бы в одном разряде выработывается сигнал ошибки.

Контроль предсказанием значения контрольных бит применяется в таких функциональных блоках и узлах, где обработка информации влечет изменение ее четности. Это, например, счетчики, сдвигатели, узел регистра результата деления, узел параметра нормализации и др.

Магистральная организация ЦП и каналов ввода-вывода ЕС-1033 обусловила и специфику построения всей системы аппаратного контроля. При традиционной организации связей для того, чтобы проконтролировать правильность передачи информации между функциональными блоками (например, 16 регистрами общего назначения), необходимо иметь 16 схем контроля, по одной для каждого регистра. Если узел или блок предназначен только для хранения информации, достаточно контролировать лишь правильность чтения из него. При магистральной организации связей между узлами и блоками, когда передача информации производится через единые информационные магистрали, оказалось

возможным сосредоточить контрольное оборудование именно на них (элементы контроля ЭК, см. рис. 5). Независимо от количества функциональных модулей, подсоединенных к магистралям, число схем контроля равно числу магистралей. При считывании информации из какого-либо функционального блока правильность чтения на магистраль контролируется схемами контроля данной магистрали. При такой организации количество схем значительно меньше, чем при контроле каждого модуля. Однако точность обнаружения ошибки и надежность контролируемого устройства в целом практически не снижаются. Более того, надежность повышается в результате уменьшения доли контрольного оборудования.

Контрольное оборудование магистралей не позволяет проверять правильность записи информации в функциональные блоки. При временном хранении информации в блоке это не имеет значения. Она будет проверена при чтении на магистраль непосредственно перед ее использованием. Сложнее обстоит дело, когда записываемая в блок информация используется в нем самом после хранения или сразу после записи. В такие устройства (АЛУ, адресные регистры и т. п.) включены специальные внутренние схемы контроля. Следует заметить, что собственные схемы контроля должны иметь также и блоки, не связанные с информационными магистралями.

В ЦП и каналах ввода-вывода ЭВМ ЕС-1033 применяются обе разновидности контроля, т. е. как с помощью схем контроля на магистралях, так и с помощью собственных схем контроля в функциональных блоках. Итак, узлы и блоки, предназначенные только для хранения информации, контролируются при чтении информации на магистраль. Остальные функциональные блоки контролируются как при чтении, так и при записи информации. Некоторые блоки, которые не связаны с магистралями, имеют собственные схемы контроля по записи.

Диагностические средства включаются в работу по сигналам от средств контроля. Это имеет место при обнаружении ошибок и при условии, что прерывание от средств контроля разрешено операционной системой ($RCCP[13] = 1$).

Диагностические средства ЦП выделены в автономный блок, который имеет собственное микропрограммное управление (см. п. 1.3).

Диагностические средства каналов ввода-вывода входят в состав самих каналов и имеют общее с ним микропрограммное управление, т. е. диагностическим микропрограммам каналов выделена определенная область в ПЗУ каналов.

1.5. ЭЛЕМЕНТНАЯ БАЗА

Для построения ЭВМ ЕС-1033 использованы интегральные схемы (ИС) серий К131 и К155. Их применение в сочетании с новыми структурными принципами, развивающими идеи магистраль-

ной организации и микропрограммного управления, обеспечило создание машины достаточно высокой производительности при сравнительно невысоких энергопотреблении и стоимости. Поскольку имеющийся состав ИС серий К155 и К131 не мог полностью обеспечить реализацию принятых в модели ЕС-1033 технических решений, отечественной промышленностью специально для ЭВМ ЕС-1033 был разработан и освоен ряд ИС повышенной степени интеграции, в том числе и ранее упоминавшаяся ИС К155ХЛ1. По электрическим параметрам эти ИС соответствуют ИС серии К155. Конструктивно они выполнены в прямоугольных корпусах 2-го типа с числом выводов 14, 16 или 24 и с шагом выводов 2,5 мм согласно ГОСТ 17467—79 «Микросхемы интегральные. Основные размеры».

Из широко известных ИС [2] в ЭВМ ЕС-1033 применяется девять типов серии К131: К131ЛА1 (два логических элемента 4И-НЕ), К131ЛА2 (логический элемент 8И-НЕ), К131ЛА3 (четыре логических элемента 2И-НЕ), К131ЛА4 (три логических элемента 3И-НЕ), К131ЛА6 (два логических элемента 4И-НЕ с большим коэффициентом разветвления по выходу), К131ЛД1 (два 4-входных логических расширителя по ИЛИ), К131ЛР1 (два логических элемента 2И-2ИЛИ-НЕ, один расширяемый по ИЛИ), К131ЛР3 (логический элемент 2-2-2-3И-4ИЛИ-НЕ с возможностью расширения по ИЛИ) и два типа ИС серии К155: К155ЛА8 (четыре 2-входные схемы И-НЕ с открытым коллекторным выходом) и К155ЛА7 (две 4-входные схемы И-НЕ с открытым коллекторным выходом и повышенной нагрузочной способностью).

Из вновь освоенных отечественной промышленностью в рамках проекта ЕС-1033 в машине применены следующие ИС повышенной степени интеграции (рис. 6).

К155ИДЗ — дешифратор-демультиплексор на четыре информационных входа и 16 выходов. При наличии на обоих входах управления W0 и W1 сигналов низкого уровня на одном из выходов присутствует сигнал низкого уровня. Номер этого выхода определяется 4-разрядной кодовой комбинацией на входах 1, 2, 4, 8. При комбинации сигналов 01, 10 или 11 на входах W0 и W1 на всех выходах дешифратора будет высокий уровень. В машине эта ИС применяется как дешифратор 4-разрядных кодов (например, дешифратор полей микрокоманд).

К155ИПЗ — арифметико-логическое устройство. Оно обеспечивает выполнение 16 арифметических и 16 логических операций (табл. 1) над 4-разрядными операндами. Операнды подаются на входы А0 — А3 и В0 — В3. Код выполняемой операции поступает на входы S0 — S3 выбора функций, вид операции (арифметическая или логическая) задается сигналом на входе М (вход задания режима работы). Входной перенос поступает на вход С0.

Выходы F0 — F3 — выходы сигналов результата, Р — выход сигнала, соответствующего функции распространения переноса, G — выход сигнала, соответствующего функции формирования

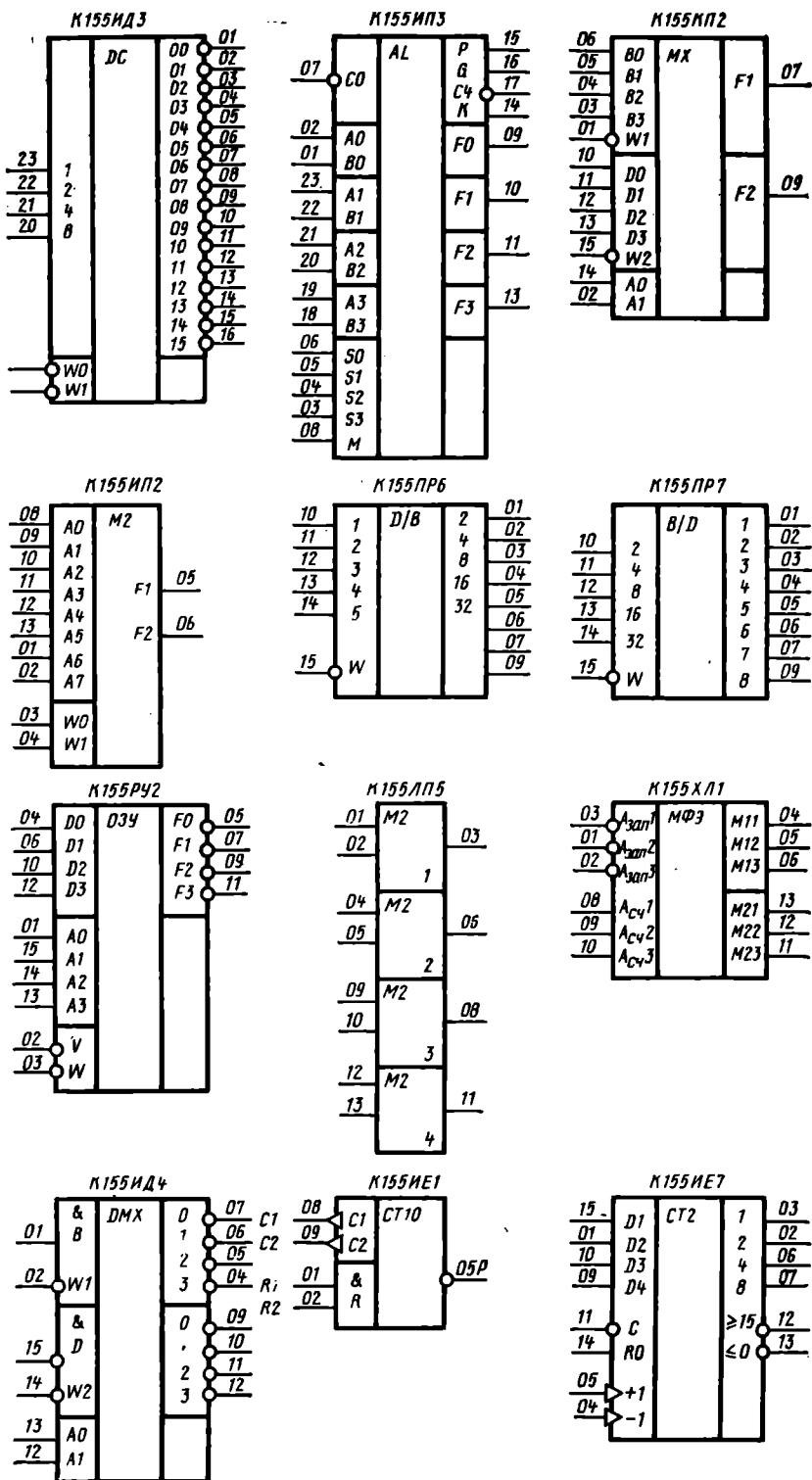


Рис. 6. ИС повышенной степени интеграции

Таблица функций, выполняемых АЛУ

| Код функции | Положительная логика | | Отрицательная логика | |
|-------------|-----------------------------|----------------------------------|------------------------------|--------------------------------------|
| | М-1 (логические функции) | М-0 (арифметические функции) | М-1 (логические функции) | М-0 (арифметические функции) |
| 0000 | \bar{a}_i | $A + \rho_0$ | \bar{a}_i | $A + \rho_0$ |
| 0001 | $\overline{a_i \vee b_i}$ | $(A \vee B) + \rho_0$ | $\overline{a_i b_i}$ | $AB + \rho_0$ |
| 0010 | $\bar{a}_i b_i$ | $(A \vee \bar{B}) + \rho_0$ | $\bar{a}_i \vee b_i$ | $A\bar{B} + \rho_0$ |
| 0011 | Логический нуль | Логическая единица $+ \rho_0$ | Логический нуль \downarrow | Логическая единица $+ \rho_0$ |
| 0100 | $\overline{a_i b_i}$ | $A + A\bar{B} + \rho_0$ | $\overline{a_i \vee b_i}$ | $A + (A \vee \bar{B}) + 1 + \rho_0$ |
| 0101 | b_i | $(A \vee B) + A\bar{B} + \rho_0$ | b_i | $AB + (A \vee \bar{B}) + 1 + \rho_0$ |
| 0110 | $a_i \oplus b_i$ | $A - B - 1 + \rho_0$ | $\overline{a_i \oplus b_i}$ | $A - B + \rho_0$ |
| 0111 | $a_i \bar{b}_i$ | $A\bar{B} - 1 + \rho_0$ | $a_i \vee b_i$ | $A \vee \bar{B} + 1 + \rho_0$ |
| 1000 | $\bar{a}_i \vee b_i$ | $A + AB + \rho_0$ | $\bar{a}_i b_i$ | $A + (A \vee B) + 1 + \rho_0$ |
| 1001 | $\overline{a_i \oplus b_i}$ | $A + B + \rho_0$ | $a_i \oplus b_i$ | $A + B + 1 + \rho_0$ |
| 1010 | b_i | $(A \vee \bar{B}) + AB + \rho_0$ | b_i | $A\bar{B} + (A \vee B) + 1 + \rho_0$ |
| 1011 | $a_i b_i$ | $AB - 1 + \rho_0$ | $a_i \vee b_i$ | $A \vee B + 1 + \rho_0$ |
| 1100 | Логическая единица | $A + A^* + \rho_0$ | Логическая единица \ominus | $A + A^* + 1 + \rho_0$ |
| 1101 | $a_i \vee b_i$ | $(A \vee B) + A + \rho_0$ | $a_i \bar{b}_i$ | $AB + A + 1 + \rho_0$ |
| 1110 | $\bar{a}_i \vee b_i$ | $(A \vee \bar{B}) + A + \rho_0$ | $a_i b_i$ | $A\bar{B} + A + 1 + \rho_0$ |
| 1111 | a_i | $A - 1 + \rho_0$ | a_i | $A + 1 + \rho_0$ |

Примечание. $A = a_0 a_1 a_2 a_3$; $B = b_0 b_1 b_2 b_3$; * — сдвиг на один разряд влево. При положительной логике: $\rho_0 = 0$, если $\bar{C}0 = 1$ и $\rho_0 = 1$, если $\bar{C}0 = 0$. При отрицательной логике: $\rho_0 = -1$, если $C0 = 0$, и $\rho_0 = 0$, если $C0 = 1$.

переноса, С4 — выход сигнала переноса, К — выход сигнала результата сравнения ($A = B$). Эта ИС применяется при построении регистров-счетчиков в арифметико-логическом блоке (БАЛ) процессора, осуществляющего счет на +2 и на +4 (последние четыре разряда регистра адреса процессора и адресной части регистра слова состояния программы), в схемах модификации адреса в каналах ввода-вывода.

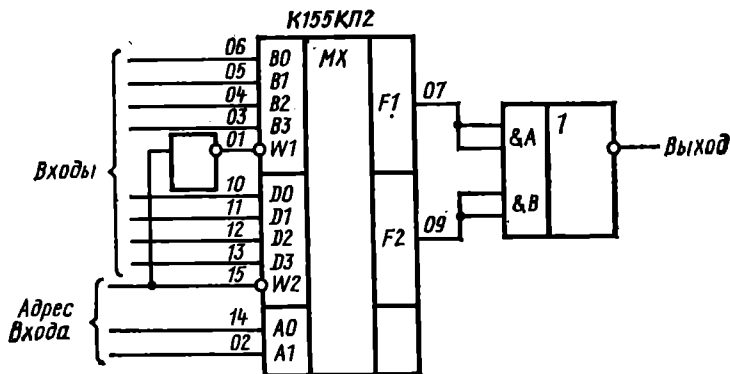


Рис. 7. Селектор-мультиплексор 8-1

К155КП2 — двойной цифровой селектор-мультиплексор 4-1, работающий в соответствии с табл. 2.

Таблица 2

Таблица работы ИС К155КП2

| Адресные входы | | Информационные входы | | | | Управляющие входы | Выходы |
|----------------|----|----------------------|---------|---------|---------|-------------------|---------|
| A0 | A1 | B0 (D0) | B1 (D1) | B2 (D2) | B3 (D3) | W1 (W2) | F1 (F2) |
| X | X | X | X | X | X | 1 | 0 |
| 0 | 0 | 0 | X | X | X | 0 | 0 |
| 0 | 0 | 1 | X | X | X | 0 | 1 |
| 0 | 1 | X | 0 | X | X | 0 | 0 |
| 0 | 1 | X | 1 | X | X | 0 | 1 |
| 1 | 0 | X | X | 0 | X | 0 | 0 |
| 1 | 0 | X | X | 1 | X | 0 | 1 |
| 1 | 1 | X | X | X | 0 | 0 | 0 |
| 1 | 1 | X | X | X | 1 | 0 | 1 |

Примечание: X — безразличное состояние.

В машине эта ИС применяется как двойной коммутатор сигналов четырех направлений на одно, а также (рис. 7) как коммутатор восьми направлений на одно.

К155ИП2 — 8-разрядная схема контроля четности и нечетности, работающая в соответствии с табл. 3.

Таблица 3

Таблица работы ИС К155ИП2

| Входы | | Выходы | | | |
|-----------------------------------|--------|--------|--------|--------|--|
| 08, 09, 10, 11, 12, 13, 01, 02 | 03 | 04 | 05 | 06 | |
| Сумма единиц четная | 1 0 | 0 1 | 1 0 | 0 1 | |
| Сумма единиц нечетная | 1 0 | 0 1 | 0 1 | 1 0 | |
| Сумма единиц любая | 1 0 | 1 0 | 0 1 | 0 1 | |

Таблица 4

Таблица работы ИС К155РУ2

| Разрешение чтения V | Разрешение записи W | Операция | Состояние F0—F3 |
|---------------------|---------------------|----------|--|
| 0 | 0 | Запись | Инверсный код числа поданного на входы D0—D3 |
| 0 | 1 | Чтение | Инверсный код выбранного числа |
| 1 | 0 | Запрет | Инверсный код числа, поданного на входы: D0—D3 |
| 1 | 1 | — | Высокий потенциал |

В машине эта ИС применяется в схемах формирования значения контрольного разряда для байта информации.

К155ИП6 — преобразователь двоично-десятичного кода в двоичный. Интегральная схема имеет выходы с открытым коллектором. Она допускает каскадирование и применяется в блоке конвертирования.

К155ИП7 — преобразователь двоичного кода в двоично-десятичный. Эта ИС также имеет выходы с открытым коллектором и допускает каскадирование. Она применяется в блоке конвертирования.

К155РУ2 — ОЗУ на 64 бита с произвольной выборкой (16×4). D0 — D3 — информационные входы, F0 — F3 — информационные выходы. Код адреса записи (или чтения) поступает на адресные входы A0 — A3, V — вход разрешения чтения, W — вход разрешения записи. ИС работает в соответствии с табл. 4.

В машине ИС этого типа применяются для построения регистровой памяти: регистров общего назначения и памяти ключей защиты — в ЦП, памяти мультиплексного канала — в каналах ввода-вывода.

К155ЛП5 — четыре 2-входовых логических элемента «исключающее ИЛИ». Эта ИС применяется для формирования значения контрольного разряда тетрады информации, а также для поразрядного сравнения.

К155ИД4 — сдвоенный дешифратор-мультиплексор 2-4, работающий в соответствии с табл. 5.

Эта ИС применяется как сдвоенный дешифратор (рис. 8), работающий от общих адресных входов A0 и A1 с отдельным стро-

бированием или как два коммутатора сигналов со входа (входы В и D) на четыре направления. Номер выхода, на который коммутируется входной сигнал, определяется кодом на адресных входах. Каждый коммутатор строится отдельно. ИС работает как дешифратор 3-8 или коммутатор 1-8.

K155IE1 — декадный счетчик (см. рис. 6) с фазоимпульсным представлением информации. Используется как делитель частоты с коэффициентом деления 10. Если на вход С1 (С2) подавать последовательность импульсов с частотой f , то на выходе вырабатывается последовательность импульсов с частотой $f/10$. Входы R1 и R2 — входы сброса. Применяется в задающем генераторе блока синхронизации машины.

Таблица 5

Таблица работы ИС K155ИД4

| Адресные входы | | Информационные входы | | Управляющие входы | | Выходы | | | |
|----------------|----|----------------------|---|-------------------|----|--------|---|---|---|
| A0 | A1 | B | D | W1 | W2 | 0 | 1 | 2 | 3 |
| X | X | X | X | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| X | X | 0 | 1 | X | X | 1 | 1 | 1 | 1 |

Примечание: X — безразличное состояние.

довательность импульсов с частотой f , то на выходе вырабатывается последовательность импульсов с частотой $f/10$. Входы R1 и R2 — входы сброса. Применяется в задающем генераторе блока синхронизации машины.

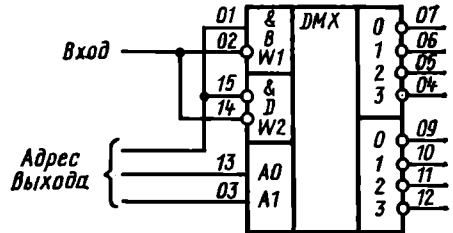


Рис. 8. Дешифратор 3-8, демультиплексор 1-8

K155IE7 — 4-разрядный двоичный реверсивный счетчик; R0 — вход сигнала сброса; С — вход записи константы, подаваемой на входы D1 — D4. Входы +1 и -1 служат для соответствующей модификации содержимого счетчика. Выходы 1—4 — выходы содержимого счетчика, ≥ 15 — выход переноса, ≤ 0 — выход заема. Два последних выхода могут служить для расширения разрядности счетчика. ИС K155IE7 применяется для построения адресных регистров, регистров общего назначения и регистров операционных, а также в блоках диагностики ЦП и каналов ввода-вывода.

K155XЛ1 — многофункциональный элемент, содержащий два триггера (рис. 9), объединенных общими цепями управления записью и считыванием информации. Сигналы на входах $A_{зап}1$, $A_{зап}2$, $A_{зап}3$ и $A_{сч}1$, $A_{сч}2$, $A_{сч}3$ определяют соответственно адрес входа-выхода (M11, M12, M13 и M21, M22, M23), с которого производится запись информации в триггер или на который происходит считывание информации из триггера. Например, при подаче сигнала низкого уровня на вход $A_{зап}1$ происходит запись информации с входа-выхода M11 и M21 в оба триггера. При подаче сигнала

высокого уровня на вход $A_{сч}2$ происходит считывание информации из обоих триггеров на входы-выходы $M12$ и $M22$.

В ЕС-1033 эта ИС применяется как ячейка магистральной структуры для построения узлов и блоков, соединенных друг с другом через информационные или внутренние магистрали. Одна ИС $K155ХЛ1$ реализует два разряда регистра. Возможно производить запись в оба разряда с двух или трех направлений одновременно с осуществлением логической функции ИЛИ. Например, при подаче на входы $A_{зап}1$ и $A_{зап}2$ сигналов записи при

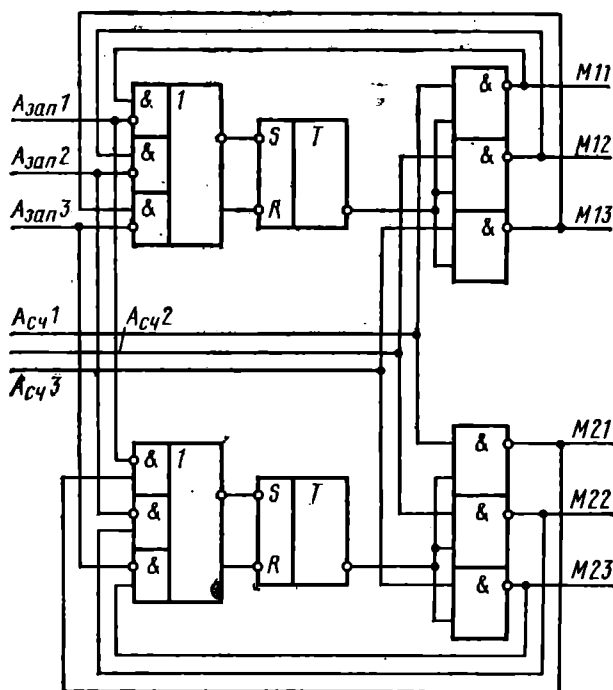


Рис. 9. Функциональная схема ИС $K155ХЛ1$

считывания более чем из одного регистра реализуется логическая функция И.

Для реализации магистральной структуры входы-выходы ИС $K155ХЛ1$ могут быть объединены также с выходами других ИС этой серии, имеющих выходы с открытым коллектором.

При разработке большинства ИС повышенной степени интеграции в составе серии $K155$ был учтен накопленный отечественной и зарубежной промышленностью опыт создания и применения аналогичных по функциональному содержанию схем для других систем элементов. Например, ряд перечисленных ИС имеют аналоги в составе комплекта схем серии $SN74$ фирмы «Texas Instruments», США [2]. Возможности разработанного комплекта существенно расширили включение в его состав оригинальных схем. Среди них особое место занимает ИС $K155ХЛ1$, которая оказалась эффективной при реализации магистральных структур технических средств. Использование этих ИС в модели ЕС-1033, кроме

наличия на входах-выходах $M11$ и $M12$ единичной и нулевой информации соответственно в первый разряд будет записана единичная информация. Возможно также считывание информации на два или три направления одновременно.

Входы-выходы ИС $K155ХЛ1$ в режиме считывания имеют открытоколлекторные вентили. Поэтому для целей реализации магистральной структуры соответствующие входы-выходы могут объединяться проводным монтажом. При таком объединении в режиме одновременного

уменьшения объема аппаратуры позволило повысить быстродействие машины за счет сокращения длин связей и, следовательно, уменьшения задержек как в цепях распространения сигналов, так и на самих ИС.

Кроме схем с открытым коллекторным выходом сокращению аппаратурных затрат способствовало применение соединений выходов ИС на основе использования так называемого третьего состояния ИС.

Выходы ИС серии К131, которые не имеют открытых коллекторов, при обычном включении ИС проводным монтажом объединены быть не могут. Рассмотрим базовую схему ТТЛ (рис. 10) и объединение выходов таких ИС (рис. 11). Если все ИС находятся в состоянии логической единицы, то такое объединение, вообще говоря, возможно: все $R_{VT_{\text{бвн}}} \ll 1$,

$R_{VT_{\text{бвн}}} \gg 1$, режим работы выходных транзисторов не изменяется и на объединенном выходе состояние логической единицы сохраняется. Если же одна из ИС (допустим, первая) переключится в состояние логического нуля, то $R_{VT_{\text{бвн}}} \gg 1$, $R_{VT_{\text{бвн}}} \ll 1$. Резистор R_6 имеет величину порядка 40 Ом, внутреннее сопротивление открытого транзистора $R_{\text{бвн}} \approx 10$ Ом. Даже в случае

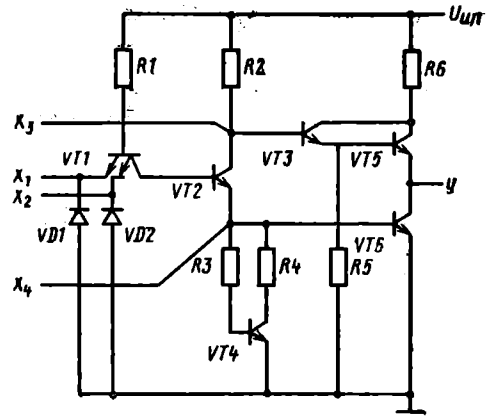
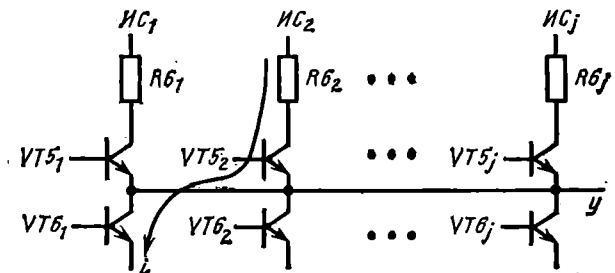


Рис. 10. Базовая схема

Рис. 11. Объединение выходов ИС



объединения двух ИС (допустим 1 и 2) в цепи $R_{6_2} - R_{VT_{\text{бвн}}} - R_{VT_{\text{бвн}}}$ возникнет недопустимый ток, примерно равный 0,08 А.

В рассматриваемых микросхемах предельный ток открытого транзистора должен быть не более 0,02 А. Следовательно, в данном случае объединение выходов схем невозможно.

Чтобы иметь возможность объединить выходы ИС для организации чтения на магистраль, в ЭВМ ЕС-1033 используется «третье состояние» ИС. Если на входы X_3 и X_4 ИС (см. рис. 10) через диоды подать уровень логического нуля ($\sim 0,4$ В), то на базах обоих выходных транзисторов VT_5 и VT_6 будет низкий потенциал

и оба транзистора закрываются. На выходе ИС не будет ни логического нуля, ни логической единицы, поскольку выход будет изолирован от обоих полюсов источника высоким ($\sim 10^6$ Ом) сопротивлением закрытых транзисторов. Интегральная схема будет находиться в третьем состоянии, которое входами других микросхем воспринимается как логическая единица.

Выходы ИС, находящихся в третьем состоянии, могут быть объединены проводным монтажом — магистралью. Для считыва-

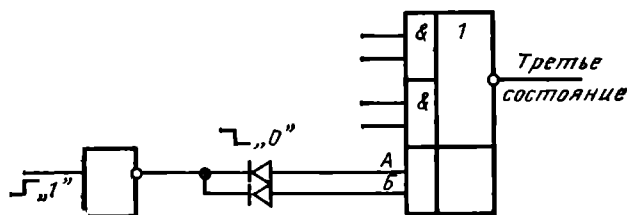


Рис. 12. Включение третьего состояния ИС

ния состояния какой-либо ИС на магистраль эта ИС сигналом чтения выводится из третьего состояния и вся магистраль переходит в состояние логического нуля или единицы в зависимости от сигналов на входе ИС.

Практически третье состояние может быть реализовано только для ИС, допускающих расширение по ИЛИ (рис. 12), и применяется в основном на внутренних магистралях функциональных узлов и блоков процессора и каналов ввода-вывода, а также на магистрали анализов.

1.6. СИСТЕМА СИНХРОНИЗАЦИИ

В ЕС-1033 используется синхронный принцип работы, при котором информация и управляющие сигналы имеют строго определенное размещение во времени. Для организации синхронной работы формируется ряд синхросигналов, следующих с единой тактовой частотой генератора, равной 2,5 МГц.

Генератор формирует пять основных синхросерий: С1, С2, С1', С2' и СБРМ — сброс магистралей (рис. 13). Синхросерии С1 и С2, а также С1' и С2' сдвинуты относительно друг друга на половину такта. В машине принято, что любое чтение информации на информационные магистрали производится по синхросерии С1, а все записи с магистралей — в конце такта по синхросерии С2. Промежуток между С1 и С2 отведен для работы комбинационных обрабатывающих схем. Обмен между теми регистрами, которые имеют прямые немагистральные связи, а также выполнение других микроприказов может происходить либо по синхросерии С1, либо С2. Синхросерии С1' и С2' используются для обеспечения контроля информации при записи и чтении.

Для работы комбинационных схем, а также для разделения во времени процессов чтения информации на магистраль и записи с нее магистраль имеет элементы памяти. Информация, считанная

на магистраль по серии С1, хранится на магистрали до конца импульса серии С2. В конце такта, после окончания импульса серии С2, магистраль сбрасывается в исходное состояние по импульсу синхросерии СБРМ. Обычно сброс магистрали происходит каждый такт, однако имеется возможность заблокировать сброс магистралей на необходимое число тактов.

Генератор синхросерий (рис. 14) состоит из задающего генератора ЗГ, коммутатора КМ, элементов задержки и формирователей Φ синхросерий. Задающий генератор вырабатывает последовательность импульсов ЗГ (см. рис. 13) в периодом 200 нс и

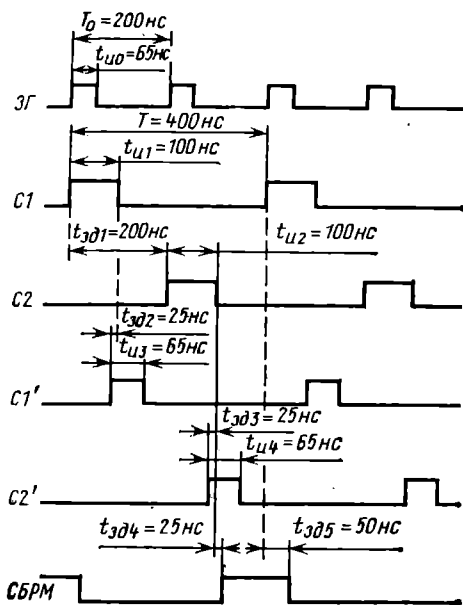


Рис. 13. Временная диаграмма следования основных синхросерий.

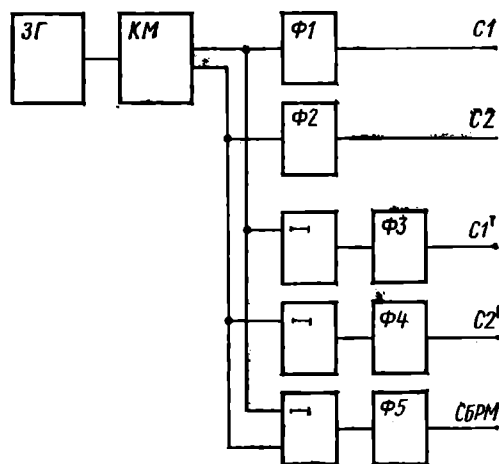


Рис. 14. Функциональная схема генератора

длительностью примерно 65 нс. Каждый из импульсов поочередно коммутируется на два направления, образуя две последовательности импульсов, каждая из которых следует с периодом 400 нс. Обе последовательности сдвинуты относительно друг друга на полпериода и используются для формирования основных синхросерий с помощью формирователей и элементов задержки.

Блок синхронизации (рис. 15), содержащий генератор Г синхросерий и усилитель мощности УМЩ, расположен в раме С (п. 1.8) стойки ЕС-2433. С выхода УМЩ основные синхросерии по кабелям равной электрической длины поступают к центральным усилителям рам А и В, откуда они разводятся по периферийным усилителям и далее к потребителям. Для потребителей рамы С УМЩ вырабатывает задержанные синхросерии С13, С23, С1'3, С2'3 и СБРМ3 с задержкой, равной задержке в названных выше кабелях. Нагрузочная способность задержанного выхода УМЩ достаточна для всех потребителей и периферийных усилителей рамы С.

В каналы ввода-вывода, расположенные в рамах А и С, поступают только синхросерии С1 и С2 (С13 и С23). В процессор, размещенный в раме В, и блок диагностики, расположенный в раме С, поступают все пять основных синхросерий. В раме В

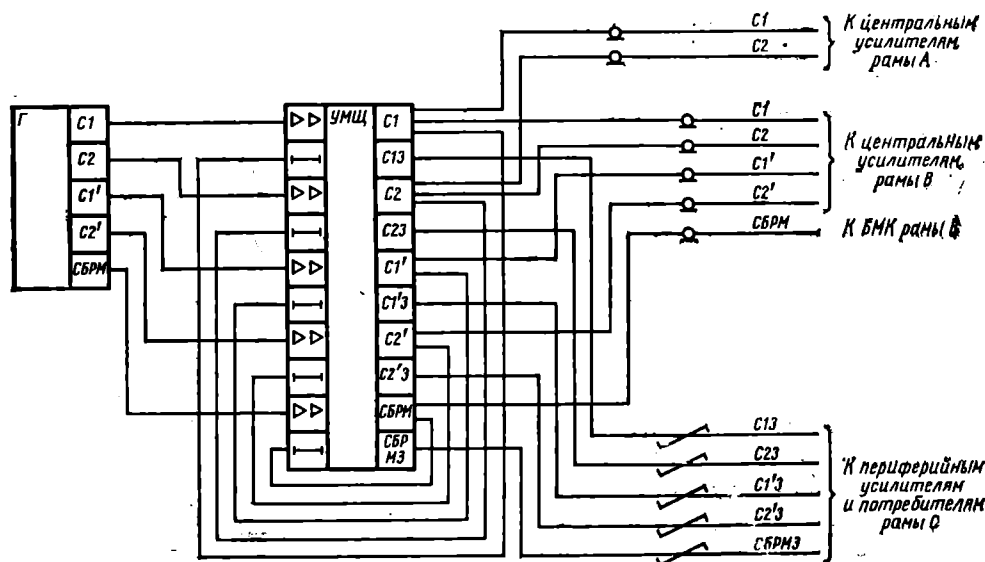


Рис. 15. Функциональная схема блока синхронизации

синхросерия СБРМ поступает непосредственно в блок магистрального коммутатора БМК.

В дальнейшем изложении материала для упрощения на временных диаграммах будем полагать, что время задержек $t_{эд_2}$, $t_{эд_3}$, $t_{эд_4}$ и $t_{эд_5}$ равно нулю (см. рис. 13).

1.7. ПОСТОЯННАЯ ПАМЯТЬ МИКРОПРОГРАММ

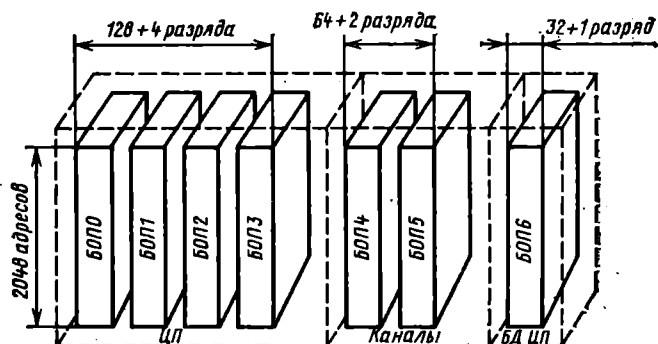
В ЭВМ ЕС-1033 для хранения микропрограмм ЦП, каналов ввода-вывода и блока диагностики ЦП используются самостоятельные постоянные запоминающие устройства (ПЗУ). Каждое из этих ПЗУ состоит из унифицированных блоков односторонней памяти — БОП 2048-33. ПЗУ ЦП состоит из четырех блоков — БОП0, БОП1, БОП2 и БОП3; ПЗУ каналов ввода-вывода состоит из двух блоков — БОП4 и БОП5; ПЗУ блока диагностики ЦП — из одного блока БОП6 (рис. 16). Это соответствует формату микрокоманды, который для ЦП составляет 128 информационных разрядов, в каналах ввода-вывода — 64 и в блоке диагностики ЦП — 32 разряда.

БОП 2048-33 представляет собой постоянное запоминающее устройство трансформаторного типа с произвольной выборкой. Емкость памяти 2048 слов 33-разрядного формата (один разряд контрольный), период обращения не менее 370 нс, время выборки не более 150 нс. Напряжение питания устройства +27 и +5 В, потребляемый ток равен соответственно 0,5 и 2 А.

Для хранения информации используются П-образные ферритовые сердечники типа М1500 с прямоугольной петлей гистерезиса. Запоминающая часть ПЗУ построена по принципу «сердечник — разряд», т. е. в одном сердечнике хранится один разряд всех слов, количество сердечников равно количеству разрядов, а информация записывается адресными проводами. Если адресный провод проходит через сердечник, то в данном разряде запоминаемого по этому адресу числа содержится единица, если огибает, то — ноль.

Конструктивно БОП 2048-33 имеет четыре линейки Л0 — Л3 разрядных сердечников 1 (рис. 17). Все 2048 адресных проводов разделены на 64 жгута по 32 провода в каждом. В одну линейку,

Рис. 16. Память микропрограмм



содержащую 33 разрядных сердечника, укладывается 16 жгутов. Обмотки считывания 3 всех сердечников одноименных разрядов подключены к усилителям У0 — У32. Каждый из адресных проводов включен между источником напряжения +27 В и общей шиной через верхний ключ (ВК), развязывающий диод VD и нижний ключ (НК). Каждый из верхних ключей ВК0 — ВК31 подключен к 32 адресным проводам одноименных разрядов всех жгутов. Каждый из нижних ключей НК0 — НК63 подключен ко всем проводам одного из 64 жгутов. Диоды VD0 — VD 2047 включены в каждый адресный провод. Таким образом, нижние ключи определяют жгут, а верхние — адресный провод в жгуте. Пара верхних и нижних ключей определяет один из 2048 адресных проводов.

Код адреса поступает по адресным шинам в дешифраторы нижних (ДШ НК) и дешифраторы верхних (ДШ ВК) ключей. Дешифраторы стробируются сигналом чтения блока односторонней памяти ЧТ БОП (рис. 18), и сигналы с соответствующих выходов дешифраторов открывают пару ВК и НК. При этом по выбранному адресному проводу протекает импульс тока, наводящий ЭДС в обмотках считывания тех разрядов, где содержится единица. Считанная информация заносится в регистр информации (РИ) по сигналу СТР РИ. Информация в РИ хранится в течение машинного такта до следующего сигнала СТР РИ.

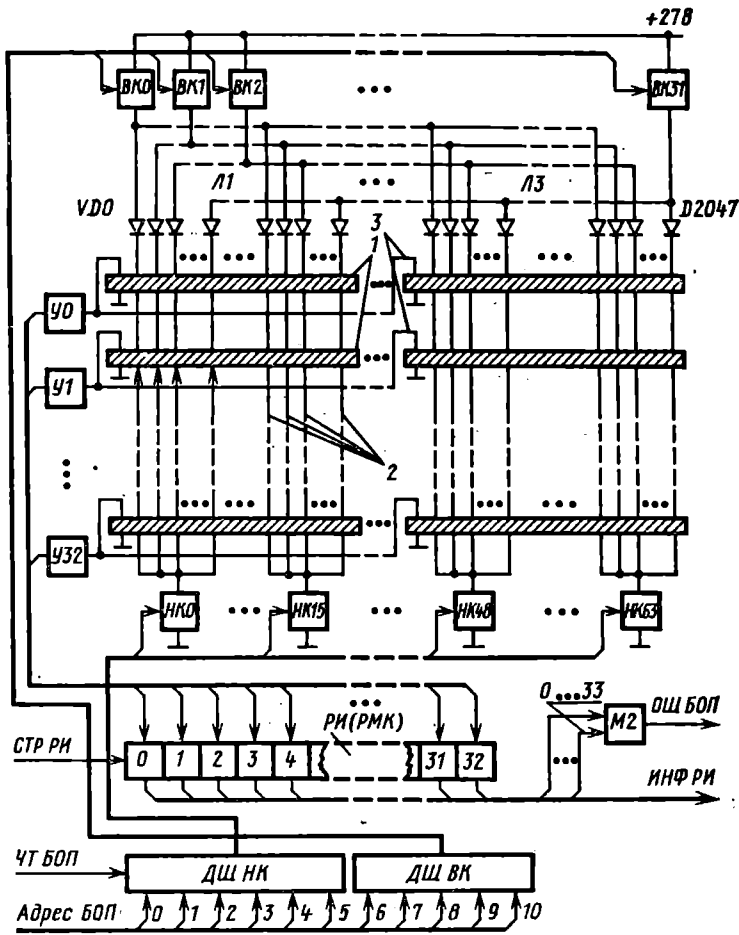


Рис. 17. Структурная схема БОП 2048-33:

1 — сердечники; 2 — обмотки считывания; 3 — адресные провода одного жгута

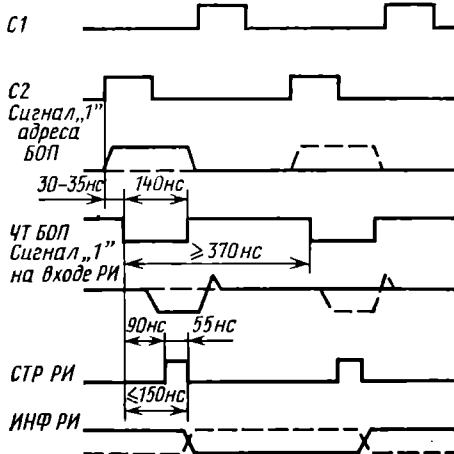


Рис. 18. Временная диаграмма работы БОП

Контроль правильности считанной информации осуществляется по $\text{mod}2$ дополнением до нечетности. Если сумма единиц информации будет четной, вырабатывается сигнал ошибки ОШ БОП.

Во время выполнения текущей микрокоманды формируется адрес следующей (адрес БОП). Его формирование, с учетом работы схем анализа, должно закончиться до конца текущего такта к началу синхроимпульса $C1$ следующего такта (рис. 18). Сигнал СТР РИ подается в конце сигнала ЧТ примерно за 25—30 нс до начала $C1$. Информация в РИ меняется перед началом $C1$. Таким образом, к импульсу $C1$ следующего такта коды микроприказов, выполняемых по $C1$, уже поступают на соответствующие схемы.

БОП 2048-33 является конструктивно законченным устройством. РИ блоков БОП0, БОП1, БОП2 и БОП3 являются регистрами микрокоманд (РМК0, РМК1, РМК2 и РМК3) процессора, РИ БОП4 и БОП5 — регистрами микрокоманд (РМК4 и РМК5) каналов ввода-вывода, а РИ БОП6 является регистром микрокоманд (РМК6) блока диагностики процессора.

1.8. КОНСТРУКЦИЯ МАШИНЫ И СИСТЕМА ЭЛЕКТРОПИТАНИЯ

ЭВМ ЕС-1033 выполнена на основе унифицированной конструктивной базы ЕС ЭВМ (рис. 19). В машине можно выделить пять уровней конструктивных модулей:

1 — интегральные схемы ИС, выполняющие универсальные логические функции И-НЕ, ИЛИ-НЕ и их сочетания, или интегральные схемы повышенной степени интеграции, реализующие какой-либо электронный узел — счетчик, дешифратор и т. д. или его часть;

2 — типовые элементы замены (ТЭЗ), содержащие ИС и реализующие отдельные функциональные узлы или их части;

3 — панели, состоящие из 40 ТЭЗ каждая. В панели реализованы один или несколько функциональных узлов устройства. Например, панель $1E - 1H$ рамы B ЭВМ ЕС-1033 включает часть блока магистрального коммутатора, часть арифметико-логического устройства, счетчики и т. д.;

4 — рамы, имеющие несколько панелей. Каждая рама реализует одно или несколько функциональных устройств. Например, рама A содержит общий, мультиплексный и два селекторных канала, рама B — центральный процессор ЕС-1033, рама C — блок диагностики и один селекторный канал;

5 — стойки (шкафы), содержащие какие-либо отдельные устройства, например, стойка питания ЕС-1033/С000, стойка процессора и каналов ЕС-2433.

Типовой элемент замены — модуль второго уровня. Он представляет собой двустороннюю печатную плату с размерами $140 \times 150 \times 1,5$ мм, на которой обычно устанавливают 24 ИС. Плата ТЭЗ имеет печатный разъем с 52 контактами. Цепи питания

ИС напряжением 5 В разводятся навесными металлическими шинами, установленными на плате. На шинах питания для уменьшения помех монтируются высокочастотные конденсаторы. Электрические соединения между контактами разъема и ИС внутри ТЭЗ осуществляются с помощью печатных проводников. На плате ТЭЗ устанавливаются также резисторы, диоды, транзисторы и другие дискретные элементы.

Конструктивный адрес ТЭЗ содержит номер рамы, полупанели и свой порядковый номер в последней. Например, *B-1A04* — рама *B* (ЦП), полупанель *1A*, порядковый номер ТЭЗ *04*.

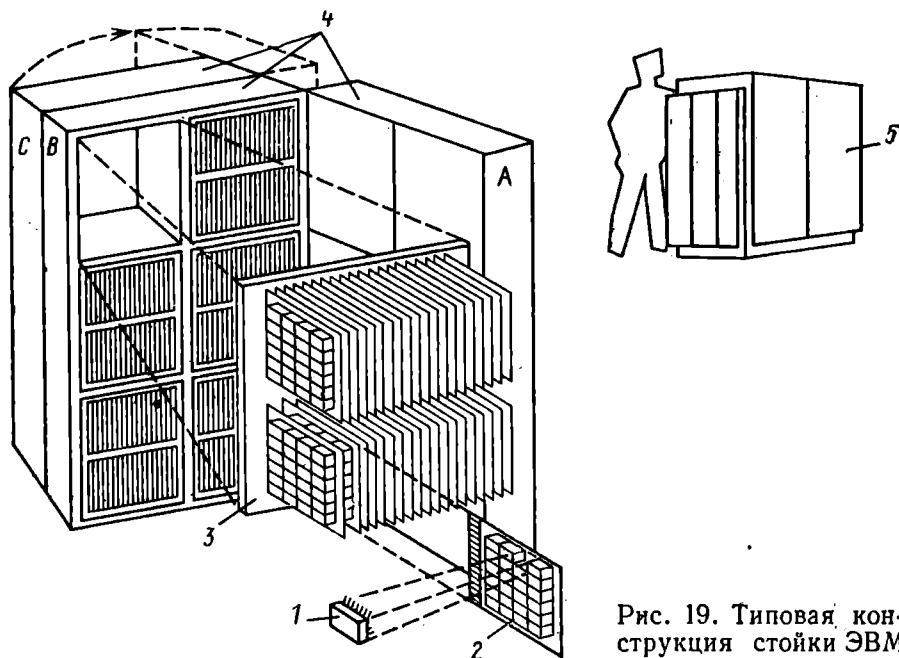


Рис. 19. Типовая конструкция стойки ЭВМ

Конструктивный модуль третьего уровня — панель состоит обычно из двух полупанелей, по 20 ТЭЗ в каждой. Применяются также комбинированные панели, содержащие наряду с ТЭЗ другие конструктивно законченные узлы. Например, панель *5C — 5Д* рамы *C* содержит ТЭЗ блока диагностики, а также блок односторонней памяти БОП6. Панели внутри рамы имеют следующие обозначения: *1A — 1B*, *1C — 1Д*, *1E — 1H*, *5A — 5B*, *5C — 5Д*, *5E — 5H* и т. п.

Модуль четвертого уровня — типовая рама — представляет собой цельносварную конструкцию из спецпрофиля и предназначена для установки модулей третьего уровня (панелей). Максимальное количество панелей в раме — шесть. Для соединения одной рамы с другими используются разъемы типа «набор» и кабели.

Питание панелей осуществляется по шинам, установленным на боковых поверхностях рам. Рядом с шинами смонтированы

фильтры питания. В нижней части каждой рамы установлены два вентилятора. Прогон воздуха осуществляется снизу вверх.

Модуль пятого уровня — стойка. Ее конструктивной основой является металлический каркас сварного типа, на которой с торцов навешиваются щиты, а с передней и задней сторон — открывающиеся двери. На верху каркаса имеется вентиляционная решетка для выхода охлаждающего воздуха. В типовой стойке установлены три рамы — две подвижные *A* и *C* и одна неподвижная рама *B* в середине. На торцовой стороне стойки процессора расположен пульт управления, выполненный в виде навешиваемой приставки. Пульт имеет поворотную панель с расположенными на ней органами управления и индикации.

Система электропитания машины подает к устройствам ЭВМ напряжение постоянного и переменного тока, обеспечивает местное и дистанционное включение и отключение электропитания центрального процессора, каналов ввода-вывода, оперативной памяти и внешних устройств, формирует сигналы защиты информации в оперативной памяти при аварии питания и сигналы аварии в других цепях электропитания, обеспечивает дистанционное управление режимом профилактического контроля по питанию центрального процессора и каналов ввода-вывода, а также контролирует напряжение в каждой фазе первичной цепи.

Напряжение первичной цепи 380/220 В \pm 10—15 % с частотой 50 ± 1 Гц подается на одно или несколько распределительных устройств — щиты питания. От последних энергия подается к блоку управления системой электропитания, блокам управления питанием, блокам включения внешних устройств, которые обеспечивают подачу с определенной последовательностью переменного напряжения к унифицированным вторичным источникам питания и обеспечивают соответствующую блокировку и защиту узлов, вторичных источников питания и устройств ЭВМ, а также сигнализируют об их рабочем состоянии. Управление системой электропитания устройств и ЭВМ в целом дистанционное, с помощью специального блока. Устройства электропитания конструктивно выполнены по модульному принципу, с размерами, кратными типовой полупанели. Каждый блок управления питанием с несколькими вторичными источниками обеспечивает электропитание одного или нескольких функциональных блоков или устройств ЭВМ.

1.9. ОПЕРАТИВНАЯ ПАМЯТЬ

В качестве оперативной памяти в состав ЭВМ ЕС-1033 могут входить устройства ЕС-3203, ЕС-3207, ЕС-3208, ЕС-3263, ЕС-3263.01.

Устройство ЕС-3203 имеет емкость 64К слов (256К байт), длину слова — 36 разрядов (32 информационных и 4 контрольных),

цикл обращения — не более 1,5 мкс, время выборки информации — не более 0,8 мкс. Оно состоит из оперативных запоминающих устройств ОП1, ОП2 и устройства их питания. Запоминающие устройства ОП1 и ОП2 размещены в стойке ЕС-3203/С001, устройство питания — в стойке ЕС-3203/С002. Каждое из запоминающих устройств, идентичных по составу, техническим данным и принципу действия, является автономным и состоит из магнитного запоминающего блока (БЗМ) и электронных схем управления.

БЗМ содержит адресные и разрядные диодно-магнитные дешифраторы, а также магнитные матрицы на кольцевых ферритовых сердечниках. Он выполнен по структуре 2,5D с трехпроводной прошивкой. Для выбора элементов памяти используется совпадение импульсов тока в адресных и разрядных шинах. Информационная емкость одного БЗМ составляет 32К слов (128К байт). Магнитные блоки БЗМ1 и БЗМ2 устройства ЕС-3203 выполнены в виде конструктивно законченных устройств и электрически связаны с электронными схемами управления ОП через разъемные соединения.

Для устройства ЕС-3203 в модели ЕС-1033 предусмотрены следующие режимы работы запоминающих устройств ОП1 и ОП2:

запись информации в магнитный блок полными словами или байтами (режим «Запись»);

считывание информации из магнитного блока полными словами и выдача ее в процессор (режим «Чтение»);

прерывание режима записи и переход в режим считывания по сигналу от процессора (режим «Блокировка записи»);

считывание информации с последующей записью единиц в заданный байт (режим «Проверить и установить»);

диагностические режимы работы «Запись РА в РИ», «Запись по фиксированному адресу», «Наложение информации», «Установка контрольного кода»;

сохранение информации в устройстве при аварийном отключении питания.

В режиме «Чтение» в устройство ЕС-3203 из процессора поступает код адреса для выборки информации из БЗМ, сигналы разрешения приема адреса в оперативную память (ПРМА) и выдачи из нее информации (ВЫД ИНФ).

В режиме «Запись» в память поступают код адреса, записываемая информация, сигналы ПРМА и ЗП, а также признаки байт. Запись информации в ОП происходит только на место тех байт, признаки которых поступают из процессора. Для остальных байт осуществляются чтение и регенерация информации.

В режиме «Запись РА в РИ» из процессора поступают сигналы передачи содержимого регистра адреса в регистр информации — РА в РИ и выдачи информации — ВЫД ИНФ. При этом код РА, сохранившийся от предыдущего цикла, передается в РИ. По сигналу ВЫД ИНФ код из РИ передается в процессор.

В режиме «Наложение информации» происходит наложение двух кодов: кода, оставшегося в РИ от предыдущего цикла работы, и кода, поступающего в РИ одновременно с сигналом НЛЖ ИНФ из процессора. Полученный таким образом код записывается в БЗМ.

В режиме «Установка контрольного кода» информация из процессора принимается в РИ. Эта информация в общем случае может быть некорректной, т. е. значения контрольных разрядов байт информации могут не соответствовать принятому дополнению. По сигналу УСТ КК из процессора восстанавливается корректность информации, и она записывается в БЗМ.

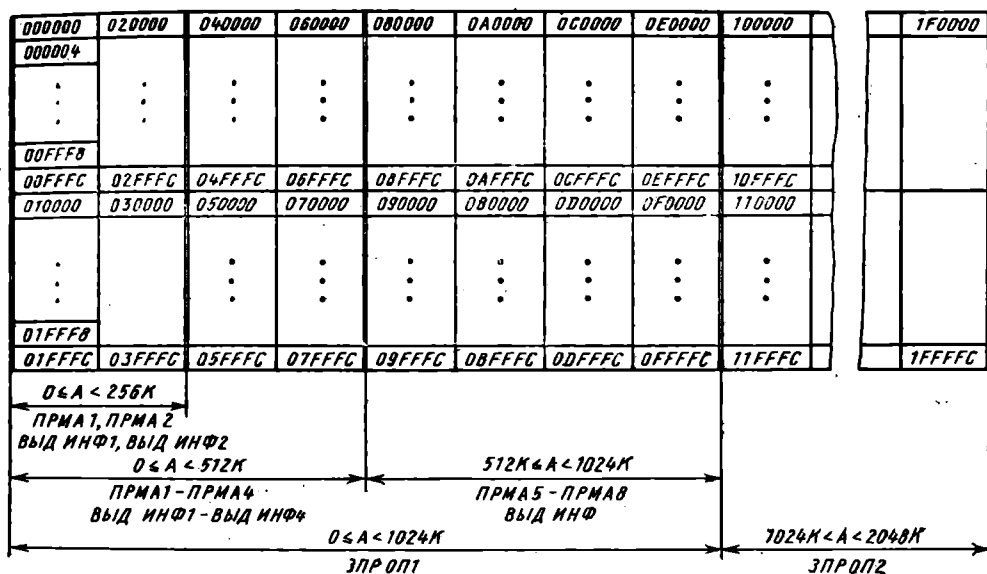


Рис. 20. Структура оперативной памяти и сигналы ее активизации

В ОП1 (ОП2) предусмотрена возможность записи и чтения информации по четырем фиксированным адресам 80, 84, 88, 8C (код шестнадцатиричный). Запись и чтение по любому фиксированному адресу производятся по входному сигналу 1ФЯ — 4ФЯ независимо от кода РА. При обращении по этим адресам через РА указанные ячейки используются как обыкновенные. В ЭВМ ЕС-1033 используются только ячейки 1ФЯ и 2ФЯ.

При емкости ОП1 (ОП2) 128К байт максимальный адрес обращения равен 01FFFC (рис. 20). Два последних разряда кода адреса нулевые, поскольку обращения в БЗМ происходят по словам. Поэтому два младших и семь старших разрядов кода адреса отбрасываются и из процессора в устройство ЕС-3203 поступают оставшиеся пятнадцать разрядов, с седьмого по двадцать второй, по значению которых происходит считывание (запись) информации из БЗМ. Код адреса поступает одновременно в ОП1 и ОП2. По значению старшего, 6-го разряда процессор формирует сигналы

ПРМА1, ПРМА2 и ВЫД ИНФ1, ВЫД ИНФ2. Соответствующая пара сигналов используется для активизации ОП1 или ОП2.

ЭВМ ЕС-1033 может работать с двумя устройствами ЕС-3203 общим объемом 512К байт, которые рассматриваются как оперативная память машины. Такая память содержит четыре устройства ОП1 — ОП4. При этом код адреса поступает одновременно на все устройства, а активизация одного из них при считывании (записи) информации происходит соответствующей парой сигналов из двух групп ПРМА1 — ПРМА4 и ВЫД ИНФ1 — ВЫД ИНФ4. Эти сигналы формируются в ЦП по значениям старших разрядов (разряды 5, 6) кода адреса оперативной памяти.

Устройство ЕС-3207 имеет емкость 128К слов (512К байт), длину слова 36 бит, минимальный цикл обращения 1,2 мкс, время выборки информации — не более 0,7 мкс. Устройство ЕС-3207 имеет модульную конструкцию, состоит из восьми модулей памяти ЕС-3941, электронных схем управления и системы электропитания. Управление для всех восьми модулей памяти является общим.

Конструктивно устройство ЕС-3207 вместе с системой электропитания выполнено в одной типовой стойке.

Модуль памяти ЕС-3941 представляет собой съемный блок и является автономным запоминающим устройством емкостью 64К ячеек. Каждая ячейка модуля памяти имеет 9-разрядный формат. При считывании слова из памяти обращение по одному адресу осуществляется одновременно к четырем модулям. Таким образом, 4 модуля памяти составляют запоминающий массив емкостью 64К слов, и обращение к нему осуществляется по адресам 000000 — 03FFFC. Остальные 4 модуля составляют аналогичный запоминающий массив, обращение к которому происходит по адресам 040000 — 07FFFC. Модуль памяти выполнен на кольцевых ферритовых сердечниках по структуре 2,5D с трехпроводной прошивкой.

Устройство ЕС-3207 в составе ЭВМ ЕС-1033 используется в таких же режимах работы, что и устройство ЕС-3203. Для записи (считывания) информации из процессора поступает 15-разрядный код адреса. Этот код соответствует максимальному адресу обращения 01FFFC, тогда как в ЕС-3207 максимальный адрес обращения составляет 07FFFC. Поэтому в ОП формируются значения двух старших разрядов кода адреса (разряды 5, 6) из сигналов ПРМА1 — ПРМА4 и сигнал ВЫД ИНФ из сигналов ВЫД ИНФ1 — ВЫД ИНФ4, поступающих из процессора (см. рис. 20).

ЭВМ ЕС-1033 может работать с двумя устройствами ЕС-3207 (ОП1 и ОП2), которые рассматриваются как ОП машины объемом 1024К байт. Код адреса поступает на оба из них; сигналы из процессора ПРМА1 — ПРМА4 и ВЫД ИНФ1 — ВЫД ИНФ4 поступают в ОП1 для формирования в ней двух старших, 5 и 6, разрядов кода адреса и сигналов ПРМА и ВЫД ИНФ, а сигналы ПРМА5 — ПРМА8 и ВЫД ИНФ поступают из процессора в ОП2 для формирования этих же старших разрядов кода адреса и сиг-

нала ПРМА для ОП2. В процессоре сигналы ПРМА1 — ПРМА8 и ВЫД ИНФ1 — ВЫД ИНФ4, ВЫД ИНФ формируются по значениям разрядов 4, 5, 6 кода адреса ОП.

Устройство ЕС-3208 по своим техническим данным аналогично устройству ЕС-3207. Конструктивно ЕС-3208 также расположено в одной типовой стойке, имеет модульную структуру, состоит из четырех модулей памяти ЕС-3945, электронных схем управления и системы электропитания. Управление для всех четырех модулей памяти является общим.

Модуль памяти ЕС-3945 конструктивно представляет собой съемный блок и является автономным запоминающим устройством емкостью 64К 18-разрядных ячеек с выборкой информации по структуре 3D с трехпроводной прошивкой:

Подключение одного или двух устройств ЕС-3208 к ЭВМ ЕС-1033 также аналогично.

Все три устройства в качестве запоминающих элементов содержат кольцевые ферритовые сердечники. Считывание происходит с разрушением и регенерацией информации.

Устройство ЕС-3263 имеет емкость 256К слов (1024К байт), минимальный цикл обращения — 1,5 мкс, время выборки информации — не более 0,8 мкс. Оно имеет модульную конструкцию, состоит из восьмидесяти модулей памяти, электронных схем управления и системы электропитания. Конструктивно устройство ЕС-3263 выполнено в одной типовой стойке вместе с системой электропитания.

Модуль памяти ЕС-3263.Е001 выполнен на съемной печатной плате и служит для приема, хранения и выдачи информации. В качестве запоминающих элементов используются интегральные микросхемы К565РУ1А. Считывание осуществляется без разрушения информации.

Микросхема К565РУ1А (рис. 21) представляет собой запоминающее устройство динамического типа емкостью 4096 однобитных ячеек.

В режиме «Чтение» на адресные входы А0 — А11 поступает 12-разрядный код адреса, на вход \overline{CS} подается сигнал низкой полярности ВЫБОР КРИСТАЛЛА, на вход CE — сигнал разрешения высокой полярности. С выхода $\overline{D0}$ снимается значение считанного бита информации в обратном коде.

В режиме «Запись» на входы микросхемы поступает такой же набор сигналов; дополнительно на вход \overline{WE} подается сигнал

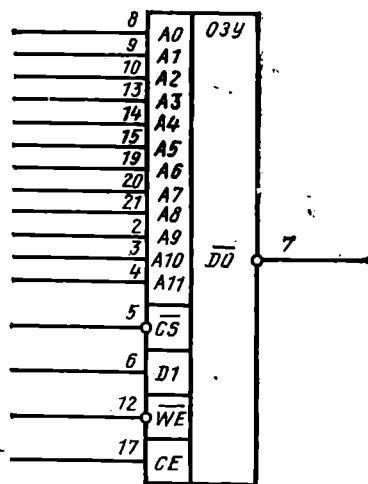


Рис. 21. Микросхема К565РУ1А

низкой полярности «Запись», а на вход D1 — бит записываемой информации в прямом коде. Минимальное время цикла записи (считывания) составляет 400 нс.

Микросхема обеспечивает сохранность записанной информации в течение 2 мс. Более длительное хранение требует периодической регенерации информации. Регенерация осуществляется за один цикл обращения в любом режиме (записи или считывания). Особенностью микросхемы является наличие «третьего состояния» выхода, которое обеспечивается отсутствием одного из сигналов на входах \overline{CS} либо CE . Это позволяет объединять выходы нескольких микросхем при наращивании объема памяти. Номиналы питающих напряжений составляют +5, —5 и +12 В.

Модуль памяти ЕС-3263.Е001 является запоминающим блоком емкостью 16К 8-разрядных ячеек и содержит 32 микросхемы К565РУ1А.

Применение ИС К565РУ1А в качестве запоминающих элементов обусловило ряд особенностей построения устройства ЕС-3263. О первом из них — необходимости регенерации информации — говорилось выше. В силу этой же особенности при отключении питания информация в ОП не сохраняется. Другой особенностью является применение метода контроля считывания информации из модуля памяти с исправлением одинарной и обнаружением двойной ошибки с использованием корректирующего кода Хэмминга.

Поскольку в ЭВМ ЕС-1033 используется метод контроля каждого байта информации по $\text{mod}2$, процесс контроля при чтении из ОЗУ осуществляется в 2 этапа: информация, считанная из накопительного блока, контролируется с помощью кода Хэмминга, затем блок управления формирует по $\text{mod}2$ контрольный бит для каждого байта считанного слова, и информация передается в процессор. В режиме записи каждый байт информации, поступающей из процессора, подвергается в ОЗУ контролю также по $\text{mod}2$, затем, если не обнаружено ошибок, контрольные биты игнорируются, блок управления формирует биты корректирующего кода, и информация вместе с этими битами запоминается в накопительном блоке. При обнаружении ошибки в информации, поступающей из процессора, записи в накопительный блок не происходит, а в процессор выдается сигнал ОШ ИНФ.

В устройстве ЕС-3263 периодически осуществляется регенерация как при считывании, так и при записи. При отсутствии запроса процессора регенерация осуществляется по сигналу внутреннего запроса ОЗУ на регенерацию ЗПР РЕГ, который вырабатывается в блоке управления. При наличии запроса процессора, имеющего более высокий приоритет, чем запрос регенерации, формирование сигнала ЗПР РЕГ блокируется. В этом случае цикл регенерации осуществляется после обслуживания запроса процессора. Длительность цикла регенерации — 400 нс. Цикл обращения к памяти в ЭВМ ЕС-1033 составляет 1,6 мкс при работе

с любым ОЗУ, что обусловлено длительностью такта процессора, равной 400 нс. Цикл регенерации устройства ЕС-3263 осуществляется внутри цикла обращения процессора к памяти, поэтому регенерация не вызывает замедления работы ЭВМ.

Таким образом, устройство ЕС-3263 используется в составе ЭВМ ЕС-1033 в тех же режимах работы, что и вышеописанные ферритовые запоминающие устройства. Максимальный адрес обращения равен 0FFFFC. Из процессора в ОП поступают все 18 разрядов этого адреса. Сигналом, активизирующим работу ОП, является сигнал запроса к ней ЗПР ОП, а сигналы ПРМА1 — ПРМА8 используются для контроля значений разрядов 4, 5, 6 кода адреса (см. рис. 20).

Устройство ЕС-3263.01 имеет емкость 512К слов (2048К байт) и содержит два устройства ЕС-3263 (ОП1 и ОП2) в одной типовой стойке вместе с системой электропитания. Максимальный адрес обращения равен 1FFFFC. Все 18 разрядов кода адреса подаются на ОП1 и ОП2 одновременно, а по значению старшего разряда адреса процессор формирует два сигнала запроса к ОП — ЗПР ОП1 или ЗПР ОП2, которые активизируют соответствующую оперативную память (см. рис. 20).

1.10. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Программное обеспечение ЭВМ ЕС-1033 базируется на операционной системе ОС ЕС. Кроме операционной системы оно включает набор пакетов прикладных программ общего назначения, которые поддерживают проблемные ориентации ЭВМ ЕС-1033, и пакеты прикладных программ, расширяющих возможности ОС ЕС и обеспечивающих использование ЭВМ в различных режимах работы. В состав программного обеспечения входят также комплект программ технического обслуживания (КПТО) и комплект неавтономных тестов устройств (КНТУ).

Возможности операционной системы ОС ЕС подробно изложены в литературе, например, в работах [35, 47] и др. Здесь следует лишь отметить, что ОС ЕС, входящая в состав программного обеспечения ЭВМ ЕС-1033, включает средства, позволяющие восстанавливать вычислительный процесс после ошибок центрального процессора и каналов ввода-вывода в тех случаях, когда аппаратурные средства процессора и каналов не в состоянии сами это сделать. Использование этих средств в составе ОС ЕС значительно повышает живучесть вычислительной системы.

В состав пакетов прикладных программ общего назначения входят пакеты научных подпрограмм на языках ПЛ/1 и ФОРТРАН-IV, пакеты программ для решения матричных задач и задач математического программирования, пакеты программ для моделирования аналоговых систем и непрерывных процессов, пакеты программ для моделирования дискретных систем.

Пакеты научных программ обеспечивают решение задач линейной алгебры, численного дифференцирования и интегрирования, решение линейных и дифференциальных уравнений, вычисление собственных векторов, собственных значений и специальных функций, позволяют выполнять элементарные операции с матрицами, решать задачи аппроксимации, интерполяции, математической статистики и численного анализа.

Пакет программ для решения матричных задач позволяет обрабатывать матрицы большой размерности, автоматически распределяя основную и внешнюю память.

Пакет программ для решения задач математического программирования предназначен для решения оптимизационных задач методами линейного и сепарабельного программирования. Эти программы могут быть эффективно использованы при решении задач управления в народном хозяйстве.

Пакет программ для моделирования аналоговых систем и непрерывных процессов ориентирован на численное моделирование систем и процессов, описываемых обыкновенными дифференциальными уравнениями. Программы пакета могут успешно применяться в таких областях, как теория автоматического управления, электроника, механика и электротехника. Пакет программ имеет входной язык, не требующий больших навыков в программировании, позволяет легко описывать модель непосредственно по структурной схеме или системе дифференциальных уравнений, предоставляет возможность построения проблемно ориентированных расширений пакета для различных областей науки и техники.

Пакет программ для моделирования дискретных систем позволяет проводить исследования информационных (в процессе проектирования и эксплуатации) и вычислительных систем, осуществлять выбор конфигурации средств вычислительной техники, исследовать системы массового обслуживания. Программы пакета обеспечивают автоматический сбор и выдачу разнообразных статистических данных о функционировании системы, например, коэффициентов использования оборудования, параметров очередей и т. д.

К пакетам прикладных программ, расширяющим возможности ОС ЕС, относятся супервизор реального времени, система разделения времени и подсистемы планирования КРОС и РОС (программы расширения операционной системы, К — индекс разработчика). Эти программы, а также упоминавшиеся выше средства восстановления вычислительного процесса являются оригинальными разработками [4, 41, 42, 53].

Супервизор реального времени обеспечивает эффективное использование ЭВМ в системах управления объектами в реальном масштабе времени. Он позволяет обслуживать нестандартные устройства ввода-вывода и линии прямого доступа; обеспечивает малое время ответа при обработке поступающих от них данных,

возможность жесткой согласованности моментов выполнения обрабатываемых программ вычислительной системы в реально протекающим временем и возможность использования методов и техники программирования, соответствующих условиям применения систем управления процессами или объектами при жестких ограничениях на время ответа.

Система разделения времени представляет собой совокупность программных средств, которые позволяют обслуживать одновременно нескольких пользователей, связанных с ЭВМ с помощью абонентских пунктов или устройств ввода-вывода. Пользователи могут создавать, транслировать, редактировать, отлаживать и выполнять программы в диалоговом режиме, а также передавать задания для выполнения их в режиме пакетной обработки. Система разделения времени обеспечивает ряду пользователей одновременный доступ практически по всем средствам, предоставляемым ОС ЕС, и выдачу результатов за минимальное время. Одновременно с обслуживанием пользователей в режиме разделения времени на ЭВМ может выполняться пакетная обработка заданий.

Подсистемы планирования КРОС и РОС предназначены для повышения производительности соответственно одно- и многомашиной вычислительных систем.

КПТО и КНТУ представляют собой тестовое обеспечение ЭВМ и служат для проверки оборудования при проведении пусконаладочных, профилактических и регламентных работ.

ГЛАВА 2

ОПЕРАЦИОННАЯ ЧАСТЬ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

2.1. СТРУКТУРА ПРОЦЕССОРА

Центральный процессор ЕС-2433 выполняет арифметические операции с фиксированной и плавающей запятой, операции десятичной арифметики, логические операции, операции переходов и управления, операции обращения к каналам ввода-вывода, организует обращение к оперативной памяти, обрабатывает прерывания.

Принцип управления в ЦП — микропрограммный. Микрокоманды имеют 132 информационных разряда. Они хранятся в ПЗУ, состоящем из четырех блоков БОП 2048-33 (п. 1.7). Цикл работы процессора равен 0,4 мкс, что обеспечивает среднее выполнение короткой операции за время, не превышающее 1 мкс. Оборудование ЦП размещено в одной типовой раме ЕС ЭВМ (рама В стойки процессора ЕС-2433).

Как уже отмечалось, в системах с магистральной структурой можно организовать связь между несколькими любыми функциональными блоками по одной из магистралей, причем один блок будет являться источником информации на магистраль, а все остальные — приемниками.

В процессоре ЕС-1033 для передачи информации используется несколько однотипных магистралей, а функциональные блоки связаны более чем с одной из них. Таким образом, в каждом такте можно организовать несколько трактов передачи информации. Поскольку микропрограммное управление процессора является общим для всех его функциональных блоков, то оказывается возможным организовать связь между ними оптимальным для данного такта образом. В большинстве тактов обработки информации участвует три числа, два из которых — исходные операнды, а третье — результат. Кроме того, в каждом такте нужно передавать информацию о состоянии блоков процессора от схем анализа к схемам управления, которая необходима для выбора пути продолжения исполняемой микропрограммы. Таким образом, для организации связей между блоками процессора необходимо и достаточно иметь три информационных магистрали и одну магистраль для сбора информации о состоянии функциональных блоков — магистраль анализов.

Процесс обработки информации в ЦП заключается в выполнении ряда пересылок между блоками и арифметической (логической) обработке в комбинационных схемах отдельных блоков. Для обеспечения работы комбинационных схем магистрали имеют встроенные элементы памяти, вследствие чего считанная на них информация запоминается и сохраняется до тех пор, пока не поступит сигнала на сброс магистралей. Указанная особенность позволяет записать результат операции в тот же регистр, откуда производилось чтение операнда в начале такта работы, причем чтение и запись могут быть выполнены как с одной, так и с разных магистралей. Как правило, магистрали сбрасываются каждый такт, но предусмотрена возможность хранения информации на магистралях заданное время в пределах нескольких тактов.

В процессоре принято, что любые считывания на магистрали производятся по первой синхросерии С1, а все записи выполняются в конце такта по второй синхросерии С2. Промежуток между С1 и С2 отведен для работы комбинационных обрабатывающих схем.

В общем случае результатом операции может быть не только результат арифметической или логической обработки операндов, но и установка признака результата и выработка сигнала прерывания. Ряд команд выполняется именно с целью установки признака результата или с целью вызвать прерывание. В процессоре ЭВМ ЕС-1033, в отличие от других машин, выработка признака результата и сигналов прерываний производится аппаратурно в процессе обработки операндов. Это позволило достичь высокой

скорости выполнения команд и, в особенности, команд обработки данных фиксированных форматов.

Центральным блоком процессора является блок магистрального коммутатора (БМК) (рис. 22). Он предназначен для приема информации от всех других блоков процессора, хранения ее в те-

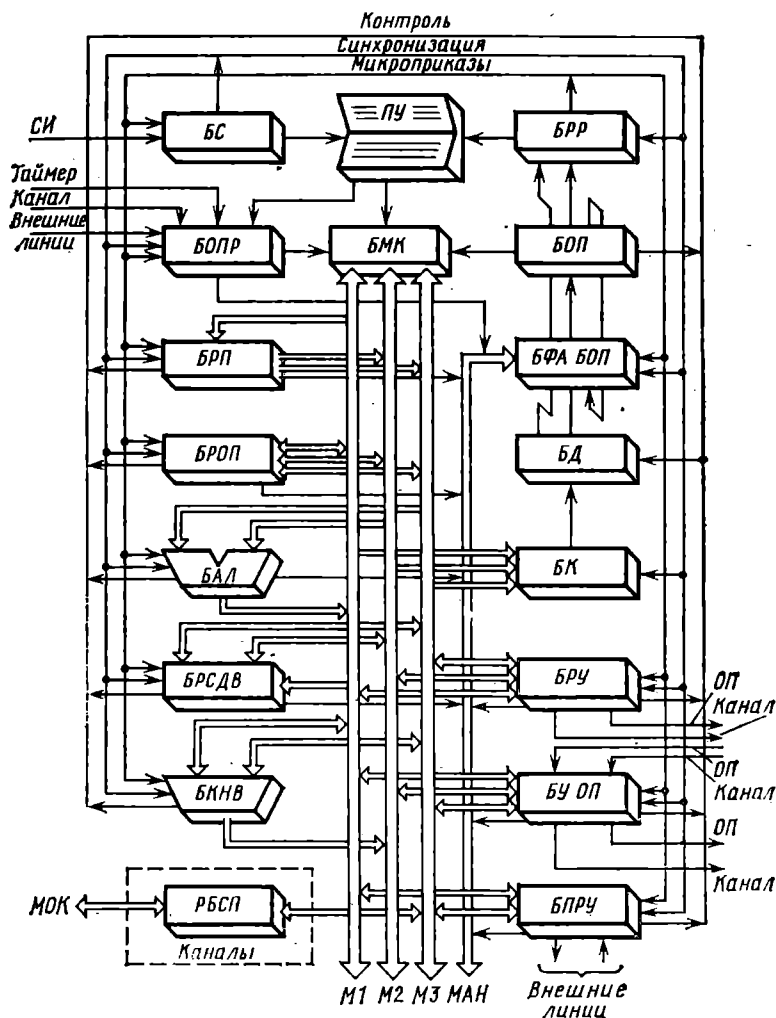


Рис. 22. Структурная схема процессора

чение заданного времени и передачи информации в блоки процессора вместе с ее логической обработкой и контролем. Три 32-разрядные информационные магистрали, служащие основными трактами передачи данных в процессоре, равноценны и именуются первой M1, второй M2 и третьей M3 магистралями.

Для хранения промежуточных результатов вычислений предназначен блок операционных регистров (БРОП) [А. с. 877613 (СССР)]. Шестнадцать 32-разрядных регистров, входящих в этот блок, связаны с тремя магистралями. В одном такте работы процессора считывание данных из различных регистров и запись в них

осуществляется независимо и одновременно относительно всех магистралей.

Блок программных регистров (БРП) [А. с. 613402 (СССР)] состоит из шестнадцати 32-разрядных регистров общего назначения и четырех 64-разрядных регистров для операндов с плавающей запятой. Особенностью этого блока является то, что данные из его двух различных регистров могут быть считаны на разные (М2 и М3) магистрали одновременно, и в этом же такте может быть осуществлена запись информации в один из регистров с магистрали М1.

Арифметико-логический блок (БАЛ) [38] является основным блоком обработки поступающей информации. Он выполняет 45 различных операций обработки двоичных операндов, а также двоично-десятичное сложение и вычитание. Схема обрабатываемой части комбинационная и не имеет каких-либо входных и выходных регистров. При задании управляющей функции результат операции над числами, поступающими в БАЛ с М2 и М3, выдается на М1 через 0,1—0,12 мкс.

Отличительной особенностью процессора является наличие специального мощного блока для сдвига информации (БРСДВ) [А. с. 585755 (СССР), а. с. 591078 (СССР)]. Он обеспечивает сдвиг вправо и влево до 128 разрядов одновременно. Блок может работать в режимах как логического, так и арифметического сдвигов, т. е. с распространением знака и контролем переполнения, и служит для накопления результатов умножения, деления, выполняет операции нормализации и выравнивания операндов и т. п.

Операции преобразования чисел из двоичной системы счисления в двоично-десятичную, и наоборот, выполняются аппаратно блоком конвертирования (БКНВ) [А. с. 591073 (СССР)]. Преобразование выполняется за несколько циклов под микропрограммным управлением.

Блок управляющих регистров (БРУ) состоит из регистров, содержащих информацию о состоянии системы. Это 64-разрядный регистр слова состояния программы (РССП) и регистр адреса процессора (РАП), через который в оперативную память системы поступают адреса команд и данных. Буферный регистр адреса хранит адрес выполняемой команды для ее автоматического повторения при сбоях.

Блок управления оперативной памятью (БУ ОП) обслуживает не только центральный процессор, но и каналы ввода-вывода. Он организует обращение к оперативной памяти, осуществляя ее активизацию и управление работой, задает режим записи или считывания, выдает в ОП адреса обращения, принимает и выдает информацию от ЦП и каналов. Запросы процессора и каналов ввода-вывода удовлетворяются по приоритету, причем высшим обладают каналы. Память ключей защиты, входящая в состав блока, обеспечивает постраничную защиту оперативной памяти при записи и считывании. Это необходимо для организации мультипрограммной работы машины.

Блок прямого управления (БПРУ) реализует связь процессора с другой ЭВМ или специальными внешними устройствами через интерфейс прямого управления. Обмен информацией выполняется по командам ПРЯМОЕ ЧТЕНИЕ и ПРЯМАЯ ЗАПИСЬ.

Блок режимов работы (БРР) предназначен для перевода машины в различные режимы работы — автоматический, шаговый или тактовый и для выработки микроприказов. Режимы задаются вручную с пульта управления. Микроприказы вырабатываются группой дешифраторов, управляющая информация (коды полей микрокоманд) на которые поступает из блоков односторонней памяти.

Вся управляющая микропрограммная информация записана в блоке памяти микропрограмм, который состоит из четырех унифицированных блоков — БОП0—БОП3 (п. 1.7). Управляющая информация (микрокоманда) считывается одновременно из всех четырех блоков по одному адресу. Часть информации поступает на информационные магистрали в виде константы или маски. Адресная часть служит для формирования адреса следующей микрокоманды. Операционная часть, состоящая из полей определенной разрядности, декодируется в БРР и управляет работой ЦП в данном такте.

Управление выборкой микрокоманд из БОП осуществляет блок формирования адреса БФА БОП [А. с. 613401 (СССР)]. Адресная часть текущей микрокоманды содержит базовый адрес, который на линейном участке микропрограммы является адресом следующей микрокоманды. Если необходимо изменить ход микропрограммы, то базовый адрес изменяется (модифицируется). Модификация осуществляется в зависимости от результатов анализа состояния тех или иных схем, от особенностей данных и результата обработки [А. с. 591075 (СССР), а. с. 615538 (СССР), а. с. 648984 (СССР)]. Информация о результатах выполнения анализов собирается на специальной четырехразрядной магистрали анализов (МАН).

Реакцию на внутренние и внешние прерывания обеспечивает блок обработки прерываний (БОПР). Он фиксирует сигналы причин прерываний и организует обработку последних в порядке приоритетности. Блок подключен к пульту управления системой (для возможности задания прерывания оператором), к таймеру, к каналам ввода-вывода и внешним синхролиниям от другой машины.

Блок диагностики (БД) процессора обеспечивает анализ и исправление сбоев, а также быструю автоматическую локализацию возникшей неисправности [48]. Особенностью этого блока является наличие собственной управляющей памяти, в которой хранятся микропрограммы диагностических процедур [А. с. 613651 (СССР)].

Информация об обнаруженных в блоках машины ошибках поступает в БД от схем контроля, имеющихся во всех блоках цен-

трального процессора. Условно эти схемы объединены в блок контроля (БК) [49].

Пульт управления системой (ПУ), традиционно относящийся к процессору, предназначен для визуального контроля работы ЦП и каналов, включения электропитания, инициирования начальной загрузки, вмешательства оператора в ход вычислительного процесса, для технического обслуживания и поиска неисправности. С пульта можно записать в ОП и прочитать из нее информацию, занести данные в программные регистры, вызвать внешнее прерывание и т. п.

Центральный процессор обменивается информацией с каналами ввода-вывода через буферный регистр связи (РБСП). Запоминающие элементы этого регистра одними своими входами-выходами подключены к третьей магистрали ЦП, а другими — к магистралям общего канала (МОК).

Блок синхронизации (БС) процессора обеспечивает прием в раму B синхроимпульсов СИ пяти основных синхросерий от блока синхронизации машины (п. 1.6), их усиление и размножение. Кроме того, он формирует ряд вспомогательных синхросерий. Главными из них являются блокируемые серии С1БЛ, С2ЗП, С1'БЛ и С2'БЛ, которые формируются соответственно из синхросерий С1, С2, С1' и С2'. Необходимость в блокируемых сериях вызвана следующими причинами.

При работе процессора с ОП необходимо задержать выполнение микропрограммы на промежуток времени от того момента, когда процессору требуется информация из ОП, запрос на которую был уже дан, до момента, когда эта информация фактически будет получена. Поскольку цикл работы ОП в четыре раза больше цикла работы процессора, то после выдачи в ОП запроса на информацию процессор должен ожидать ее минимум три такта. Обычно в микропрограмме запрос выдается заранее за три или более тактов для того, чтобы к нужному моменту времени информация из ОП была уже получена. Но даже и в этом случае промежуток времени от выдачи запроса до получения информации может оказаться довольно значительным, если в момент (или до) выдачи запроса процессором ОП была занята обслуживанием запросов каналов (запросы каналов имеют более высокий приоритет). Задержка микропрограммы осуществляется зацикливанием ее на определенной микрокоманде. Чтобы запретить многократное выполнение некоторых действий, предусмотренных этой микрокомандой, производится блокирование поступления синхросерий (рис. 23) в те узлы и блоки, которые выполняют эти действия (заметим, что выполнение каждого микроприказа, как правило, строится определенной синхросерией).

Для взаимодействия ОП с микропрограммным управлением процессора служит триггер очередности (ТОЧП), обычное состояние которого единичное. Триггер сбрасывается сигналом запроса процессора (ЗПРП), а вновь устанавливается тогда, когда разре-

шен прием считанной из ОП информации в регистр информации процессора (РИП) микроприказом $РИП := ОП$. Другими словами, состояние $ТОЧП = 0$ соответствует промежутку времени от выдачи запроса в ОП до получения информации из ОП. Анализ состояния $ТОЧП$ происходит микропрограммно по сигналу $ТОЧП \neq 0$. Как нулевое состояние $ТОЧП$, так и наличие сигнала $ТОЧП \neq 0$ низкого уровня приводит к блокированию синхросерий С1 и С2. На рис. 23 показан случай, когда сигналы ЗПРП и $ТОЧП \neq 0$ были сформированы в одной микрокоманде, а между выдачей запроса в ОП и началом его обслуживания (формирования сигнала ПРМА) прошло некоторое время. Микроприказ

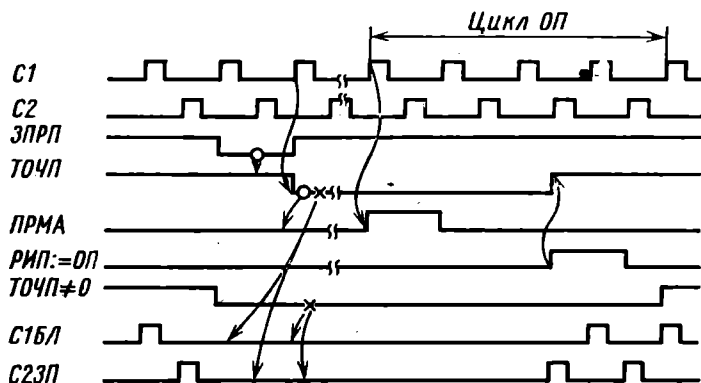


Рис. 23. Формирование блокируемых синхросерий С1БЛ и С23П

$ТОЧП \neq 0$ всегда выдается в той микрокоманде, где фактически должна быть использована получаемая из ОП информация, а запрос в ОП за нее может быть сделан либо в том же такте, как на рис. 23, либо заранее за любое число тактов. Возможен случай, когда информация из ОП оказывается полученной до того, как будет выдан микроприказ $ТОЧП \neq 0$. Тогда остановки микропрограммы и других действий, с ней связанных, не происходит.

Если $ТОЧП = 0$, то микрокоманда, содержащая анализ $ТОЧП \neq 0$, заикливаясь сама на себя до тех пор, пока не выполнится условие $ТОЧП \neq 0$. Заикливаясь, микрокоманда выполняется многократно. Если в этой микрокоманде имеются микроприказы изменения содержимого адресных регистров (РАП, адресная часть РССП, адресные регистры РОН, выполненные в виде счетчиков), то может произойти их многократное изменение. Кроме того, если в этой же микрокоманде имеются микроприказы записи считанной из ОП информации в регистры, то может произойти запись недостоверной информации, что, в свою очередь, может вызвать, например, ложное прерывание. Чтобы предотвратить эти явления, выполнение указанных действий стробируется блокируемыми синхросериями. Таким образом, содержимое счетчи-

ков меняется только один раз, запись в регистры происходит только тогда, когда из ОП поступит запрошенная информация, а результат обработки будет истинным.

По синхросерии С1БЛ происходит изменение содержимого всех адресных регистров. Содержимое РАП и адресной части РССП изменяется в процессе выборки команды (а РАП — и при выполнении команды) на +2 или +4 (т. е. на 2 или 4 байта), а значение содержимого счетчиков РОН, которые являются адресными регистрами РОН — на +1 или -1 (т. е. номер РОН может быть уменьшен или увеличен на единицу). По синхросерии С1'БЛ осуществляется контроль этих процессов.

По синхросерии С2ЗП происходит запись в регистры результатов обработки полученной из ОП информации, по синхросерии С2'БЛ — контроль этой записи. Блокировка серий С1' и С2' происходит аналогично.

Некоторые блоки ЦП (например, блок конвертирования) имеют в комбинационной части схем значительное число последовательно включенных логических элементов. Для их работы необходимо время, превышающее длительность одного машинного такта. С этой целью синхросерия СБРМ может блокироваться в текущем такте микроприказом БЛ СБРМ. При выполнении диагностических процедур запись с магистрали также может быть выполнена через несколько тактов после чтения информации на магистраль. Для сохранения информации на магистрали блок диагностики ЦП с помощью своего микроприказа блокировки сброса магистрали БСМ может блокировать серию СБРМ.

Кроме названных в ЦП для отдельных узлов и схем вырабатываются и другие дополнительные синхросерии.

Центральный процессор обрабатывает информацию в соответствии с командами программы. Операции обработки подразделяются на арифметические и логические. В зависимости от типа структуры данных арифметические операции могут быть операциями над числами с фиксированной запятой, над числами с плавающей запятой и над десятичными числами. Есть операции, объединяющие некоторые виды обработки, — это выполнение преобразований чисел из двоичной в двоично-десятичную систему счисления, и наоборот.

В логических операциях многоразрядные операнды воспринимаются как наборы двоичных символов. Над ними производятся поразрядные действия в соответствии с правилами алгебры логики. К логическим относятся и операции преобразования форматов данных — упаковка, распаковка, редактирование текстов, транслирование таблиц и т. п. Следует отметить, что обработка адресов данных и команд выполняется с использованием тех же принципов и того же оборудования, что и при работе с потоком преобразуемых данных.

Сочетание эффективных средств обработки данных с магистральным принципом их организации позволило достичь высокого

быстродействия ЭВМ. Например, для операций формата RR оно составляет примерно 10^6 операций в секунду.

К основным обрабатывающим блокам относятся арифметико-логический блок, блок регистров сдвига, блок преобразования чисел из одной системы счисления в другую. К средствам обработки относятся также схемы подготовки управляющей информации для БРСДВ — узел вычисления параметра нормализации, узел вычисления параметра сдвига при выравнивании операндов переменной длины, а также специальные связи регистра информации РИП с первой магистралью для байтной обработки ЛРИП и обработки полуслов ПРИП (п. 3.4).

2.2. БЛОК МАГИСТРАЛЬНОГО КОММУТАТОРА

БМК является центральным блоком процессора и предназначен для приема информации от других функциональных блоков, ее контроля и хранения в течение заданного времени, логического преобразования передаваемой информации и передачи ее в другие блоки процессора и каналы ввода-вывода.

В состав БМК входят (рис. 24): три магистрали М1, М2 и М3 по 32 информационных разряда в каждой; три коммутатора магистральных КМ М1, КМ М2 и КМ М3; два регистра масок РМС1 и РМС2 по 32 информационных разряда каждый; схемы контроля К; схема байтного обнуления.

В блоке магистрального коммутатора узлы КМ М1, КМ М2 и КМ М3 являются коммутаторами первой, второй и третьей магистралей соответственно. У каждого узла показаны три секции, обозначенные соответствующим индексом, и «быстрый» выход, помеченный буквой Б (для М1, например, соответственно М1-1, М1-2, М1-3 и М1Б). Каждая магистраль имеет собственную схему контроля.

Для обеспечения работы комбинационных схем арифметического блока, конверторов, схем анализа, а также возможности записи обработанных данных обратно в регистр-источник магистрали снабжены элементами памяти. Они осуществляют хранение считанной и отмаскированной информации в течение требуемого времени (в большинстве случаев на один такт ЦП). Для обеспечения работы схем контроля информации, считанной на магистрали, необходимо ее зафиксировать до начала маскирования, так как оно может изменить ее четность. Сохранение исходной информации осуществляется на отдельных элементах памяти.

Маска в виде 32-разрядной константы (КОНСТ) поступает из БОПЗ непосредственно через регистр микрокоманд РМКЗ в коммутатор масок, а также в два специальных регистра масок РМС1 и РМС2. В последние маска записывается по микроприказам РМС1 := КОНСТ и РМС2 := КОНСТ. Таким образом, в коммутаторе имеется возможность подавать маски на магистрали в любых комбинациях как из регистров маски, так и непосредственно из

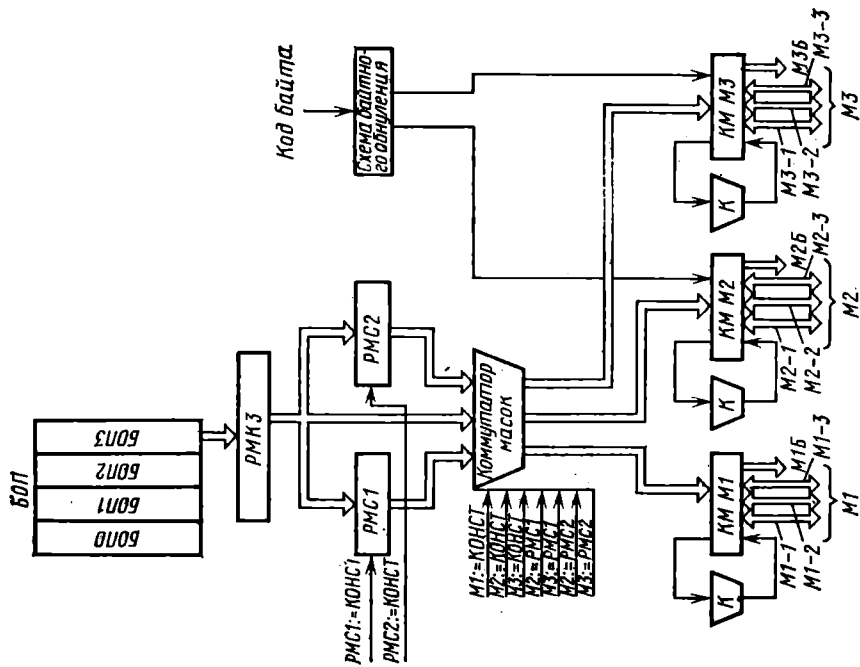


Рис. 24. Структурная схема блока магистрального коммутатора

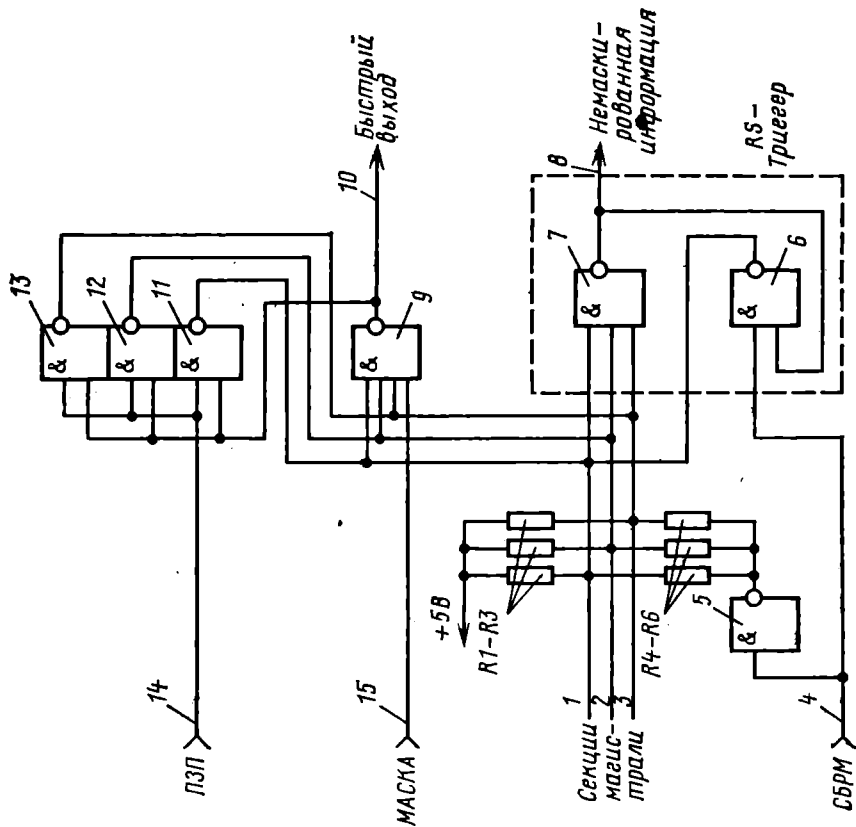


Рис. 25. Разряд магистрали

РМКЗ. Чтобы подать на магистрали три различные маски одновременно, две из них должны быть записаны предварительно в РМС1 и РМС2. На первую магистраль маска подается только из РМКЗ.

После выполнения маскирования обработанная информация снабжается истинными контрольными разрядами, которые формируются специальными схемами свертки по mod2. Одной из самых сложных задач является обеспечение устойчивости обмена информацией между блоками на частоте порядка 5 МГц. Магистрали обладают большой распределенной емкостью монтажа. Это требует принятия специальных мер борьбы с влиянием паразитных емкостей. С этой целью осуществлено секционирование гальванических связей. Секции затем логически объединяются. Кроме того, для ускорения подготовки очередного цикла работы специальными средствами форсируется заряд паразитных емкостей магистралей.

Каждый разряд магистрали разбит на три секции, гальванически не связанные между собой (рис. 25). Такой прием позволяет значительно сократить время заряда паразитных емкостей магистрали и, следовательно, повысить скорость передачи информации. К каждой секции магистрали подключается не более 16 входов и 10 выходов логических схем блоков.

Функционально три секции магистрали представляют собой единую информационную магистраль. Физически же это автономные, электрически развязанные цепи. Именно к этим секциям и подключаются входы-выходы блоков процессора, причем каждая секция объединяет блоки, конструктивно расположенные близко друг от друга. Резисторы R1—R3 являются общей нагрузкой элементов с открытым коллектором, подключенных к соответствующим секциям магистралей.

Исходным (сброшенным) состоянием магистрали является единичное состояние (потенциал около 3,5 В). Всякое считывание единицы на магистраль не изменяет этого состояния, чтение нуля приводит к изменению состояния магистрали (ее потенциал понижается до величины порядка 0,4 В). Таким образом, информация на магистрали представляется в обратном коде.

Считанная сигналом чтения ЧТ (рис. 26) по синхросерии С1 на любую из трех секций магистрали, информация поступает на один из входов четырехвходовой схемы И-НЕ 9 (см. рис. 25). На

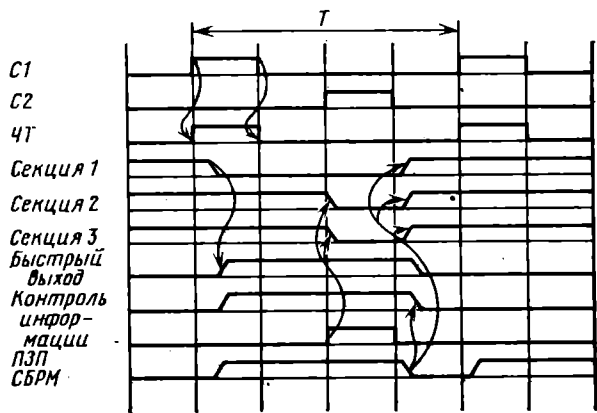


Рис. 26. Временная диаграмма работы магистрали при считывании нулевой информации на первую секцию

вход 15 этой же схемы поступает значение разряда маски. При единичной маске на выходе 10 элемента И-НЕ 9 сохраняется значение считанной на магистраль информации. Выход 10 называется «быстрым выходом магистрали». Он подключен ко входам БАЛ, схемам сдвигателя и т. п. для немедленной обработки информации, поступающей на магистраль от любого из блоков.

Подача нулевой маски на элемент И-НЕ 9 эквивалентна считыванию в данном разряде нулевой информации. Одновременно немаскированная информация запоминается в RS-триггере, образованном элементами И-НЕ 6 и 7, и поступает с его выхода 8 в схемы контроля. Считанная в одну из секций (например, первую) информация фиксируется с помощью триггера. В результате значение потенциала сохраняется на «быстром выходе» 10 и после снятия сигнала чтения (см. рис. 26). Для записи информации в другие блоки, которые могут быть подключены к любой секции магистрали, необходимо распространить информацию на все три секции к моменту действия сигнала записи, подаваемого в соответствующий блок по серии С2. Это осуществляется подачей сигнала перезаписи ПЗП, через вход 14 на элементы И-НЕ 11, 12 и 13. Они передают маскированную информацию на секции 1, 2 и 3. Каждый из этих элементов в паре с элементом 9 образует на время действия сигнала ПЗП триггер, в который записывается информация с любой секции магистрали.

Перед следующим тактом работы процессора магистрали должны быть сброшены, т. е. на них должен быть восстановлен единичный потенциал. По окончании процесса записи информации в блоки сигнал ПЗП снимается и подается сигнал СБРМ на вход 4. При этом RS-триггер на элементах 6 и 7 сбрасывается и на секции магистрали подается единичный потенциал.

Для ускорения процесса восстановления единичного состояния магистралей (см. рис. 25) применяется форсирующая схема. Она состоит из резисторов $R4—R6$ и мощного инвертора 5. Так как на выходе инвертора, находящегося в состоянии логического нуля, напряжения примерно 0,2 В, то пары резисторов $R1, R4$; $R2, R5$; и $R3, R6$, образующие делители напряжения, поддерживают на секциях магистрали напряжение, близкое к 3,5 В. При подаче сигнала СБРМ на выходе инвертора 5 образуется положительный потенциал, и резисторы $R4, R5$ и $R6$ оказываются включенными параллельно резисторам $R1, R2$ и $R3$ в цепь источника положительного напряжения. Зарядный ток значительно возрастает, что способствует форсированному заряду распределенной паразитной емкости секций магистрали и быстрому восстановлению исходного единичного состояния.

Если нужно сохранить состояние магистрали более чем на один такт, сигнал СБРМ не вырабатывается, а длительность сигнала ПЗП увеличивается.

Магистральный коммутатор построен на ТЭЗ двух типов: коммутатор магистральный 1 (КМ1) и коммутатор магистральный

2 (КМ2). КМ1 содержит четыре информационных разряда магистрали, КМ2 — по одному контрольному разряду магистралей М1, М2 и М3. Контрольным разрядом снабжаются каждые 8 разрядов магистрали.

Узел контроля БМК осуществляет контроль информации, считанной на магистраль (контроль числа на магистрали); контроль маски, подаваемой на нее, и формирование контрольных бит отмаскированной на магистралях информации.

Контроль информации, считанной на магистраль, производится побайтно. Значение каждой пары разрядов (М1 [0] и М1 [1] или М1 [2] и М1 [3]) сворачивается по mod2 и запоминается в отдель-

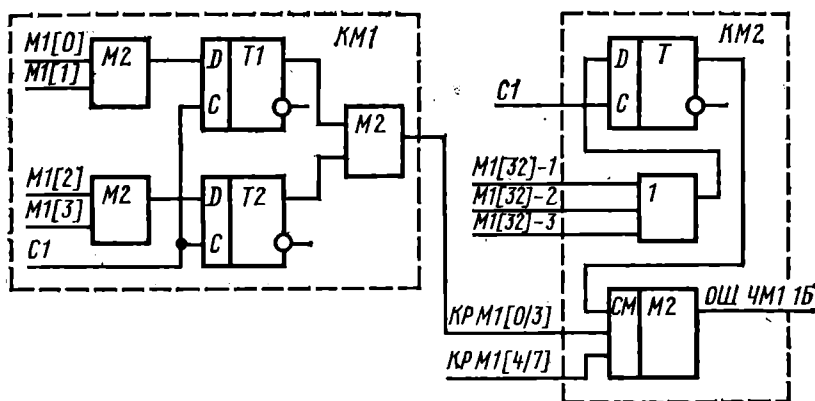


Рис. 27. Контроль числа на магистрали

ном D-триггере (Т1 или Т2) по синхросерии С1 (рис. 27). Это необходимо для того, чтобы как можно быстрее запомнить значение контрольных разрядов принятой на магистраль информации, так как после окончания синхросерии С1 информация на магистрали может измениться. Это случается, например, в результате маскирования. Контрольные биты двух тетрад одного байта поступают на схему формирования сигнала ошибки, которая представляет собой одноразрядный двоичный сумматор СМ. На третий вход сумматора поступает значение контрольного разряда байта числа, принятого на магистраль. Оно поступает по одной из секций магистрали (на эту секцию было считано число из какого-либо функционального блока ЦП) и запоминается в D-триггере Т контрольного разряда также по синхросерии С1. Рассмотренная схема контроля вырабатывает сигнал ошибки первого байта числа, находящегося на первой магистрали ОШ ЧМ1 1Б (см. табл. 35). Контроль числа байтов всех трех магистралей построен одинаково. Контрольные схемы расположены в ТЭЗ КМ2.

Контроль маски на магистрали (рис. 28) позволяет определить правильность считывания константы из БОПЗ в РМС1, РМС2 или на любую из магистралей. Контроль на одной магистрали производится побайтно в восьми ТЭЗ КМ1 и четырех ТЭЗ «Регистра

маски» РМС. Контрольный бит маски поступает на свертку либо из РМКЗ, либо из одного из D-триггеров Т1 (Т2) по микроприказу чтения маски $M2 := РМС1$ ($M2 := РМС2$). В эти триггеры контрольный бит записывается при записи маски в РМС1 (РМС2) по микроприказу $РМС1 := КОНСТ$ ($РМС2 := КОНСТ$). Схема контроля вырабатывает сигнал ошибки первого байта маски на второй магистрали ОШМ М2 1Б. Контроль всех байт всех трех магистралей построен аналогично.

Значение контрольного разряда байта отмаскированной информации формируется путем сворачивания байта по mod2 и по синхросерии С23П распространяется на три секции магистрали вместе с самой отмаскированной информацией.

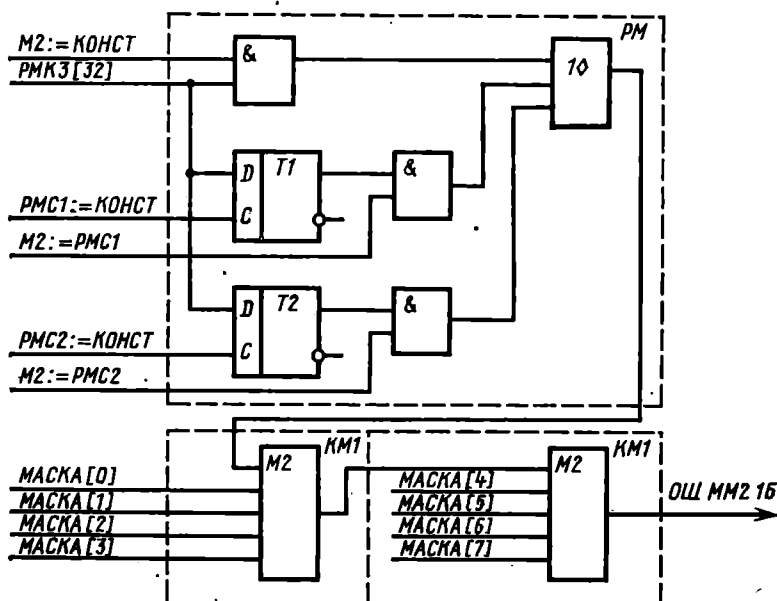


Рис. 28. Контроль маски на магистрали

Схема байтного обнуления предназначена для того, чтобы обнулить путем наложения специальной маски одну или сразу несколько 8-разрядных групп магистрали. Этот прием применяется при обработке десятичных операндов и при логической обработке данных переменной длины [А. с. 648984 (СССР)].

Данные этого типа могут располагаться в оперативной памяти, начиная с любого байта слова ОП, и заканчиваться также любым байтом. В ЭВМ ЕС-1033 выборка из ОП всегда производится целым словом, а это значит, что вместе с первыми и последними байтами операнда в выбранном слове ОП может содержаться посторонняя, не связанная с операндом, информация. Для удаления с помощью байтной маски посторонней информации используются байтные маски нескольких классов (табл. 6). Вид маски внутри каждого из классов определяется адресами (кодами позиции) начального или конечного байт операнда. Эти коды являются составной частью адресов названных байт.

К нулевому классу относятся маски, с помощью которых можно обнулить байты слева от байта с заданным адресом. Эти маски используются для того, чтобы освободить поступивший на обработку операнд от посторонней информации, расположенной перед его первым байтом. Если операнд начинается с байта с адресом 00, т. е. с границы слова, то обнулять ничего не надо и в этом случае накладывается маска FFFF FFFF. Если операнд начинается с байта с адресом 01, то следует удалить нулевой байт, наложив на операнд маску 00FF FFFF, и т. д.

Маски второго класса предназначаются для обнуления байт справа от байта с заданным адресом. Эти маски используются для удаления ненужной информации справа от последнего байта операнда.

Маски первого класса позволяют выделить информацию, расположенную левее байта с заданным адресом. С помощью масок этого класса не устраняется лишняя информация, а выделяется необходимая (п. 9.2).

Если строить микропрограмму операций обработки данных переменной длины так, что при обработке, например, первого слова операнда, в микропрограмме анализируется код позиции первого байта операнда в слове, то далее микропрограмма должна разветвиться на четыре направления. В первой ветви (код позиции байта 00) наложения маски не требуется, в остальных трех накладывается соответствующая маска. В противоположность этому при использовании байтных масок обработка выполняется по единой микропрограмме. Класс маски задается в микропрограмме, а код позиции байта, являющейся функцией адреса, определяет конкретный вид требуемой байтной маски. В результате сама информация программирует свою обработку. Этот прием значительно снижает объем микропрограмм. Код класса маски и адрес байта помещается в регистр формирования маски. По этой информации комбинационные схемы формируют соответствующие сигналы обнуления байт. По микроприказам $M2 := \text{МАСКА}$ и $M3 := \text{МАСКА}$ производится наложение байтных масок на магистрали.

Поскольку «очистка» операндов выполняется перед их записью в сдвигатель, подключенный ко 2-й и 3-й магистралям, байтное

Т а б л и ц а 6

Классификация байтных масок

| Класс маски | Управляющая информация | | Вид байтных масок |
|-------------|------------------------|-------------|-------------------|
| | код класса | адрес байта | |
| 0 | 00 | 00 | FFFF FFFF |
| | 00 | 01 | 00FF FFFF |
| | 00 | 10 | 0000 FFFF |
| | 00 | 11 | 0000 00FF |
| 1 | 01 | 00 | 0000 0000 |
| | 01 | 01 | FF00 0000 |
| | 01 | 10 | FFFF 0000 |
| | 01 | 11 | FFFF FF00 |
| 2 | 10 | 00 | FF00 0000 |
| | 10 | 01 | FFFF 0000 |
| | 10 | 10 | FFFF FF00 |
| | 10 | 11 | FFFF FFFF |

обнуление распространяется только на M2 и M3. Кроме того, по микроприказу $M2 := 0$ можно наложить на M2 нулевую маску, которая фактически используется в вычислениях как константа «нуль».

2.3. МЕСТНАЯ ПАМЯТЬ

В состав местной памяти входят блоки программных и операционных регистров БРП и БРОП.

Первый состоит из 16 регистров общего назначения и 4 регистров для операндов с плавающей запятой (РПЗ). Адреса этих регистров указываются непосредственно в командах программы. Второй включает 16 операционных регистров, предназначенных для запоминания промежуточных результатов вычислений, промежуточного (буферного) хранения исходных операндов, адресов и т. п. Быстродействие местной памяти и ее организация во многом определяют быстродействие самого процессора.

Структурная организация ЭВМ ЕС ориентирована на интенсивное использование программных регистров. Из 143 команд машин семейства «Ряд 1» 95 (команды форматов RX и RR) предполагают использование хотя бы одного операнда, расположенного в РОН или РПЗ. В 49 командах формата RR оба исходных операнда извлекаются из указанных регистров, а результат помещается в один из них, как правило, на место первого операнда. При выполнении команд формата RR, таких, как сложение, вычитание, сравнение и т. д., обращение в оперативную память производится только за командным словом (2 байта), а все пути прохождения информации замыкаются в центральном процессоре. Эти команды выполняются менее, чем за одну микросекунду, что обеспечивает быстродействие на указанном формате более миллиона операций в секунду.

Имеется также группа команд для выполнения действий над десятичными числами и обработки данных переменной длины, в которых исходные данные берутся из ОП, в нее же помещаются и результаты обработки (команды формата SS). Однако и в этих командах используются регистры общего назначения, так как система адресации ЕС ЭВМ предусматривает косвенную и относительную адресацию ячеек ОП, когда компоненты адреса хранятся в РОН. В командах форматов RX, RS, SS адрес операнда указан не непосредственно, а является суммой смещения, содержимого индексного регистра и регистра базы. В командном слове указано непосредственно только смещение, а индекс и база адресованы номерами общих регистров, где эти данные расположены. Таким образом, и в тех командах, где операнды расположены в ОП, подразумевается работа программных регистров.

Блок программных регистров представляет собой законченный функциональный блок, подключенный своими входами и выходами к системе информационных магистралей ЦП [А. с. 613402 (СССР)]. На рис. 29 показаны адресные и информационные связи БРП при

выполнении команд формата RR. Содержимое полей $R1$ и $R2$ команды (адреса операндов) помещается на этапе выборки команды в счетчики $CT1$ POH и $CT2$ POH , которые являются регистрами адресов операндов. По микроприказам $M2:=POHCT1$ и $M3:=POHCT2$ содержимое соответствующих программных регистров, номера которых определяются содержимым счетчиков, поступает на информационные магистрали ЦП, к которым подключены входы арифметико-логического блока. Комбинационная схема АЛУ через 100—120 нс выдает на магистраль $M1$ результат операции, который по микроприказу $POHCT1:=M1$ записывается в регистр, адресованный первым счетчиком, т. е. на место первого

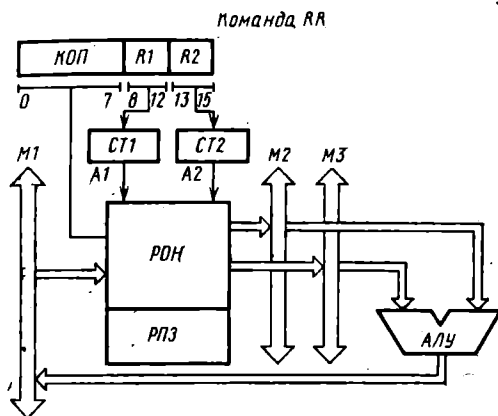


Рис. 29. Адресные и информационные связи программных регистров при выполнении команд формата RR

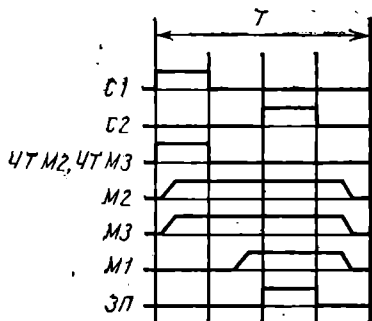


Рис. 30. Обмен информацией через магистрали

операнда. Временная диаграмма процесса показана на рис. 30. Здесь $C1$ и $C2$ — основные синхросерии; $ЧТМ2$, $ЧТМ3$ — микроприказы чтения на магистрали; $M2$, $M3$ — информация из регистров, прочитанная на магистрали; $M1$ — результат обработки, выданный на $M1$; $ЗП$ — микроприказ записи результата. Как видно из рисунка, гонки фронтов не возникает, так как магистрали хранят прочитанную на них информацию до конца такта. Другие действия с содержимым POH , например, суммирование базы и индекса, пересылки, загрузки и т. п., выполняются аналогично.

Использование счетчиков в качестве адресных регистров POH ($РПЗ$) не случайно. Весьма часто в командах задается адрес соответствующего регистра неявно. Например, адресом частного при делении является номер регистра, на единицу больший, чем указанный в команде адрес регистра делимого, и т. п. Счетчиками обеспечивается также простое выполнение команд массовой записи и загрузки, когда по одной команде пересылается содержимое нескольких подряд расположенных регистров, адресуемых полями $R1$ и $R2$ команды.

Информационные связи в процессоре рассчитаны на параллельную передачу 32-разрядных кодов (слов). Такую же разрядность

имеют и регистры общего назначения. Регистры для операндов с плавающей запятой с номерами 0, 2, 4 и 6 имеют двойную длину, т. е. 64 разряда. Чтобы извлечь для обработки полное содержимое регистра с плавающей запятой, необходимо обратиться в блок дважды: за старшим словом и, продвинув адрес регистра в счетчике на единицу, — за младшим.

РОН и РПЗ адресуются одними и теми же счетчиками. По коду операции определяется, с каким из типов регистров надлежит работать.

Для организации одновременного считывания информации на два направления, в том

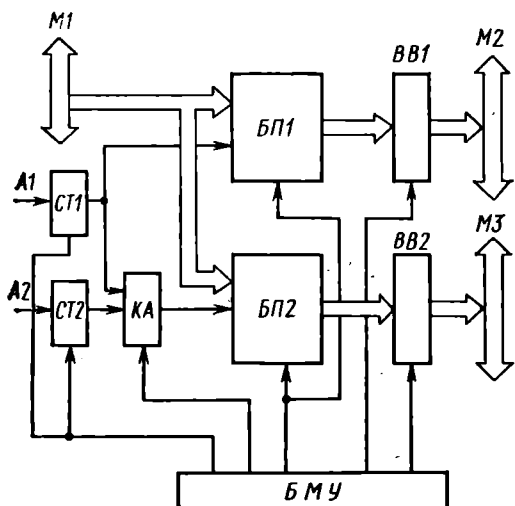


Рис. 31. Структурная схема блока программных регистров

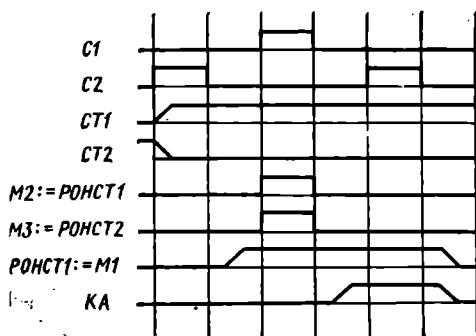


Рис. 32. Временная диаграмма работы программных регистров

числе и из одного и того же регистра, необходимо, чтобы каждый запоминающий элемент имел пару независимо управляемых выходных вентилях. Существующие микросхемы, кроме К155ХЛ1, не обладают такими возможностями. Поэтому противоречие между требованиями компактности (применение ИС К155ХЛ1 потребовало бы для реализации блока 24 ТЭЗ) и функциональными требованиями было решено дублированием блока памяти программных регистров, выполненного на ИС К155РУ2 (п. 1.5). Регистры общего назначения состоят из двух блоков памяти БП1 и БП2 (рис. 31), каждый из которых содержит по 16 32-разрядных регистров и дешифраторы 4 × 16. Информационные входы обоих блоков памяти подключены к первой магистрали ЦП параллельно. Адресация регистров в первом блоке производится от первого счетчика СТ1 РОН. Адресация второго блока выполняется через коммутатор адреса либо от первого, либо от второго счетчиков СТ1 РОН и СТ2 РОН.

Выходы первого и второго блоков памяти через выходные вентиля ВВ1 и ВВ2 подключены ко второй и третьей магистралям процессора соответственно. Управление осуществляется блоком микропрограммного управления (БМУ). Устройство регистров плавающей запятой аналогично.

Запись информации в программные регистры всегда происходит с магистрали М1 по микроприказу $РОНСТ1 := M1$. Адрес регистра, в который происходит запись, определяется содержимым счетчика СТ1 РОН. При этом коммутатор адреса (КА) подключает СТ1 РОН к дешифраторам адреса обоих блоков БП1 и БП2, в результате чего запись в них происходит по одному и тому же адресу. В обычном состоянии СТ2 подключен к БП2, а оба блока памяти находятся в режиме постоянного чтения, т. е. информация из них выдается всегда, когда отсутствует запись. При этом из БП1 читается информация из регистра, номер которого хранится в СТ1, а из БП2 — из регистра, номер которого хранится в СТ2.

Чтение информации из БРП на два направления и запись информации в него в одном и том же такте происходит, например при выполнении операции сложения (п. 6.1), следующим образом.

По микроприказам $M2 := РОНСТ1$ и $M3 := РОНСТ2$ выходные вентили ВВ1 и ВВ2 подключают выходы БП1 и БП2 к магистралям М2 и М3 на время действия синхроимпульса С1 (рис. 32). Обычно информация, считанная на М2 и М3, поступает в БАЛ для обработки, а результат выдается на М1. По микроприказу $РОНСТ1 := M1$ этого же такта в промежутке между синхросериями С1 и С2 коммутатор адреса подключает первый счетчик СТ1 к БП2 и запись с М1 происходит в БП1 и БП2 одновременно по одному и тому же адресу в течение времени действия синхроимпульса С2. Таким образом содержимое обоих блоков памяти всегда идентично. При построении блоков на микросхемах с одним выходом наличие второго блока памяти позволяет считывать информацию независимо на два направления. Использование ИС К155РУ2 позволило разместить программные регистры в пяти ТЭЗ «РОН». Счетчики РОН размещены в двух ТЭЗ СТ.

Регистры РОН (РПЗ) не имеют собственных схем контроля и контролируются при чтении из них информации на магистраль. Счетчики РОН имеют автономный контроль, который проверяет правильность приема информации в счетчик и правильность модификации его содержимого. Счетчики также контролируются при чтении информации из них на магистраль.

Блок операционных регистров выполнен с учетом возможности максимального использования преимуществ магистральной организации ЦП ЭВМ ЕС-1033 [А. с. 877613 (СССР)]. Все 16 32-рядных регистров блока построены на ячейках многофункциональной памяти — микросхемах К155ХЛ1 и подключены тремя своими входами-выходами к трем основным магистралям ЦП.

Управление работой БРОП осуществляется блоком микропрограммного управления в соответствии с алгоритмами микропрограмм. Адреса чтения содержимого операционных регистров на магистрали и записи информации с магистралей в регистры вырабатываются адресными дешифраторами, которые выполняют также функции коммутаторов. Исходные адреса задаются либо полями М1 ЧТ, М1 ЗП, М2 ЧТ, М2 ЗП, М3 ЧТ, М3 ЗП микрокоманды,

либо поступают из четырехразрядных счетчиков циклов *СТЦ1* и *СТЦ2* (рис. 33). Шестиразрядный управляющий регистр *ТРА5* (табличный регистр анализа, п. 3.5) позиционно определяет режим работы дешифраторов. Запись кода в *ТРА5* происходит путем записи соответствующей константы из поля микрокоманды.

В первом полутакте работы ЦП выполняется считывание содержимого операционных регистров, адресуемых полями *РМК0*: *М1 ЧТ*, *М2 ЧТ* и *М3 ЧТ* (п. 3.5) на соответствующие магистрали.

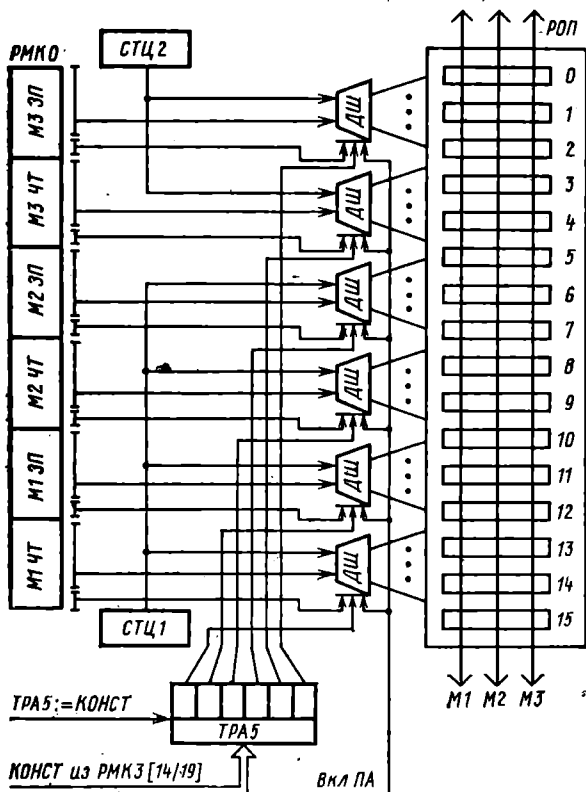


Рис. 33. Организация управления операционными регистрами

Во втором полутакте с магистралью производится запись в операционные регистры, заданные полями *М1 ЗП*, *М2 ЗП* и *М3 ЗП* соответственно. Четыре разряда каждого поля определяют номер регистра с 0 по 15, а наличие единицы в пятом свидетельствует о том, что задается адрес регистра РОП. Нулевое состояние пятого разряда поля блокирует выполнение действий регистра в данном полутакте с соответствующей магистралью ЦП, а код поля в этом случае является кодом какого-либо микроприказа. Адрес РОП может быть задан также счетчиками.

Благодаря высокой функциональной гибкости операционные регистры интенсивно используются при работе ЦП. Уже на этапе выборки команд (п. 4.4) и операндов в них закладывается необходимая для выполнения операции информация. Стандартно во всех пяти форматах команд в РОП0 засылается константа «ноль» (табл. 21, 22). Таким образом, запись нулевого результата, очистка какого-либо регистра или ячейки памяти осуществляется в процессе выполнения микрокоманды простой пересылкой содержимого РОП с нулевым номером. В РОП1 и РОП3 обычно расположено командное слово. При необходимости оно может быть использовано и в ходе выполнения микропрограммы. В командах формата *SS* РОП14 отведен под адрес первого операнда в ОП, а при выполнении команд формата *SI* его адрес засылается в РОП15. В командах других форматов (*RX*, *RS*, *SS*) РОП15 со-

держит адрес второго операнда. Для связи микропрограммы обработки прерываний с прочими микропрограммами в качестве ячейки связи используется РОП10. Если при выполнении какой-либо микропрограммы происходит прерывание, в этот регистр записывается константа, отражающая код прерывания, и управление передается микропрограмме обработки прерываний. Последняя стандартно переписывает код прерывания из РОП10 в старое слово состояния программы (ССП), записывает его в ОП и извлекает новое ССП. Таким образом, единая микропрограмма обработки прерываний обслуживает все программные прерывания. При работе с другими микропрограммами в качестве ячеек связи используются другие РОП в соответствии с соглашением между разработчиками микропрограмм.

Особенно интенсивно операционные регистры используются при обработке операндов в десятичных и логических операциях (гл. 8, 9). Интересно использование РОП в командах УПАКОВАТЬ и РАСПАКОВАТЬ (п. 9.4), где поля исходного операнда и результата обработки могут перекрываться в оперативной памяти, а операция должна выполняться таким образом, как будто обработка производится побайтно. В связи с этим возможна такая ситуация, когда очередное слово результата будет воспринято как исходный операнд и обработано еще раз, а результат повторной обработки вновь воспринят как исходный операнд и т. д. В ЕС-1033 при словном доступе в память, цикл работы которой к тому же в четыре раза больше цикла ЦП, переход на побайтную обработку с обращением в ОП за каждым байтом существенно снижает быстродействие.

Использование операционных регистров устраняет эти ограничения. Сначала слова операндов загружаются в группу РОП, а потом выполняется побайтная обработка информации по тем же принципам, как если бы она находилась в ОП, но уже в темпе работы ЦП. Таким образом, регистрами РОП моделируется участок ОП, работающий в сверхоперативном режиме (время доступа 30 вместо 1600 нс в ОП).

Режим переадресации РОП, как метод управления чтением и записью РОП, применен в ЕС-1033 для удобства микропрограммирования операций обработки данных переменной длины. Сущность его заключается в том, что адреса регистров задаются не полями микрокоманды, а адресными регистрами, в качестве которых используются счетчики циклов СТЦ1 и СТЦ2. Предположим, что необходимо сложить два операнда, каждый длиной в четыре слова, расположенных в РОП0—РОП3 и РОП4—РОП7 с помощью АЛУ, ориентированного на пословную обработку. Очевидно, что микропрограмма будет состоять из четырех последовательных участков (сложить РОП0 с РОП4, РОП1 с РОП5 и т. п.), отличающихся только адресами РОП. Усложним задачу: каждый операнд может быть длиной от 1 до 16 байт. Очевидно, что при непосредственной адресации РОП полями микрокоманды необходимо на-

писать столько вариантов микропрограммы, сколько сочетаний для операндов может встретиться.

В ЦП ФС-1033 в подобных ситуациях адресация РОП производится посредством счетчиков, в которые предварительно заносится начальное значение, зависящее от длин операндов. Микропрограмма имеет циклический характер и завершается, когда операнды будут полностью обработаны. С каждым циклом содержимое счетчиков увеличивается на единицу, и та же самая микрокоманда в другом цикле будет работать уже с другими РОП. Указанный прием значительно снижает объем микропрограмм и упрощает их построение.

Включение режима переадресации производится по микроприказу ВКЛ ПА, который блокирует поступление адреса РОП из полей РМК0 и подключает к дешифраторам соответствующие счетчики в качестве адресных регистров РОП (см. рис. 33). Адресация РОП при записи и чтении с М1 и М2 происходит по содержимому счетчика циклов СТЦ1, а адресация РОП при записи и чтении с М3 — по содержимому СТЦ2. Поскольку содержимое полей ЗП и ЧТ микрокоманды игнорируется, то при переадресации режим записи или чтения на определенные магистрали задается специальным шестиразрядным регистром признаков записи—чтения ТРА5 (п. 3.5). Наличие единицы в каком-либо разряде ТРА5 означает, что должна производиться запись (или чтение) с соответствующей магистрали. Например, ТРА5 = 00 0011 означает, что в такте, где имеется микроприказ ВКЛ ПА, содержимое полей микрокоманды как адрес РОП не рассматривается. В этом случае должно быть произведено чтение на М3 содержимого РОП, номер которого хранится в СТЦ2, и запись с М3 в этот же РОП. ТРА5 = 00 0101 означает, что информация, считанная на М2 и М3 из какого-то другого блока (допустим, сдвинутая информация из сдвигателя), должна быть записана в два РОП. При записи с М2 номер РОП определяется содержимым СТЦ1, при записи с М3 — СТЦ2. Подробно использование режима переадресации рассмотрено в гл. 8 и 9.

Блок операционных регистров выполнен на 16 ТЭЗ РОП, трех ТЭЗ УРОП (управление регистрами операционными) и двух ТЭЗ СТ. Контроль БРОП построен аналогично контролю БРП.

Счетчики СТ РОП и СТЦ имеют автономные схемы контроля (п. 10.1). Они обеспечивают проверку правильности записи информации с магистралей и из зоны константы РМК3 (п. 3.5), позволяют контролировать правильность модификации содержимого счетчиков, осуществляемой по микроприказам СТ := + 1 и СТ := - 1. Кроме того, счетчики могут контролироваться при чтении их содержимого на магистраль. Запись константы в СТ происходит по синхросерии С2, а модификация содержимого — по С1. Поэтому в такте записи константы прием в РМО сигналов ошибок от схем контроля счетчиков в текущем такте запрещен сигналом блокировки и осуществляется лишь в следующем такте

(п. 10.2). В текущем такте разрешен только прием сигналов ошибок, возникших при модификации содержимого счетчика.

При записи информации в счетчик контрольный бит информации записывается в основной триггер Т1 контрольного разряда

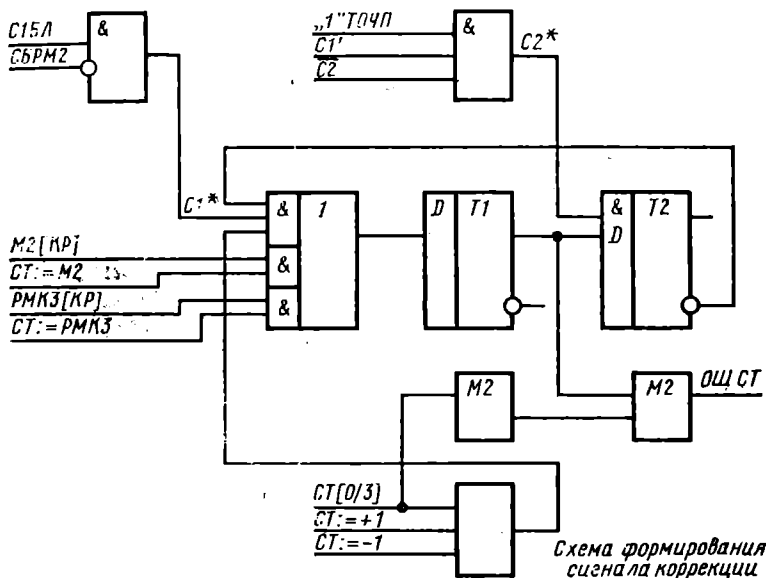


Рис. 34. Функциональная схема контроля счетчика

(рис. 34, 35). При этом запись информации с магистрали М2 по микроприказу $CТ := M$ сопровождается записью в триггер Т1 значения контрольного разряда М [КР] с магистрали. В случае записи константы из РМКЗ — значение контрольного разряда РМКЗ [КР] поступает в Т1 из соответствующей зоны РМК (п. 3.5). Запись информации в СТ происходит по С2. После того как информация в счетчике примет установившееся значение, по вспомогательной серии С2* значение контрольного разряда из основного триггера Т1 переписывается в буферный триггер Т2. Принятая информация свертывается по mod 2, и полученное значение сравнивается со значением контрольного разряда, хранящимся в триггере Т1. В случае несравнения формируется сигнал ошибки счетчика ОШ СТ. Прием этого сигнала в РМО в текущем такте блокируется сигналом блокировки. Сигнал ошибки принимается по синхросерии С2'БЛ лишь в следующем такте.

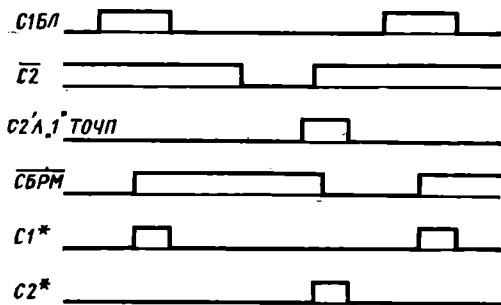


Рис. 35. Формирование блокируемых синхросерий С1* и С2*

При модификации содержимого счетчика, которая происходит по серии С1, схема формирования сигнала коррекции контроль-

ного разряда вырабатывает корректирующий сигнал, если значение контрольного разряда в процессе модификации содержимого счетчика должно измениться. Эта схема реализует предсказание четности результата в соответствии со следующими соотношениями:

$$C_{\text{кор}} = \bar{a}_0 \vee a_1 \bar{a}_2$$

в случае модификации на +1 и

$$C_{\text{кор}} = a_0 \vee \bar{a}_1 a_2$$

в случае модификации на -1,

где $C_{\text{кор}}$ — сигнал коррекции; a_0, a_1, a_2 — значения младших разрядов счетчика до модификации (a_0 — самый младший разряд). Например, при переходе значения счетчика от 0100 к 0101 $C_{\text{кор}} = 1$, а при переходе значения от 0101 к 0110 $C_{\text{кор}} = 0$.

После того как по сигналу модификации содержимое счетчика изменится и одновременно с этим будет выработан сигнал коррекции, по синхросерии $C2^*$ (см. рис. 35) инверсное значение контрольного разряда из буферного триггера Т2 будет переписано в основной триггер Т1. Значение контрольного разряда, снимаемое с выхода Т1, сравнивается с вновь сформированным. Ошибка, если она возникла, фиксируется в РМО по синхросерии $C2'БЛ$ этого же такта.

Схемы контроля расположены в ТЭЗ СТ. Контроль счетчиков СТ РОИ и СТЦ организован одинаково.

2.4. АРИФМЕТИКО-ЛОГИЧЕСКИЙ БЛОК

Центральный процессор выполняет команды и обрабатывает данные, представленные в различных форматах и системах счисления. Наибольшая нагрузка при вычислениях ложится на арифметико-логический блок [38].

Этот блок предназначен для логической и арифметической обработки двоичных данных, представленных в форме чисел с фиксированной и плавающей запятой, двоично-кодированных десятичных и символьных данных. В ходе обработки информации БАЛ формирует признак результата (код условия) и выдает его для записи в разряды 34, 35 регистра слова состояния программы. При обработке десятичных данных на входах БАЛ проверяется правильность исходных кодов чисел. Блок позволяет осуществлять параллельную обработку 32-разрядных операндов, которые могут быть либо двоичными, либо двоично-десятичными числами, содержащими по 8 двоично-кодированных десятичных цифр. Управление работой БАЛ микропрограммное. Блок состоит из восьми ТЭЗ «АЛУ», двух ТЭЗ «Переносы АЛУ» и одного ТЭЗ «Ноль сумматора».

Использование БАЛ в процессоре с магистральной структурой определило ряд его особенностей. Так, наличие в магистральных коммутаторах элементов памяти, запоминающих и сохраняющих

в течение такта работы поступившую в них информацию, обеспечивает работу комбинационных схем БАЛ без традиционных входных регистров. Возможности ключевых схем входов и выходов операционных регистров, регистров общего назначения и регистров плавающей запятой, допускающих обращение одновременно по трем адресам (например, два для чтения, один для записи), позволяют выдавать результат обработки информации в БАЛ непосредственно на магистраль М1 без промежуточных коммутаторов и

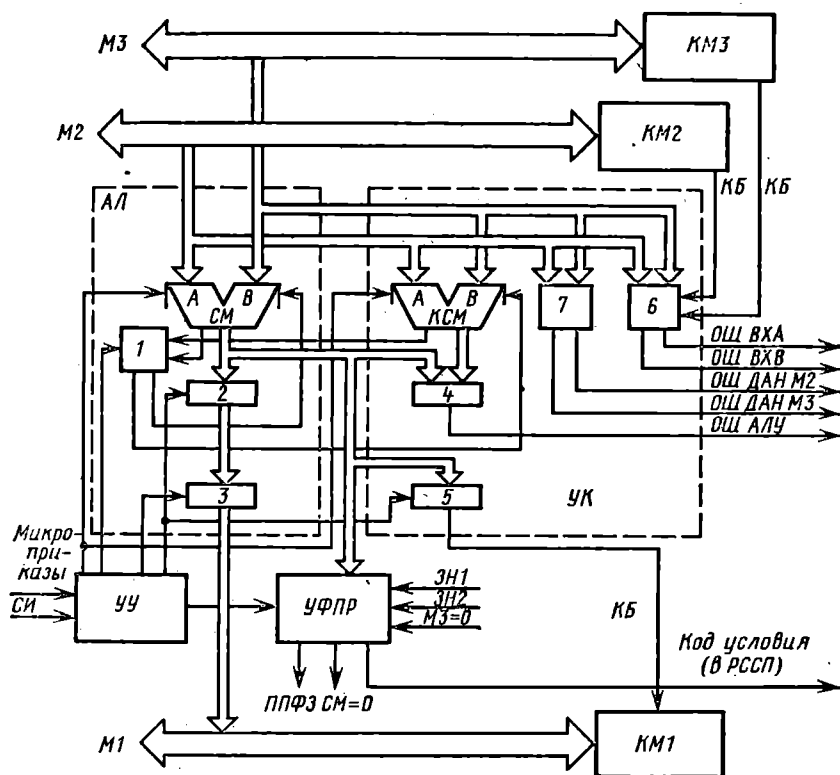


Рис. 36. Структурная схема БАЛ

применения вспомогательного выходного регистра — аккумулятора. Отсутствие традиционных схем сдвига выходной информации обусловлено наличием в процессоре отдельного быстродействующего блока сдвига.

В состав БАЛ входят (рис. 36) узел арифметико-логический (АЛ), узел контроля (УК), узел управления (УУ), узел формирования признака результата (УФПР).

Информация, подлежащая обработке, поступает на входы БАЛ со второй (первый операнд) и с третьей (второй операнд) информационных магистралей ЦП. Преобразование информации осуществляется комбинационными схемами БАЛ. Результат выдается на первую информационную магистраль не позднее чем через 120 нс после поступления входных операндов.

Таблица 7

Операции обработки, выполняемые БАЛ

| Код поля S РМК1 | Логические, M = 1 | Арифметические, M=0 | |
|--------------------|--------------------------------|----------------------------|-------------------------|
| | | ПСМ=0 | ПСМ=1 |
| 0000 | \bar{a}_i | $A-1$ | A |
| 0001 | $\overline{a_i b_i}$ | $AB-1$ | AB |
| 0010 | $\bar{a}_i \vee b_i$ | $A\bar{B}-1$ | AB |
| 0011 | 0 | -1 (в дополнительном коде) | 0 |
| 0100 | $\overline{a_i \vee b_i}$ | $A+(A\vee\bar{B})$ | $A+(A\vee\bar{B})+1$ |
| 0101 | b_i | $AB++(A\vee\bar{B})$ | $AB++(A\vee\bar{B})+1$ |
| 0110 | $\overline{a_i \oplus b_i}$ | $A-B-1$ | $A-B$ |
| | | $(A-B-1)**$ | $(A-B)**$ |
| 0111 | $a_i \vee b_i$ | $A\vee\bar{B}$ | $(A\vee\bar{B})+1$ |
| 1000 | $\bar{a}_i b_i$ | $A+(A\vee B)$ | $A+(A\vee B)+1$ |
| 1001 | $a_i \oplus b_i$ | $A+B$ | $A+B+1$ |
| | | $(A+B)**$ | $(A+B+1)**$ |
| 1010 | b_i | $A\bar{B}++(A\vee B)$ | $A\bar{B}++(A\vee B)+1$ |
| 1011 | $a_i \vee b_i$ | $A\vee B$ | $(A\vee B)+1$ |
| 1100 | 0 | $A+A*$ | $A+A*+1$ |
| 1101 | Запрет выдачи результата на M1 | | |
| 1110 | $a_i b_i$ | $A\bar{B}+A$ | $A\bar{B}+A+1$ |
| 1111 | a_i | A | $A+1$ |

Примечание. $A = a_0, a_1, \dots, a_{31}$; $B = b_0, b_1, \dots, b_{31}$.

* Сдвиг на один разряд влево.
** Операции над десятичными данными.

Узел АЛ состоит из: многофункционального сумматора - преобразователя СМ;

схемы 1 установки входного переноса сумматора (ПСМ) и хранения переноса из старшего разряда сумматора;

схемы 2 десятичной коррекции;

вентильной схемы 3 выдачи результата на M1.

Сумматор-преобразователь СМ осуществляет непосредственные преобразования 32-разрядных операндов, поступающих на его входы. СМ выполнен на восьми четырехразрядных интегральных микросхемах К155ИПЗ повышенной степени интеграции (п. 1.5). Вид обработки (арифметический или логический) и выполняемая функция задаются микропрограммно полями M и S РМК1 (п. 3.5). Значение одноразрядного поля M поступает на входы M, а четыре разряда поля S — на входы S0—S3 всех микросхем, составляющих СМ. Значение входного переноса СМ задается специальным микроприказом для каждого такта работы. Перечень операций, выполняемых БАЛ, приведен в табл. 7.

Для повышения быстродействия СМ содержит также в своем составе схему полного предварительного просмотра и генерации межгрупповых переносов. Эта схема

обеспечивает подачу входных переносов в каждую ИС К155ИПЗ не позже, чем через 30 нс после выработки микросхемой на выходах Р и G признаков этих переносов.

Схема установки и хранения переноса обеспечивает подачу на вход сумматора заданного микропрограммой значения ПСМ, а также хранение выходного переноса из сумматора, полученного в результате операции. Последнее необходимо, когда в БАЛ обрабатываются операнды длиной более 32 разрядов, например, 64-разрядные числа с плавающей запятой, десятичные операнды, размещающиеся в полях, длина которых превышает формат слова и т. п.

Как следует из табл. 7, БАЛ обеспечивает также сложение и вычитание операндов, представленных в двоично-кодированном десятичном коде. Корректировка результата осуществляется по управляющему сигналу коррекции «+10» и выполняется схемой 2 десятичной коррекции. Необходимость этого объясняется тем, что при десятичном сложении один из операндов (безразлично который) должен быть представлен в коде «с избытком 6». Это означает, что к каждой десятичной цифре должно быть заранее прибавлено число 6. Внесение избытка необходимо для организации переноса в старшую десятичную цифру, если сумма в соседней младшей превысит число 9. Если же сумма не превышает 9, то результат в данной тетраде получается на 6 больше истинного. Схема десятичной коррекции обеспечивает вычитание числа 6 (или прибавление числа 10, что эквивалентно) из тех десятичных цифр результата, из которых не было переноса. Корректировка производится непосредственно после получения результата на выходе сумматора в том же такте обработки. При вычитании десятичных чисел коррекция какой-либо цифры производится при наличии заема из соседней старшей цифры также вычитанием числа 6. Во всех случаях коррекция выполняется логическими схемами без применения дополнительных сумматоров. При обработке двоичных данных схема десятичной коррекции не оказывает влияния на получаемый результат.

С выхода узла АЛ результат поступает на магистраль М1 через вентиляционную схему 3. Так как сумматор комбинационный, то на его выходе всегда присутствует результат обработки чисел, находящихся на М2 и М3. Одной из комбинаций разрядов управляющей функции ($S = D$) запрещается подключение выхода АЛ к М1. При этом признак результата операции все же вырабатывается в РССП. Такой режим используется в операциях сравнения, где нужен только признак результата. При выполнении сравнения магистраль М1 свободна для пересылок информации и используется для реализации какой-либо параллельной ветви алгоритма. Сигнал запрета вырабатывается схемно также при наличии одного из микроприказов чтения на М1 содержимого РИП или РОП.

Признак результата операции вырабатывается при выполнении многих команд ЕС ЭВМ. Обычно он формируется совместно с результатом. Однако в ряде команд этот признак сам по себе является

единственным результатом (больше нуля, меньше нуля, равен нулю, который из операндов больше, было ли переполнение и т. д.) и используется в командах переходов. В БАЛ для выработки признака результата служит узел УФПР, подключенный непосредственно к выходу многофункционального сумматора и осуществляющий аппаратный анализ полученного результата. Этот узел представляет собой совокупность семи отдельных схем, каждая из которых вырабатывает признак для соответствующего типа команд (прил. 9). Выходы всех схем подключены к разрядам 34, 35 РССП. Активизация каждой схемы выполняется соответствующим микроприказом. Подключение УФПР непосредственно к выходу СМ позволяет сформировать признак результата одновременно с выдачей результата обработки информации в сумматоре. Вследствие этого многие команды форматов RR и RX, пройдя процесс декодирования и выборки операндов, выполняются и завершаются за один машинный такт (п. 6.1).

В табл. 8 приведены условия формирования признака результата для следующих семи типов команд:

команды арифметической обработки чисел с фиксированной запятой — признак результата арифметический (ПРА);

команды арифметической обработки чисел двойной точности с фиксированной запятой — признак результата арифметический, двойной точности (ПРАД);

команды арифметической обработки чисел с плавающей запятой — признак результата арифметических операций с плавающей запятой (ПРАП);

команды арифметической обработки чисел двойной точности с плавающей запятой — признак результата арифметических операций с плавающей запятой двойной точности (ПРАПД);

команды логической обработки — признак результата логических операций (ПРЛ);

команды арифметического сравнения — признак результата операций арифметического сравнения (ПРСРА);

команды сравнения кодов — признак результата операций логического сравнения (ПРСРЛ).

Кроме формирования признака результата УФПР вырабатывает сигнал о переполнении в операциях над числами с фиксированной запятой для перехода на обработку прерывания по переполнению (в операциях с плавающей запятой признак переполнения формируется микропрограммно). Вовне выдается также и сигнал о нулевом значении полученного результата для работы схем анализа блока микропрограммного управления.

БАЛ является одним из наиболее ответственных и интенсивно используемых блоков центрального процессора. Это предъявляет высокие требования к средствам обеспечения достоверности результатов. В БАЛ реализован сквозной непрерывный контроль приема, преобразования и выдачи информации.

Формирование признака результата

| Группа команд | Вид признака результата | Условия формирования | Код признака результата |
|---|-------------------------|--|-------------------------|
| Арифметическая обработка чисел с фиксированной запятой | ПРА | $(CM[0]=0) \wedge (CM[1/31]=0) \wedge$ $\wedge (ППФЗ=0)$ | 00 |
| | | $(CM[0]=1) \wedge (ППФЗ=0)$ | 01 |
| | | $(CM[0]=0) \wedge (CM[1/31] \neq 0) \wedge$ $\wedge (ППФЗ=0)$ | 10 |
| | | $ППФЗ=1$ | 11 |
| Арифметическая обработка чисел двойной точности с фиксированной запятой | ПРАД | $(CM[0]=0) \wedge (CM[1/31]=0) \wedge$ $\wedge (M3=0) \wedge (ППФЗ=0)$ | 00 |
| | | $(CM[0]=1) \wedge (ППФЗ=0)$ | 01 |
| | | $(CM[0]=0) \wedge (CM[1/31] \vee M3) \neq$ $\neq 0) \wedge (ППФЗ=0)$ | 10 |
| | | $ППФЗ=1$ | 11 |
| Команды логической обработки | ПРЛ | $(CM=0) \wedge (PCM=0)$ | 00 |
| | | $(CM \neq 0) \wedge (PCM=0)$ | 01 |
| | | $(CM=0) \wedge (PCM=1)$ | 10 |
| | | $(CM \neq 0) \wedge (PCM=1)$ | 11 |
| Команды арифметического сравнения | ПРСРА | $(CM=0) \wedge (PCM=1)$ | 00 |
| | | $(CM=0) \wedge [(PCM=0) \wedge$ $\wedge (3N1=3N2) \vee (PCM=1) \wedge$ $\wedge (3N1 \neq 3N2)]$ | 01 |
| | | $(CM \neq 0) \wedge [(PCM=1) \wedge$ $\wedge (3N1=3N2) \vee (PCM=0) \wedge$ $\wedge (3N1 \neq 3N2)]$ | 10 |
| Команды сравнения кодов | ПРСРЛ | $(CM=0) \wedge (PCM=1)$ | 00 |
| | | $(CM \neq 0) \wedge (PCM=0)$ | 01 |
| | | $(CM \neq 0) \wedge (PCM=1)$ | 10 |
| Арифметическая обработка чисел с плавающей запятой | ПРАП | $CM[8/31]=0$ | 00 |
| | | $(CM[0]=1) \wedge (CM[8/31] \neq 0)$ | 01 |
| | | $(CM[0]=0) \wedge (CM[8/31] \neq 0)$ | 10 |
| Арифметическая обработка чисел двойной точности с плавающей запятой | ПРАПД | $(CM[8/31]=0) \wedge (M3=0)$ | 00 |
| | | $(CM[0]=1) \wedge (CM[8/31] \vee M3) \neq 0$ | 01 |
| | | $(CM[0]=0) \wedge (CM[8/31] \vee M3) \neq 0$ | 10 |

Примечания: ЗН1, ЗН2 — знаки первого и второго операндов; ППФЗ — переполнение в операции с фиксированной запятой; CM [. . .] — значение соответствующих разрядов сумматора; CM[0] — знаковый разряд сумматора; CM = 0 — результат на выходе сумматора нулевой; PCM — перенос из сумматора; M3 = 0 — на M3 считан нулевой операнд.

Высокая достоверность результатов обработки информации обеспечивается узлом контроля. Особенности связи БАЛ с магистралями процессора и построения аппарата преобразования обусловили необходимость применения нескольких методов контроля. БАЛ оснащен схемами 6 (см. рис. 36) контроля входных данных, схемой 4 поразрядного сравнения результатов и схемами 5 формирования и коррекции контрольных разрядов результата. Во входных и выходных цепях БАЛ используется метод контроля информации по $\text{mod}2$, что отвечает общим принципам контроля передачи информации в ЦП (п. 1.4). Преобразование информации контролируется путем дублирования аппарата преобразования и последующего сравнения результатов. Основной и контрольный сумматоры (СМ и КСМ) включены параллельно и работают одновременно. При несовпадении значений хотя бы в одном разряде результатов сумматоров вырабатывается сигнал ошибки (ОШ АЛУ, см. табл. 35) от тетрадной группы, в которую входит этот разряд. Это позволяет кроме проверки исправности самих микросхем К155ИП3 контролировать также и работу схем формирования переносов и тракты их соединения с сумматором. Достоверность поступающей на обработку информации обеспечивается входными схемами потетрадного контроля. В случае несовпадения четности какой-либо тетрады с сопровождающим ее контрольным разрядом вырабатываются сигналы ОШ ВХА или ОШ ВХВ (табл. 35). Сигналы ошибок вырабатываются для каждой отдельной тетрады по каждому из входов А и В. Наличие тетрадного контроля входной информации и результата обработки позволило повысить степень локализации места неисправности, поскольку привязанный к конкретной тетраде сигнал ошибки значительно сужает область поиска.

Результат обработки поступает на магистраль М1 вместе с контрольными битами, сформированными соответствующими схемами БАЛ. Выработка их производится параллельно с работой схемы десятичной коррекции результата. При коррекции десятичного результата его четность может измениться. Поэтому для такого случая предусмотрена дополнительная коррекция контрольных битов. Контрольные биты тетрад результата поступают в магистральный коммутатор КМ1, где они преобразуются в контрольные биты байт числа на магистрали (п. 2.2). Для ускорения процессов формирования и коррекции контрольных разрядов рассматриваемые схемы выполнены на элементах с тремя состояниями (п. 1.5).

К средствам обеспечения достоверности информации относятся также схемы проверки правильности десятичных данных 7. Десятичные данные, поступающие на входы БАЛ, постоянно проверяются на правильность кодов десятичных цифр (кодами десятичных цифр являются 0000—1001). Если значение кода хотя бы одной из тетрад по любому из входов А или В превышает 1001, вырабатываются сигналы ОШ ДАН М2 или ОШ ДАН М3. Сигналы используются для прекращения выполнения команды и возбуждения процедуры обработки прерывания по особому случаю в данных.

Управление работой БАЛ осуществляется узлом управления. На его вход поступает управляющая информация из РМК (микроприказы) и синхроимпульсы из блока синхронизации. Управляющая информация определенным образом перерабатывается и в виде внутренних управляющих сигналов поступает на схемы блока. Так, в зависимости от заданной функции, в нужной временной последовательности включаются схемы коррекции формирования признаков, установки переносов и т. п.

2.5. БЛОК СДВИГА ИНФОРМАЦИИ.

Этот блок предназначен для сдвига информации при выполнении команд сдвига (пп. 6.1, 8.5), при выполнении операций умножения и деления (гл. 6.7), при выравнивании операндов в операциях десятичной арифметики (гл. 9) и операциях логической обработки (гл. 8). Блок сдвига также интенсивно используется при выполнении других микропрограмм.

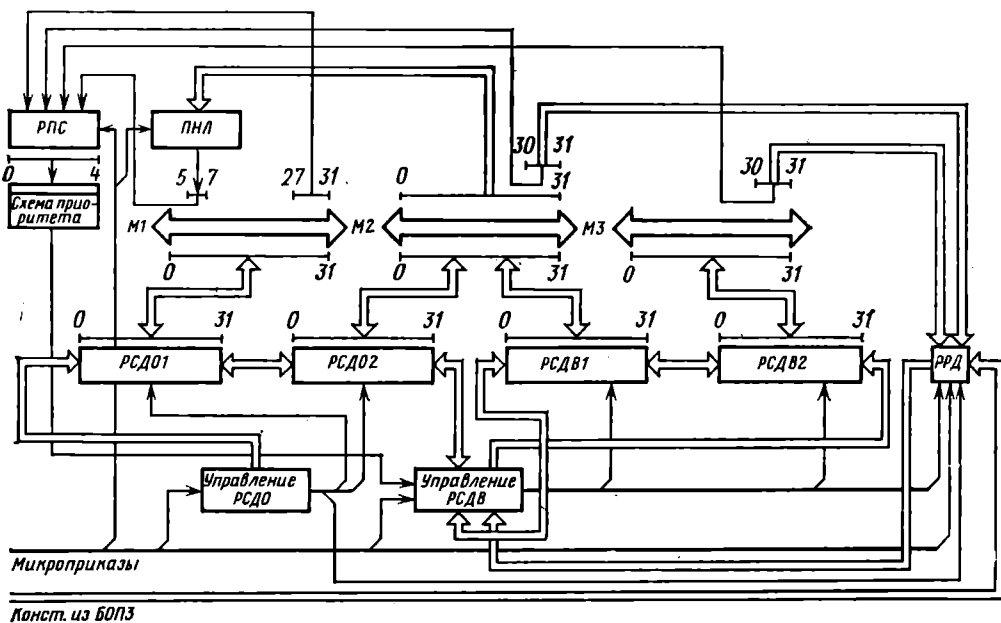


Рис. 37. Структурная схема блока сдвиговых регистров

Блок сдвига (БРСДВ) состоит из (рис. 37) двух 32-разрядных регистров сдвига РСДВ1 и РСДВ2, двух 32-разрядных регистров сдвига операционных РСДО1 и РСДО2, узлов управления этими регистрами, 4-разрядного регистра результата деления (РРД), 5-разрядного регистра параметра сдвига РПС со схемой приоритета и узла параметра нормализации (ПНЛ). Регистры РСДВ и РСДО могут работать автономно или совместно. Эти регистры иногда называют основным и дополнительным сдвигателями.

Регистры РСДВ предназначены для выполнения арифметических и логических сдвигов 64-разрядных слов [а. с. 591078 (СССР)].

Сдвиги могут быть фиксированными на 1, 2, 4, 8 и 16 разрядов, причем сдвиги на 1, 2, 4 и 8 выполняются за один машинный такт, а сдвиг на 16 — за два. Они могут осуществляться также по параметру сдвига на любое заданное число разрядов в пределах от 1 до 31 за один или несколько тактов в зависимости от значения параметра. Сдвиг более чем на 31 разряд ($N > 31$) выполняется в два этапа. Сначала микропрограммно пересылается либо младшее слово на место старшего (при сдвигах влево), либо старшее слово на место младшего (при сдвигах вправо), а затем производится сдвиг в соответствии со значением параметра влево или вправо на $n = N - 32$ разряда. Значение сдвига, равное или меньшее 31, перед сдвигом запоминается в РПС.

Регистры сдвига имеют непосредственные информационные связи с магистралями М2 и М3. Связи с РСДО и РРД осуществляются через схему управления РСДВ.

При логических сдвигах влево в освобождаемые справа разряды вдвигаются нули, которые формируются в РРД. При логических сдвигах вправо в освобождаемые слева разряды вдвигаются нули, которые формируются схемой управления РСДВ.

При арифметических сдвигах вправо в освобождаемые слева разряды распространяется значение знакового разряда сдвигаемой информации. При сдвигах в операции умножения с фиксированной запятой в освобождаемые старшие разряды распространяется значение переноса сумматора. Запоминание значений знака или переноса и их распространение осуществляет схема управления РСДО.

При арифметических сдвигах влево схема управления РСДВ запоминает значение знакового разряда перед сдвигом и восстанавливает его на М2 [0] после окончания сдвига. В освобождаемые справа разряды вдвигаются нули, которые формируются в РРД.

Операционный сдвигающий регистр РСДО предназначен для выполнения арифметических и логических сдвигов 64-разрядного слова на два и четыре разряда влево и вправо [А. с. 585755 (СССР)]. Он имеет непосредственные связи с магистралями М1 и М2 и со схемой управления РСДО. Связь с РСДВ осуществляется через схему управления РСДВ.

Арифметический сдвиг применительно к РСДО заключается в сохранении значения старшего (знакового) разряда при сдвигах вправо и влево; при сдвиге знак не распространяется. Это связано с тем, что арифметический сдвиг РСДО используется при обработке операндов в командах с плавающей запятой, где вслед за знаковым разрядом расположены разряды характеристики и в распространении значения знакового разряда нет необходимости. Сохранение значения знакового разряда и его восстановление после сдвига осуществляет схема управления РСДО.

Предусмотрена возможность синхронной работы РСДВ и РСДО под действием микроприказов управления работой РСДВ. При такой работе оба сдвигателя могут работать либо как автономные, т. е. без перехода информации из одного в другой, либо совместно,

т. е. с переходом информации из одного сдвигателя в другой. При автономной работе кроме синхронного сдвига работа сдвигателей не отличается от описанной выше. При совместной работе переход информации из РСДО в РСДВ при сдвигах вправо и из РСДВ в РСДО при сдвигах влево происходит через схему управления РСДВ. При этом число разрядов сдвигателя увеличивается до 128. Сдвиги 128-разрядных слов имеют место в операциях с плавающей запятой и десятичной арифметики. При выполнении этих операций используется также специальный режим блокировки записи в РСДО, осуществляемый микропрограммно.

Узел параметра нормализации ПНЛ предназначен для определения параметра нормализации в операциях с плавающей запятой.

Регистр результата деления кроме описанных выше функций формирует значения очередных разрядов частного. При этом они вдвигаются со стороны младших разрядов в РСДО2.

Для осуществления сдвига информации параметр сдвига разбивается на фиксированные величины, кратные степени 2, т. е. 2^4 , 2^3 , 2^2 , 2^1 , 2^0 , и преобразуется в пятиразрядный код сдвига. Каждый из разрядов этого кода задает сдвиг соответственно на 16, 8, 4, 2 и 1 разряд. Так, например, код 10010 означает сдвиг на 18 разрядов. Сдвиги на 1, 2, 4 и 8 разрядов выполняются за один машинный такт, сдвиг на 16 разрядов выполняется как два сдвига по 8 разрядов за два машинных такта. Код сдвига заносится в РПС [0/4]. По величине кода и направлению сдвига формируются сигналы управления сдвигом в РСДВ. Код сдвига может быть задан микропрограммно или сформирован в ходе выполнения микропрограммы по виду операндов или их расположению в ОП. Регистр параметра сдвига выполнен на RS-триггерах, выходы которых через схему приоритета подключены к схеме управления РСДВ. Первым обслуживается сдвиг на 1 разряд, затем — на 2 разряда и т. д. Каждый триггер в РПС, соответствующий фиксированному сдвигу на определенное число разрядов, сбрасывается после выполнения этого сдвига и разрешает прохождение разрешающего сигнала сдвига со следующего триггера. Нулевое состояние РПС означает, что все сдвиги выполнены. При этом формируется сигнал окончания выполнения сдвига на требуемое число разрядов.

Для задания типа и направления сдвига в микропрограммах используются следующие микроприказы: сдвиг правый логический СДВ ПР ЛОГ, сдвиг левый логический СДВ ЛЕВ ЛОГ, сдвиг правый арифметический СДВ ПР А, сдвиг левый арифметический СДВ ЛЕВ А. Величины сдвига задаются микроприказами: СДВ1, СДВ2, СДВ4. При этом по одному из последних трех микроприказов устанавливается в единицу соответствующий разряд РПС. Этот способ задания сдвига очень широко используется в самых разнообразных микропрограммах и, в частности, в операциях умножения (деления), в операциях упаковки (распаковки).

Сдвиг может быть задан микропрограммно еще и таким образом: код, соответствующий величине сдвига (например, код пози-

ции байта, хранящийся в одном из РОП), предварительно считывается на $M1 [27/31]$, а затем микроприказом $РПС := M1 [27/31]$ записывается в РПС. Направление и тип сдвига задаются соответствующими микроприказами.

Код параметра сдвига вместе с сигналами направления сдвига может быть сформирован в процессе выполнения микропрограммы в зависимости от ряда условий. Такой способ задания сдвига применяется, в частности, для выравнивания операндов при обработке данных переменной длины. Величина сдвига может быть задана в командном слове команд сдвига и через $M1 [27/31]$ записана в РПС.

Данные переменной длины расположены в оперативной памяти произвольно, начиная с любого байта. Перед обработкой их необходимо выравнивать, т. е. совместить крайние байты операндов перед подачей на аппаратуру обработки. При этом удобно выравнивать второй операнд по первому, так как тогда результат автоматически оказывается выровненным по границам первого операнда в ОП, на место которого он и записывается. Выравнивание может быть или по правой границе (при сложении, вычитании, сравнении десятичных данных) или по левой (например, при делении).

Поскольку границы операндов могут располагаться в любых байтах слова, то возможны 16 вариантов их взаимного расположения.

Для вычисления параметра выравнивания служит комбинационная схема, вычисляющая параметр и засылающая его в сдвигатель для отработки. Кроме величины сдвига схема вырабатывает и признак его направления, так как граница одного операнда может быть как левее, так и правее границы другого в пределах слова.

Исходными данными для вычисления параметра сдвига и его направления служат два последних разряда адреса обоих операндов, которые определяют позицию байта в слове. При этом при выравнивании по правым границам используются адреса последних байт (код крайнего правого байта — КПБ), а при выравнивании по левым границам — адреса начальных байт (код крайнего левого байта — КЛБ). Для установки требуемого параметра сдвига адреса первых (последних) слов операндов считываются на $M2 [8/31]$ и $M3 [8/31]$, при этом КЛБ (КПБ) операндов оказываются в позициях $M2 [30/31]$ и $M3 [30/31]$. Значение параметра сдвига и его направление вычисляются комбинационной схемой в соответствии с табл. 9.

Адрес операнда, относительно которого происходит выравнивание, размещается на $M2$, а адрес выравниваемого операнда — на $M3$. В РПС параметр сдвига заносится микроприказом $РПС := ПСДВ$, а направление и тип сдвига запоминаются на специальных триггерах. Согласно архитектуре ЕС ЭВМ минимально адресуемой единицей информации является байт, поэтому сдвиг при выравнивании может быть на 8, 16 или 24 разряда.

**Зависимость значения параметра и направления сдвига
от адресов крайних байт операндов**

| Код адреса байта на M2[30/31] | Сдвиги при различных значениях кода адреса байта на M3[30/31] | | | |
|-------------------------------------|---|---------------------|----------------------|----------------------|
| | 00 | 01 | 10 | 11 |
| 00 | | СДВ8 СДВ ЛЕВ ЛОГ | СДВ16 СДВ ЛЕВ ЛОГ | СДВ24 СДВ ЛЕВ ЛОГ |
| 01 | СДВ8 СДВ ПР ЛОГ | | СДВ8 СДВ ЛЕВ ЛОГ | СДВ16 СДВ ЛЕВ ЛОГ |
| 10 | СДВ16 СДВ ПР ЛОГ | СДВ8 СДВ ПР ЛОГ | | СДВ8 СДВ ЛЕВ ЛОГ |
| 11 | СДВ24 СДВ ПР ЛОГ | СДВ16 СДВ ПР ЛОГ | СДВ8 СДВ ПР ЛОГ | |

Указанные средства автоматического вычисления параметров позволяют выполнять обработку по единой микропрограмме для всех случаев расположения операндов, что значительно упрощает микропрограммы и снижает их объем. Использование аппарата быстрого сдвига позволяет значительно ускорить процесс обработки, так как основное время выполнения операции занимает извлечение операндов из памяти и их установка в удобную для обработки позицию.

При выполнении операций с плавающей запятой для увеличения точности вычислений операнды и результат нормализуются. В общем случае нормализация некоторого числа с плавающей запятой заключается в поочередных сдвигах мантиссы влево на одну тетраду (4 разряда) до тех пор, пока старшая тетрада мантиссы не станет отличной от нуля, и в уменьшении характеристики на величину, равную числу сдвигов.

Чтобы не делать поочередных сдвигов на 4 разряда с последующим анализом результата сдвига и для ускорения процесса нормализации в процессоре предусмотрено выполнение сдвигов с помощью сдвигателя на количество разрядов, соответствующее числу нулевых тетрад.

Параметр нормализации ПНЛ численно равен количеству расположенных подряд нулевых тетрад в мантиссе операнда, начиная со старшей. Для вычисления ПНЛ сдвигаемый операнд помещается на магистраль M2. Значения всех разрядов M2 поступают в схему вычисления ПНЛ, где происходит анализ на равенство нулю всех тетрад магистрали. По микроприказу $M1[5/7] := ПНЛ[8/31]$ формируется и считывается на $M1[5/7]$ параметр нормализации однословных операндов и старшего слова двухсловных операндов, а по микроприказу $M1[5/7] := ПНЛ[0/31]$ — параметр нормали-

зации младшего слова двухсловных операндов, если старшее слово содержит одни нулевые тетрады. Первый микроприказ не учитывает значения двух старших тетрад первого слова операнда, поскольку они заняты знаком и характеристикой.

Далее по микроприказу $RPC := M1[5/7]$ параметр сдвига помещается в $RPC[2/4]$. Направление и вид сдвига (СДВ ЛЕВ ЛОГ) задаются микропрограммно, а величина сдвига определяется па-

Т а б л и ц а 10

Характеристика сдвига при нормализации

| Число нулевых тетрад | Параметр нормализации $M1[5/8]$ | Значение сдвига |
|----------------------|---------------------------------|-----------------|
| 0 | 000 | СДВ 0 |
| 1 | 001 | СДВ 4 |
| 2 | 010 | СДВ 8 |
| 3 | 011 | СДВ 12 |
| 4 | 100 | СДВ 16 |
| 5 | 101 | СДВ 20 |
| 6 | 110 | СДВ 24 |
| 7 | 111 | СДВ 28 |

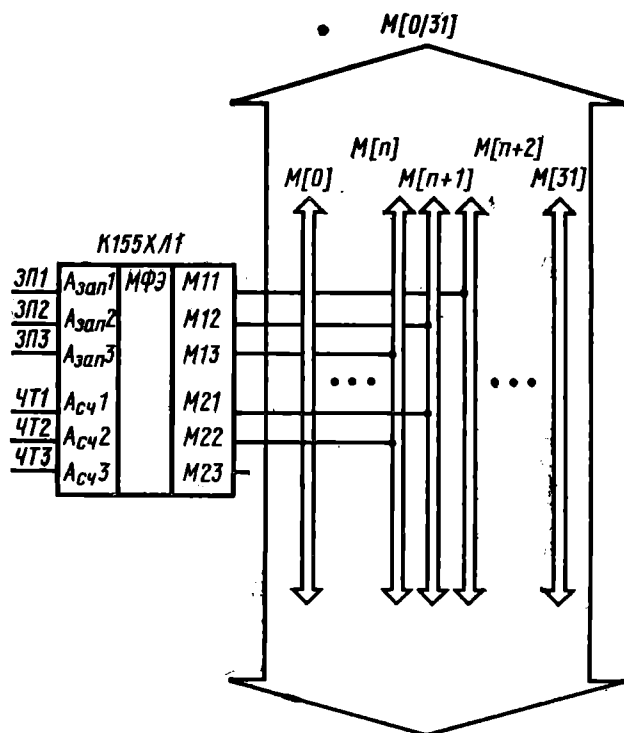


Рис. 38. Схема, поясняющая принцип работы сдвигателя

раметром нормализации согласно табл. 10.

Вычисленный параметр сдвига используется также для корректировки характеристики, которая выполняется с помощью БАЛ

вычитанием параметра нормализации из характеристики. Схема определения параметра нормализации не задает сдвиг на 32 разряда (случай нулевого операнда), так как этот случай выделяется заранее в микропрограммах. В узле параметра нормализации предусмотрено формирование значения контрольного разряда $M1[0/7]$. Это необходимо для того, чтобы проконтролировать передачу параметра нормализации на магистраль $M1[5/7]$. Контроль организован дополнением до четности.

Регистр параметра сдвига и схема приоритета реализованы в ТЭЗ «RPC», а узел ПНЛ — в ТЭЗ «ПНЛ».

Сдвигатели РСДВ и РСДО выполнены на ИС К155ХЛ1 (рис. 38), представляющих собой многофункциональные элементы (МФЭ).

Информационная магистраль $M[0/31]$ процессора имеет 32 информационных разряда. Все разряды построены одинаково,

каждый из них имеет элементы памяти, схемы перезаписи, маскирования и т. п. Выберем несколько разрядов — $M[n]$, $M[n+1]$, $M[n+2]$ и т. д. магистрали, к которым подключим выходы МЗ, М2, М1 одного разряда ИС К155ХЛ1. Если в первом полутакте по сигналу ЗП1 сделать запись в разряд МФЭ по первому адресу, т. е. через первый вход-выход, а во втором полутакте записанную информацию прочитать по сигналу ЧТ2 по второму адресу, т. е. через второй вход-выход, то окажется, что информация из разряда $M[n+2]$ переместилась в разряд $M[n+1]$. Произошел сдвиг

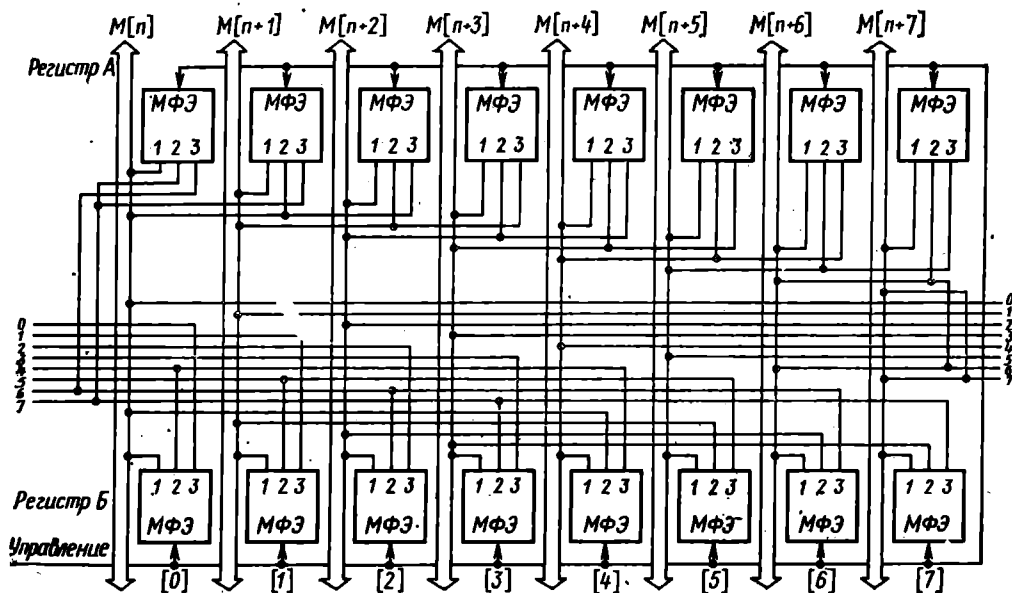


Рис. 39. Организация 8-разрядной группы РСДВ

информации влево на один разряд. Если сделать запись через первый вход-выход, а прочитать по сигналу ЧТ3 через третий вход-выход, то произойдет сдвиг информации на два разряда влево. При записи и чтении по первому вход-выходу сдвига информации не будет (сдвиг на нуль разрядов). Таким образом, сдвиг информации влево происходит при «прямой» записи в МФЭ и чтении «со сдвигом».

Если сделать запись сигналом ЗП2 через второй вход-выход, а чтение сигналом ЧТ1 через первый вход-выход, то произойдет сдвиг информации на один разряд вправо. При комбинации сигналов ЗП3—ЧТ1 произойдет сдвиг на два разряда вправо, а при комбинации ЗП1—ЧТ1 — нулевой сдвиг. Таким образом, сдвиг информации вправо происходит при записи в МФЭ «со сдвигом» и «прямом» чтении.

Рассмотрим функциональную схему организации восьмиразрядной группы регистра РСДВ (рис. 39). Регистр РСДВ состоит из двух регистров: регистра А и регистра Б, каждый из которых реализован на элементах МФЭ. На рис. 39 прямоугольником изображен один разряд МФЭ. Поскольку каждый элемент МФЭ содержит

два разряда, то в каждом из регистров *A* и *B* имеются по четыре элемента МФЭ. Цифрами 1, 2 и 3 обозначены входы-выходы *M1*, *M2* и *M3* элемента, а цепи записи и чтения объединены общей цепью «Управление». Регистры *A* и *B* отличаются соединением входов-выходов МФЭ с разрядами магистрали $M[n]—M[n+1]$. Регистр *A* реализует сдвиги на 1 и 2 разряда, а регистр *B* — на 4 и 8 разрядов в соответствии с табл. 11.

Стыковочные концы 0—7 служат для подключения слева и справа аналогичных восьмиразрядных групп для организации 64-разрядного РСДВ. Стыковочные концы крайней левой группы

Таблица 11

Сигналы записи и чтения при сдвигах РСДВ

| Число разрядов сдвига | Сдвиг | | | |
|-----------------------|-------|------|--------|------|
| | влево | | вправо | |
| 0 | ЗП1А | ЧТ1А | ЗП1А | ЧТ1А |
| 1 | ЗП1А | ЧТ2А | ЗП2А | ЧТ1А |
| 2 | ЗП1А | ЧТ3А | ЗП3А | ЧТ1А |
| 4 | ЗП1Б | ЧТ2Б | ЗП2Б | ЧТ1Б |
| 8 | ЗП1Б | ЧТ3Б | ЗП3Б | ЧТ1Б |

РСДВ1 подключены к схеме управления РСДВ (см. рис. 37), которая в зависимости от режима работы блока сдвига обеспечивает: переход информации из РСДВ в РСДО и обратно при синхронной совместной работе РСДВ и РСДО; сохранение и восстановление значения знакового разряда при арифметических сдвигах влево; распространение содержимого знакового разряда или значения переполнения сумматора или распространение нулей при сдвигах вправо. Стыковочные концы крайней правой группы

РСДВ2 также подключены к схеме управления РСДВ, которая в зависимости от режима работы обеспечивает поступление либо нулей, либо очередных бит результата операции деления из РРД в РСДВ.

Информация, предназначенная для сдвига в РСДВ, предварительно размещается на магистралях *M2* и *M3*. Сдвиг происходит в несколько этапов: информация, считанная на *M2* и *M3* в первом полутакте, во втором полутакте записывается с магистралей *M2* и *M3* в сдвиговый регистр *A* (*B*) «прямо» или «со сдвигом». В первом полутакте следующего такта информация читается на магистралях *M2* и *M3* из регистра *A* (*B*) также «со сдвигом» или «прямо». Таким образом происходит один фиксированный сдвиг и при этом сбрасывается соответствующий разряд в РПС. Если общий сдвиг, соответствующий величине параметра сдвига, должен быть выполнен за несколько фиксированных сдвигов, то рассмотренные циклы продолжают до тех пор, пока содержимое РПС не станет равным нулю. Такой сдвиг происходит за несколько тактов, и продвижение микропрограммы блокируется до тех пор, пока не поступит сигнал о нулевом значении РПС. Информация из сдвигателя по соответствующим микроприказам может быть считана на *M2* и *M3* как одновременно, так и раздельно. После сдвига информация

сохраняется в РСДВ и до задания следующего сдвига чтение на М2 и М3 может быть произведено либо сразу, либо спустя несколько тактов; возможно также многократное чтение из РСДВ.

Контроль операции сдвига в РСДВ осуществляется вычислением контрольных бит сдвинутой информации по значению исходных операндов, направлению и величине сдвига. На рис. 40 для примера изображены: сдвигаемый байт (СВБ) информации, тетрада левого байта (ТЛБ), тетрада правого байта (ТПБ). При сдвиге влево на один разряд сравниваются выдвигаемый влево из байта бит СВБ [0] и вдвигаемый в байт бит ПБ [0]. Если их значения различны, то формируется сигнал коррекции контрольного бита сдвигаемого байта. Откорректированный бит становится контрольным битом результата. При сдвиге влево на два разряда сначала

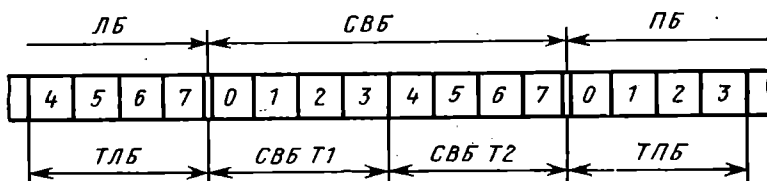


Рис. 40. Расположение байт информации перед сдвигом

сравниваются выдвигаемый влево из байта бит СВБ [0] и вдвигаемый справа в байт бит ПБ [0], затем выдвигаемый влево бит СВБ [1] и вдвигаемый справа бит ПБ [1], а результаты этих сравнений сравниваются еще раз. Результат последнего сравнения служит сигналом коррекции контрольного бита сдвигаемого байта. При сдвиге влево на четыре разряда сравниваются контрольные биты выдвигаемой влево тетрады СВБ Т1 и вдвигаемой справа тетрады ТПБ. Если же происходит сдвиг влево на 8 разрядов, то контрольный бит ПБ становится контрольным битом результата, поскольку сам правый байт становится результатом сдвига. Выработка сигналов коррекции при сдвигах вправо происходит аналогично.

Для крайних байт чисел, находящихся в РСДВ2, имеются дополнительные схемы коррекции значений контрольных разрядов. Для крайнего левого байта такая схема коррекции учитывает либо распространение слева направо нулей или единиц, либо последовательный ввод бит крайнего правого байта РСДО2 при сдвигах вправо. Это зависит от вида сдвига и режима работы блока сдвигателей. Эта же схема коррекции учитывает также восстанавливаемое значение знакового разряда при арифметических сдвигах влево. Для крайнего правого байта информации в РСДВ2 аналогичная схема коррекции учитывает либо распространение нулей справа налево, либо ввод очередных бит частного из РРД.

Сдвигатель РСДВ реализован на восьми ТЭЗ РСДВ (по одной восьмиразрядной группе со схемами контроля и коррекции в каж-

дом ТЭЗ, и двух ТЭЗ управления двигателем — УСДВ1 и УСДВ2.

В отличие от рассмотренного выше двигателя РСДВ двигатель РСДО — закрытого типа. Сдвиги на нем выполняются без выхода на внешние информационные магистрали процессора, с использованием только внутренних магистралей самого двигателя. Информация, подлежащая сдвигу, записывается в РСДО1 и РСДО2 с магистралью М1 и М2 (или М2 и М1) разряд в разряд. После этого производится ее сдвиг. Сдвинутая информация из РСДО разряд в разряд считывается на магистрали М1 и М2.

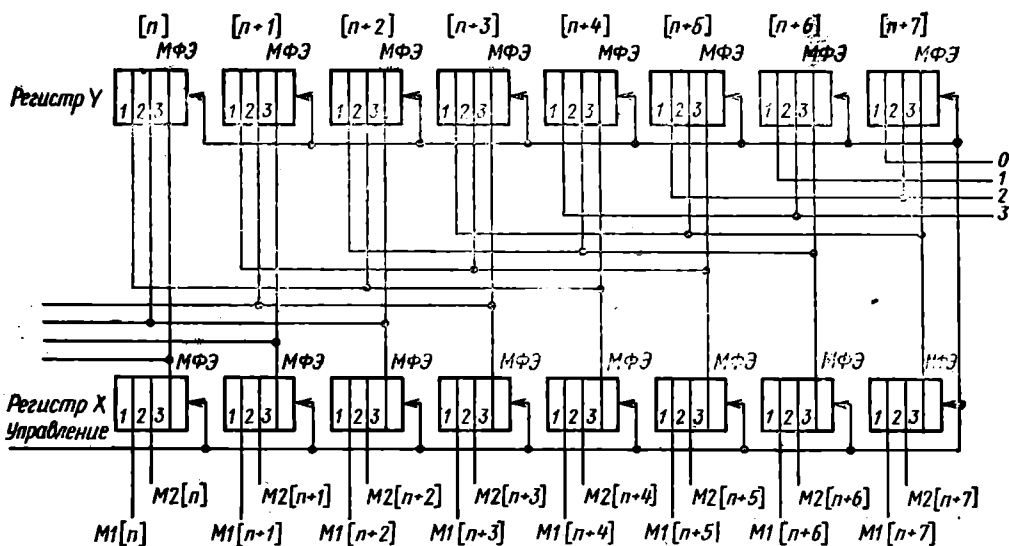


Рис. 41. Организация 8-разрядной группы РСДО

Рассмотрим функциональную схему организации восьмиразрядной группы РСДО (рис. 41). Регистр РСДО выполнен по двухрядной схеме. Он состоит из регистров X и Y, каждый из которых реализован на элементах МФЭ (K155ХЛ1). На рисунке прямоугольником изображен один разряд такого элемента. В каждом из регистров X и Y содержится по четыре элемента МФЭ, содержащих, в свою очередь, по два разряда. Цифрами 1, 2 и 3 обозначены входы-выходы М1, М2 и М3 элемента, а цепи записи и считывания объединены общей цепью «Управление». Соединение входов-выходов разрядов регистров X и Y производится через внутренние магистрали, не имеющие элементов памяти. При записи информации в РСДО с магистралью М1 и М2 она запоминается в регистре X. Сдвиг выполняется в два этапа. Пусть произведено чтение информации из регистра X через третий вход-выход сигналом чтения ЧТЗХ. Поскольку внутренние магистрали не имеют элементов памяти, то в такте чтения одновременно осуществляется запись в регистр Y. Предположим, эта запись происходит через вторые входы-выходы МФЭ регистра Y по сигналу ЗП2Y. Тогда содержащее, например, $[n + 4]$ разряда регистра X переместится в

[$n + 2$] разряд регистра Y . Далее одновременной подачей управляющих сигналов $ЧТЗУ$ и $ЗПЗХ$ информация без смещения передается в регистр X . При этом код, возвращенный в этот регистр, окажется сдвинутым относительно исходного на два разряда влево. Сдвинутая информация может быть считана на магистрали процессора $M1$ или $M2$. Остальные сдвиги реализуются при комбинации сигналов чтения и записи согласно табл. 12. При сдвигах вправо передача из регистра X в регистр Y осуществляется без смещения, а из регистра Y в регистр X — со сдвигом вправо на два или четыре разряда.

Т а б л и ц а 12

Сигналы чтения и записи при сдвигах РСДО

| Число разрядов сдвига | Сдвиг | | | |
|-----------------------|------------------|------------------|------------------|------------------|
| | влево | | вправо | |
| 0 | $ЧТЗХ$ $ЗПЗУ$ | $ЧТЗУ$ $ЗПЗХ$ | $ЧТЗХ$ $ЗПЗУ$ | $ЧТЗУ$ $ЗПЗХ$ |
| 2 | $ЧТЗХ$ $ЗП2У$ | $ЧТЗУ$ $ЗПЗХ$ | $ЧТЗХ$ $ЗПЗУ$ | $ЧТ2У$ $ЗПЗХ$ |
| 4 | $ЧТЗХ$ $ЗП1У$ | $ЧТЗУ$ $ЗПЗХ$ | $ЧТЗХ$ $ЗПЗУ$ | $ЧТ1У$ $ЗПЗХ$ |

Стыковочные выводы 0—3 слева и справа схемы на рис. 41 служат для подключения аналогичных 8-разрядных групп для организации 64-разрядного регистра РСДО. Стыковочные выводы левой крайней группы РСДО1 подключены к схеме управления РСДО, которая в зависимости от вида сдвига либо обеспечивает

вдвижение нулей слева (при логических сдвигах вправо), либо сохранение значения знакового разряда (при арифметических сдвигах влево и вправо). Сохранение знакового разряда отличает арифметические сдвиги вправо в РСДВ и РСДО. Стыковочные выводы крайней правой группы РСДВ2 подключены к схеме управления РСДО, которая обеспечивает либо продвижение нулей справа при левых сдвигах, либо переход информации из РСДО2 в РСДВ1 и обратно при синхронной совместной работе РСДО и РСДВ.

Ввиду более ограниченных функций РСДО он имеет более простые, чем в РСДВ, средства контроля. Для каждого байта сдвинутой информации формируется контрольный бит, который сопровождает сдвинутую информацию при считывании ее на информационную магистраль из РСДО (т. е. из регистра X). Для коррекции значений контрольных разрядов крайних правого и левого байт информации в РСДО имеются специальные схемы коррекции.

В схеме управления РСДО имеется аппарат блокировки записи информации в этот регистр по микроприказам блокировки записи в операциях с плавающей запятой БЛЗППЛ := 1 (п. 7.4) или блокировки записи в десятичных операциях БЛЗПД := 1 (п. 9.3). Признак блокировки запоминается в отдельном триггере, сброс которого происходит по микроприказу сброса блокировки записи СБР БЛЗП из микропрограммы операции или по микроприказу АНЗ8 в начале выборки команды (п. 4.4). Установка

этого триггера происходит из микропрограммы операций при соблюдении следующих условий:

$$\text{БЛЗПД} := 1 \wedge \text{ПАЛУ} = 0,$$

$$\text{БЛЗПЛ} := 1 \wedge \text{ПАЛУ} = 1,$$

где ПАЛУ — переносы АЛУ (пп. 2.4, 7.4, 9.3). Сигнал с выхода триггера поступает в схему управления РСДО, которая формирует сигналы блокировки записи информации в РСДО, а также сигнал блокировки модификации содержимого РРД.

Сдвигатель РСДО реализован на восьми ТЭЗ РСДО (по одной восьмиразрядной группе со схемами контроля в каждом ТЭЗ) и одном ТЭЗ управления операционным сдвигом УРСДО.

Регистр результата деления РРД (см. рис. 37) является четырехразрядным регистром, подключенным справа к регистру сдвига РСДВ. При выполнении логических сдвигов он выдает на магистраль МЗ в ее младшие разряды нули, а при выполнении операции деления — значение очередной цифры результата, который формируется в РСДВ.

При выполнении операции деления с фиксированной запятой (п. 6.3) за один цикл определяется 1 разряд результата, который по микроприказу РРД := МАН заносится в старший разряд РРД. В дальнейшем, при сдвиге на 1 разряд влево, он попадет в младший разряд формируемого в сдвигателе результата деления.

В операциях с плавающей запятой (п. 7.4), когда за 1 цикл вычисляется 2 цифры результата, используются 2 младших разряда РРД, причем его начальное значение устанавливается занесением константы из РМК.

В операциях десятичного деления (п. 9.3), когда цифра результата выражается четырьмя двоичными разрядами, используются все 4 разряда РРД.

Поскольку в двух последних способах деления очередная цифра результата определяется путем проб с последующей коррекцией, то РРД выполнен в виде двоичного счетчика, содержимое которого первоначально определяется занесенной в него константой, а впоследствии корректируется путем увеличения по микроприказу РРД := + 1. Регистр реализован в одном ТЭЗ «РРД».

2.6. БЛОК ПРЕОБРАЗОВАНИЯ ЧИСЕЛ ИЗ ОДНОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДРУГУЮ

Выполнение преобразования чисел из двоичной системы счисления в двоично-десятичную и из двоично-десятичной в двоичную выполняется с помощью команд конвертирования CVD и CVB (прил. 4). Алгоритм такого преобразования несложен и заключается в последовательных сдвигах на один разряд и последующей коррекции промежуточного результата.

В ЭВС ЕС-1033 команды конвертирования выполняются комбинированным аппаратно-микропрограммным способом [А. с. 591073

(СССР)]. Собственно преобразование чисел из одной системы счисления в другую выполняется аппаратно с помощью конверторов. Циклы преобразования, накопление результата, подача исходных операндов и перемещение промежуточных результатов организованы микропрограммно.

В качестве преобразователей используются ИС К155ПР6 и К155ПР7 (п. 1.5), называемые конверторами.

Для преобразования чисел из двоичной системы счисления в двоично-десятичную применяется ИС К155ПР7. Она представляет собой преобразователь двоичных кодов в двоично-десятичные (2/10), выполненный в виде ПЗУ емкостью 256 бит, в котором записаны двоично-десятичные эквиваленты подаваемых на его входы двоичных чисел в диапазоне от 0 до 63.

Работа простейшего преобразователя из двоичной системы счисления (рис. 42) происходит в соответствии с табл. 13.

Таблица 13

Преобразование чисел из двоичной системы счисления
в двоично-десятичную

| Двоичное число | | | | | | Двоично-десятичное число | | | | | | |
|----------------|----|---|---|---|---|--------------------------|---|---|---|---|---|---|
| 32 | 16 | 8 | 4 | 2 | 0 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| . | . | . | . | . | . | . | . | . | . | . | . | . |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| . | . | . | . | . | . | . | . | . | . | . | . | . |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . | . |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . | . |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

Из табл. 13 видно, что самый младший разряд чисел (разряд с весом «0») не преобразуется. Поэтому он на вход микросхемы преобразователя не подается. В диапазоне чисел от 0 до 9 коды двоичного и двоично-десятичного чисел совпадают. Младший разряд следующей тетрады двоичного числа имеет вес 16, а двоично-десятичного числа — 10 и т. д. Для преобразования чисел больше 63 требуется каскадирование преобразователей.

Рассмотрим преобразование восьмиразрядного двоичного числа с помощью каскадированного преобразователя (рис. 43), собранного на ИС К155ПР7 (у ИС2 выход 6 не задействован). Эта схема может преобразовывать числа в диапазоне 0—255. Пусть это будет

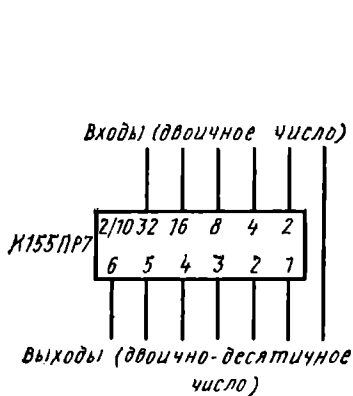


Рис. 42. Преобразователь из двоичной системы счисления в двоично-десятичную



Рис. 43. Преобразование числа 64 из двоичной системы счисления в двоично-десятичную

число 64 ($100\ 0000_2$). Для ИС1 исходный операнд — $01\ 0000_2$, т. е. 16. Промежуточный результат согласно табл. 13 в двоично-десятичной форме равен $001\ 0110$, т. е. 16_{10} . Для ИС2 исходный операнд — $1\ 1000_2$, а промежуточный результат в двоично-десятичной

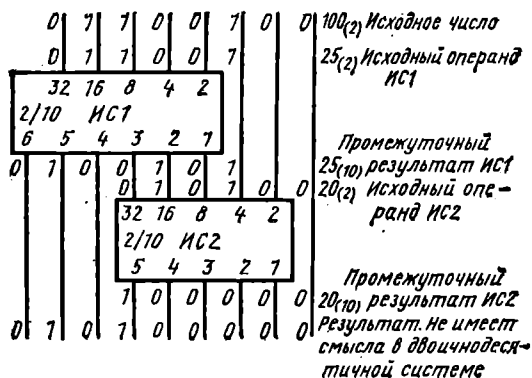


Рис. 44. Преобразование числа 100_{10} из двоичной системы в десятичную (без коррекции)

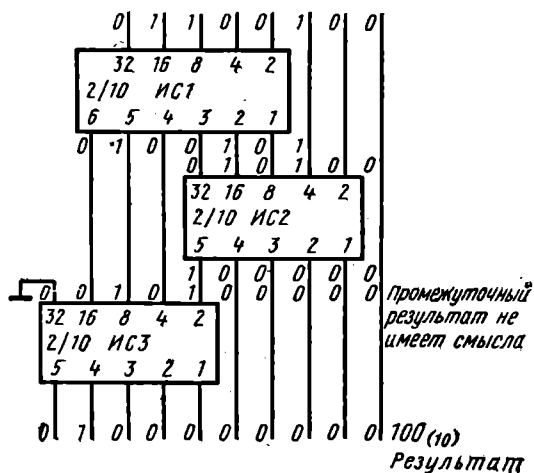


Рис. 45. Преобразование числа 100_{10} из двоичной системы счисления в десятичную (с коррекцией)

форме — $10\ 0100$, т. е. 24. Окончательный результат в двоично-десятичной форме — $110\ 0110$, т. е. 64. Пусть теперь исходным будет двоичное число $110\ 0100$, соответствующее десятичному числу 100 (рис. 44). Воспользовавшись табл. 13, определим результат. В тетраде десятков получился код цифры А (1010). Этот код является двоичным кодом цифры 10_{10} и должен быть подвергнут дополнительному преобразованию (откорректирован). Это делается

с помощью еще одной микросхемы ИСЗ (рис. 45). Схема преобразователя для 12-битного двоичного числа представлена на рис. 46, из которого легко определить правило построения преобразователя для двоичного числа любой разрядности. С увеличением разрядов числа быстро растет количество последовательно включенных каскадов и самих ИС. Так, для преобразования максимального 31-разрядного двоичного числа в десятичное (это число равно 2 147 483 647) требуется 80 ИС с 18 каскадами. При величине задержки одной ИС порядка

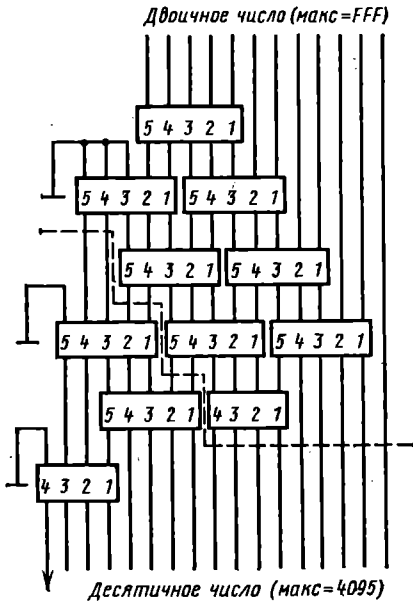


Рис. 46. Многокаскадный преобразователь 2/10

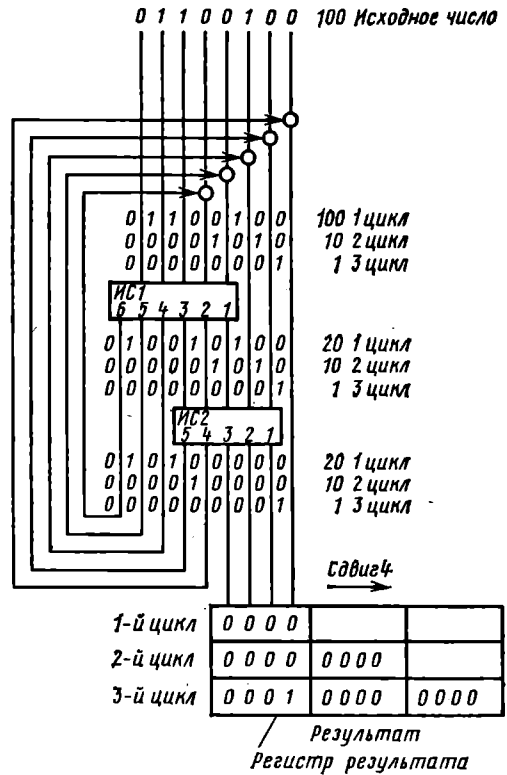


Рис. 47. Преобразование числа 100₁₀ из двоичной системы счисления в десятичную за несколько циклов

40 нс время преобразования составит 720 нс, т. е. может быть сделано за два машинных такта. Однако такой преобразователь имеет большой объем, сложен в изготовлении и наладке.

Другой путь преобразования состоит в том, что полученное по схеме, показанной на рис. 44, число считается промежуточным результатом первого цикла преобразования. В нем только младшая тетрада является окончательным результатом, для старшей необходима коррекция. Например, в конце первого цикла преобразования числа 100 (рис. 47) тетрада истинного результата помещается в регистр результата и сдвигается на 4 разряда вправо. Во втором цикле следующая старшая тетрада промежуточного результата, подлежащая коррекции, подается на вход преобразователя со сдвигом вправо на 4 разряда. На входе преобразователя теперь она будет занимать позицию младшей тетрады нового операнда. В старшую тетраду должны быть помещены нули. Этот

операнд вновь подвергается преобразованию. Следующий цикл построен аналогично. Для рассматриваемого примера количество циклов равно трем.

В ЭВМ ЕС-1033 для преобразования двоичных чисел применяется двухрядная матрица преобразователей, подобная отмеченной на рис. 46 пунктиром и расширенная по количеству входов

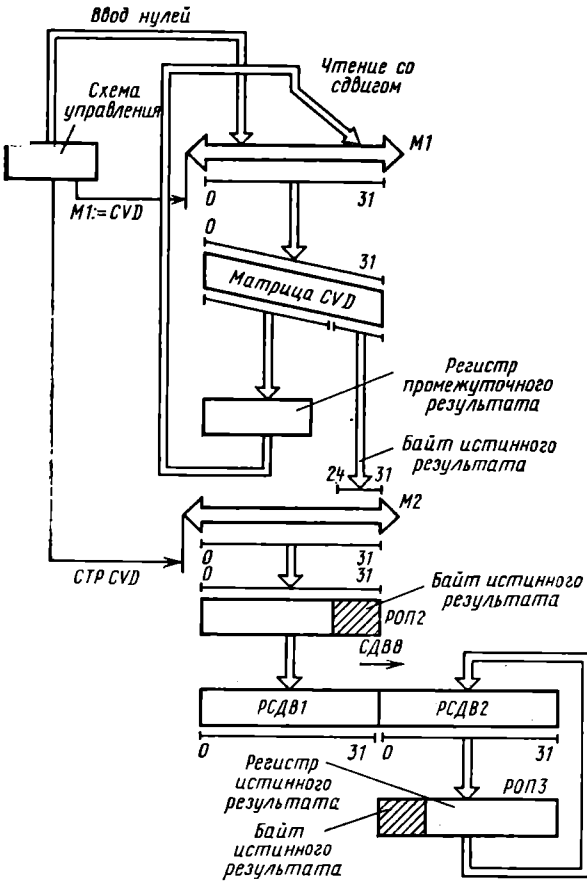


Рис. 48. Выполнение операции преобразования 2/10

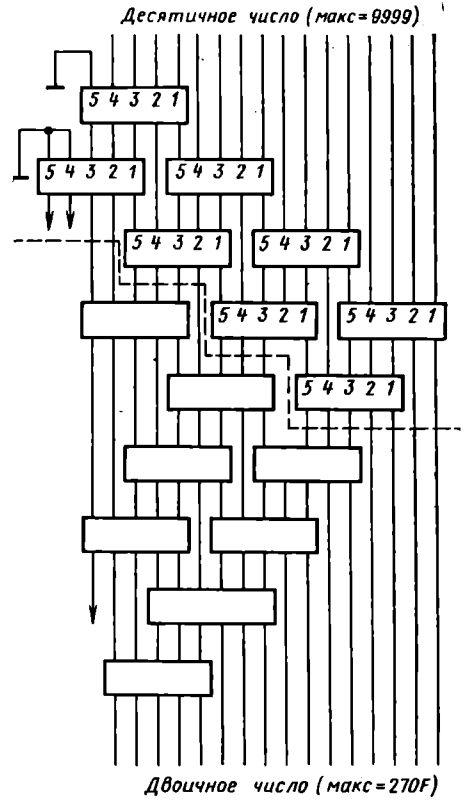


Рис. 49. Многокаскадный преобразователь 10/2

до 32. Такая матрица в каждом цикле преобразования позволяет получить один байт истинного результата; старшие 28 разрядов результата преобразования числа матрицей являются промежуточным результатом. Преобразуемое число перед началом операции располагается на магистрали M1 (рис. 48), к разрядам которой подключена матрица преобразователей. По микроприказу СТР CVD байт истинного результата выдается на M2 [24/31] и затем запоминается в РОП2. Промежуточный результат запоминается в регистре промежуточного результата. Байт истинного результата сдвигается сдвигателем РСДВ (п. 2.5) и пересылается в регистр истинного результата РОП3. Для этого байт результата записывается из РОП2 в РСДВ1, а в РСДВ2 из РОП3 заносится готовый результат (в первом цикле — нули). При сдвиге

ранее полученный истинный результат сдвигается вместе с последним полученным байтом истинного результата и из сдвигателя снова передается в РОПЗ, где цикл за циклом накапливается истинный результат. Вслед за сдвигом в основном сдвигателе очередного байта истинного результата промежуточный результат из регистра, где он хранится, по микроприказу $M1 := CVD$ читается на магистраль $M1$ со сдвигом на 8 разрядов вправо, и цикл повторяется. Максимально представимое 31-разрядное двоичное число 2 147 483 647 преобразуется за 5 циклов. Результат содержит 10 цифр.

Для преобразования числа из двоично-десятичной системы счисления в двоичную применяются ИС К155ПР6 (см. рис. 6). Работа однокаскадного преобразователя поясняется табл. 14.

Т а б л и ц а 14

Преобразование чисел из двоично-десятичной системы счисления в двоичную

| Двоично-десятичное число | | | | | | Двоичное число | | | | | |
|--------------------------|---|---|---|---|---|----------------|----|---|---|---|---|
| 5 | 4 | 3 | 2 | 1 | 0 | 32 | 16 | 8 | 4 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

Для преобразования чисел больше 39 требуется каскадирование преобразователей (рис. 49). С увеличением разрядов числа возрастает количество последовательно включенных каскадов.

В ЭВМ ЕС-1033 для преобразования десятичного числа применяется двухрядная матрица преобразователей, отмеченная пунктиром на рис. 49. Она позволяет за один цикл преобразования получить одну тетраду истинного результата, а старшие биты результата преобразования являются промежуточным результатом. Общая схема преобразования, с небольшим отличием, аналогична приведенной на рис. 48. Преобразуемое число может иметь до

64 разрядов, поэтому используется двухрядная матрица на 64 разряда. Двоичный результат не превышает 31 разряда (старшие разряды результата отбрасываются). Число, подлежащее преобразованию, перед началом операции размещается на магистралях М1 и М3, разряды которых подключены к входам двухрядной матрицы преобразователей. По микроприказу СТР CVB тетрада истинного результата читается на М2[28/31], а промежуточный результат запоминается в регистре промежуточного результата. Тетрада истинного результата сдвигается в РСДВ на 4 разряда вправо, и результат накапливается тетрада за тетрадой в регистре истинного результата. После сдвига тетрады истинного результата и ее передачи в регистр истинного результата по микроприказу М1М3 := CVB промежуточный результат считывается на М1 и М3 из регистра промежуточного результата со сдвигом на 4 разряда вправо, и цикл повторяется. Для получения 31-разрядного двоичного результата необходимо 8 циклов преобразования.

Одна из особенностей операций преобразования из одной системы счисления в другую заключается в том, что магистрали процессора, на которых хранится преобразуемое число, сохраняют свое значение в течение нескольких тактов подряд для обеспечения нормальной работы схем конвертирования. Для этого сброс магистралей микропрограммно блокируется (п. 1.6).

Узел преобразования десятичных чисел реализован в четырех ТЭЗ конвертирования КНВ1, узел преобразования двоичных чисел — в трех ТЭЗ КНВ2.

ГЛАВА 3

УПРАВЛЕНИЕ В ЦЕНТРАЛЬНОМ ПРОЦЕССОРЕ

3.1. БЛОК УПРАВЛЯЮЩИХ РЕГИСТРОВ

В ходе решения задачи процессор выполняет последовательность операций, в основном определяемую программой. При этом внутри машины и в связанной с ней внешней среде (например, в объекте управления или в другой машине) могут возникать события, требующие прерывания исполняемой программы. Информация, управляющая последовательностью выборки и выполнения программы, а также отражающая текущее состояние вычислительного процесса, включая возможность обработки того или иного прерывания, хранится в блоке управляющих регистров (БРУ, см. рис. 22). Дисциплина обслуживания прерываний определяется блоком обработки прерываний (БОПР). Связь с внешней средой обеспечивает блок прямого управления (БПРУ).

В состав БРУ (рис. 50) входят регистр слова состояния программы (РССП) (прил. 7), регистр адреса процессора (РАП), используемый при обращении к ОП, и буферный регистр адреса команды (РАКБ). Основной особенностью блока является то, что в своей работе он не использует аппаратуру обработки данных.

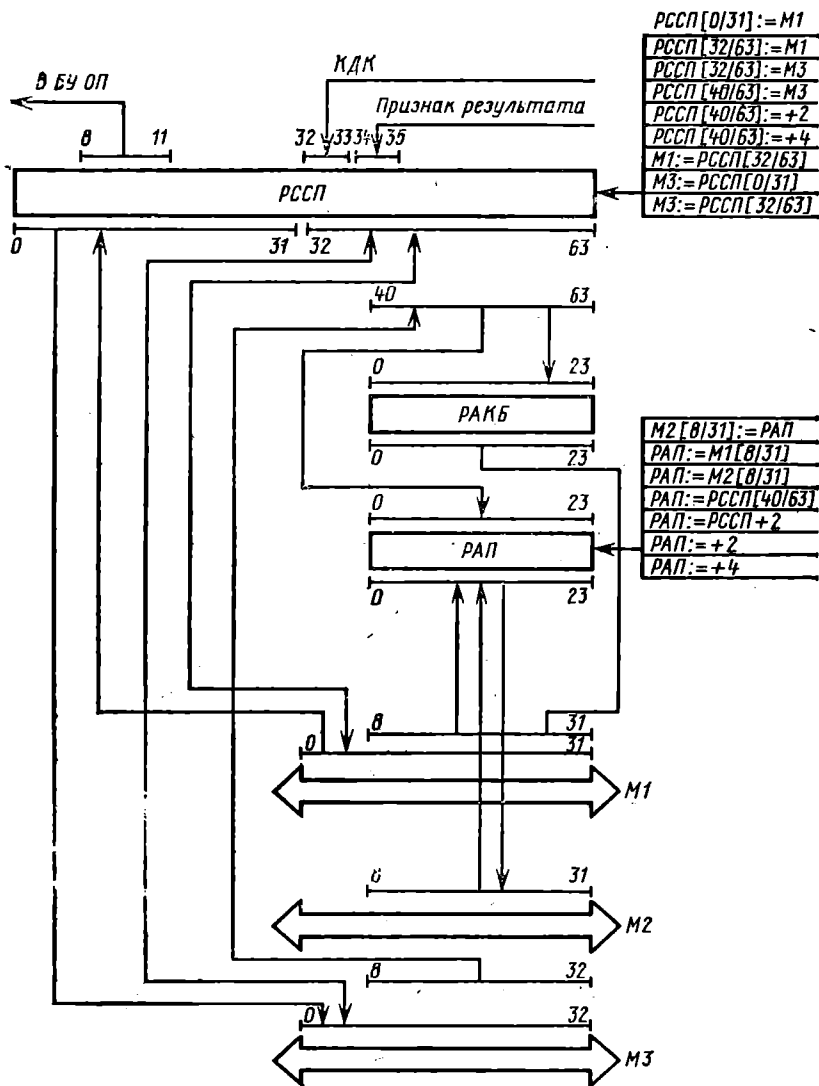


Рис. 50. Структурная схема блока управляющих регистров

Это позволило распараллелить процессы обработки данных и управления и тем самым повысить общее быстродействие процессора.

РССП[32/33] содержит код длины команды (КДК), вычитая который из продвинутого адреса, можно определить адрес исполняемой команды. В машине для оперативного повторения команды при сбоях адрес исполняемой команды хранится в РАКБ. В этот регистр он поступает по микроприказу АН38, с которого начи-

нается выборка любой команды (п. 4.4). При повторении команды содержимое РАКБ читается на М1 по диагностическому микроприказу М1 := РАКБ, затем помещается в адресную часть РССП.

РССП может обмениваться 32-разрядными словами с магистралями М1 и М3 для выполнения команд ЗАГРУЗКА ССП (п. 5.2), УСТАНОВИТЬ МАСКУ ПРОГРАММЫ, УСТАНОВИТЬ МАСКУ СИСТЕМЫ, при смене ССП в команде ОБРАЩЕНИЕ К СУПЕРВИЗОРУ и т. д. Обмен может осуществляться также с целью изменения значения различных полей и отдельных разрядов ССП при выполнении микропрограмм тех или иных команд, для анализа значения этих полей и разрядов.

Кроме того, в адресную зону РССП [40/63] можно записать информацию с М3 [8/31]. Это используется в командах перехода (п. 5.1), при повторении команды и др. В частности, при повторении команды после сбоя содержимое РАКБ, считанное на М1, пересылается через БАЛ на М3, откуда записывается в РССП [40/63], обеспечивая повторную выборку команды, в которой произошел сбой. Адресная зона выполнена в виде счетчика. По микроприказам РССП [40/63] := + 2 и РССП [40/63] := + 4 осуществляется указанное увеличение содержимого счетчика, что соответствует продвижению адреса на полуслово или на слово. Одновременно по любому из этих микроприказов в РССП [32/33] записывается КДК, который вырабатывается схемно в зависимости от формата выбираемой команды.

Занесение в РССП признака результата происходит по одному из сигналов записи признака (см. табл. 8).

Содержимое зоны РССП [8/11], в которой хранится ключ защиты, передается в виде потенциальных сигналов в БУ ОП (п. 3.4), где сравнивается с ключом защиты, хранящимся в памяти ключей защиты (ПКЗ).

Регистр адреса предназначен для передачи в ОП кода адреса команды или данных при обращениях со стороны ЦП. Так же, как и адресная часть РССП, РАП выполнен в виде счетчика, в который поступают адреса с магистралей М1 [8/31] и М2 [8/31]; содержимое РАП может быть считано на М2 [8/31]. Кроме того, адрес в этот регистр может быть записан из адресной зоны РССП [40/63] для выборки команды из ОП. На этапе выборки команды адрес в РАП, так же как и в РССП, может быть продвинут по микроприказам РАП := + 2 и РАП := + 4. По микроприказам РАП := РССП [40/63] и РАП := РССП [40/63] + 2 осуществляется перепись в РАП содержимого адресной части РССП либо без изменений, либо увеличенного на 2. Изменение содержимого РАП происходит также по сигналу запроса выборки (ЗПРВ). По этому сигналу осуществляется передача РАП := := РССП [40/63], если за командой надо обращаться в ОП. Если же очередная команда или ее часть была выбрана из ОП раньше и теперь находится в регистре хранения команды (РХК), то по микроприказу ЗПРВ осуществляется действие РАП :=

: = РССП [40/63] + 2 для довыборки оставшейся части команды из ОП. Реализация РАП и РССП [40/63] в виде регистров-счетчиков позволяет осуществлять продвижение адреса параллельно с работой основного оборудования ЦП, что также повышает общее быстродействие процессора.

При обращении к ОП с пульта управления в РАП передается адрес, набранный на клавиатуре ПУ. Выходы РАП подключены к коммутатору адреса ОП в блоке управления памятью БУ ОП.

Коммутатор адреса передает в ОП либо код адреса процессора из РАП, либо код адреса канала из регистра адреса канала.

Конструктивно БРУ реализован в следующих ТЭЗ: регистр слова состояния программы («РССП [32/39]») — 1 шт., регистр адреса процессора РАП — 5, контроль счетчиков КСТ — 3, счетчик РАП «СТРАП» — 2 шт., «регистр хранения команды» РХК — 3 шт.

В ТЭЗ РССП [32/39] реализованы разряды РССП [0/7] вместе с контрольным разрядом байта и РССП [32/39]. Разряды РССП [8/15], РССП [16/23], РССП [24/31] вместе с контрольными разрядами реализованы в ТЭЗ РХК. В каждом из пяти ТЭЗ РАП реализовано по 4 разряда адресных регистров: (РССП [40/63], РАКБ [0/3], РАП [0/3]) — (РССП [56/59], РАКБ [16/19], РАП [16/19]). Последняя тетрада разрядов регистров РССП и РАП выполнена в виде счетчика, построенного на ИС К155ИПЗ и поэтому РССП [60/63], РАКБ [20/23] и РАП [20/23] реализованы в ТЭЗ «СТРАП». Контрольные разряды байт реализованы в ТЭЗ КСТ. Организация контроля РАП и РССП аналогична организации контроля СТЦ (п. 2.3).

3.2. СИСТЕМА ПРЕРЫВАНИЙ И ПРИОРИТЕТОВ

Существует 5 классов прерываний: прерывания от ввода-вывода, внешние, при обращении к супервизору, программные и от схем аппаратного контроля. Самым высоким приоритетом обладает запрос на прерывание от схем контроля, сигналом прерывания служит сигнал ошибки (п. 10.1). При этом выполнение текущей команды прекращается. Управление передается блоку диагностики, осуществляющему запись состояния процессора в момент ошибки. При этом состояние магистралей и регистров ЦП записывается в диагностическую область ОП. Затем выполняется ряд диагностических процедур, после чего происходит собственно прерывание от схем контроля, состоящее в замене ССП. Сигналы запросов от остальных источников прерываний поступают в блок обработки прерываний. Он состоит (рис. 51) из регистра фиксации прерываний (РФП), дополнительного регистра фиксации прерываний (РФПД), схемы анализа наличия прерываний, приоритетной схемы и схемы кодирования.

Сигналы прерываний асинхронно и независимо поступают в триггеры трехразрядного регистра РФП, информация из которого для анализа переписывается в четырехразрядный регистр

РФПД. В четвертый разряд РФПД из общего канала поступает синхронно запрос на прерывание от ввода-вывода, минуя РФП. Разряды РФП и РФПД по классам прерываний распределены следующим образом: РФП[1] — программные прерывания; РФП[2] — прерывания при обращении к супервизору; РФП[3] — прерывания от внешнего источника; РФПД[4] — прерывания от ввода-вывода.

РФП[1] устанавливается в единичное состояние либо микроприказами $\text{РФП}[1] := 1$ и $\text{РФП}[1] := 1(\text{ДП})$, когда причина

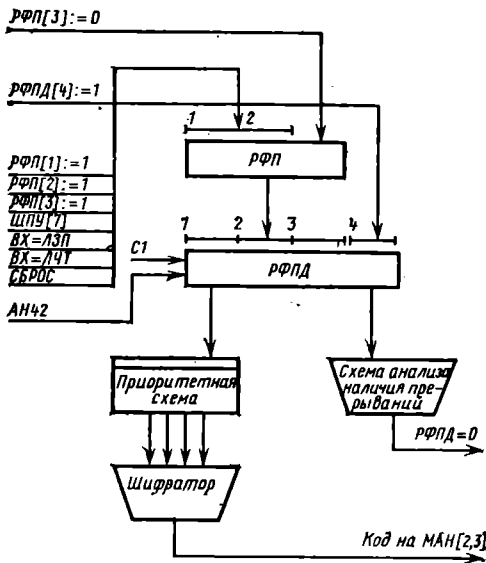


Рис. 51. Структурная схема блока обработки прерываний

прерывания определяется микропрограммно, либо аппаратно вырабатываемыми сигналами при нарушении защиты памяти, ошибке адресации и неправильной спецификации из БУ ОП; при некорректности данных — из блока анализов; при переполнении с фиксированной запятой — из БАЛ. Одно из программных прерываний — десятичное переполнение (ДП) может быть замаскировано нулевым значением разряда $\text{РССП}[37]$ (прил. 7, 8). Схемные сигналы поступают в блок режимов работы, где с помощью шифратора пульта управления (ШПУ) формируются начальные адреса микропрограмм прерывания (п. 4.3): О. А. 8 — защиты памяти, О. А. 9 — ошибки адресации, О. А. А — спецификация, О. А. В — ошибки данных, О. А. С — переполнения с фиксированной запятой. Все эти адреса имеют единичное значение разряда 7, поэтому при названных выше прерываниях сигнал ШПУ[7] из БРР устанавливает РФП [1] в единицу.

РФП[2] устанавливается в единичное состояние микроприказом $\text{РФП}[2] := 1$ при выполнении команды ОБРАЩЕНИЕ К СУПЕРВИЗОРУ.

РФП[3] устанавливается в единичное состояние микроприказом $\text{РФП}[3] := 1$ по окончании микропрограммы обработки таймера (п. 4.3), а также от кнопки прерывания на пульте управления и сигналами от внешних источников, проходящих по входным линиям. Внешние прерывания могут быть замаскированы разрядом $\text{РССП}[7]$.

РФПД[4] устанавливается сигналом $\text{РФПД}[4] := 1$ из каналов ввода-вывода.

Из приоритетной схемы информация поступает в кодирующую схему. Для выхода на микропрограмму обслуживания соответ-

прерывания определяется микропрограммно, либо аппаратно вырабатываемыми сигналами при нарушении защиты памяти, ошибке адресации и неправильной спецификации из БУ ОП; при некорректности данных — из блока анализов; при переполнении с фиксированной запятой — из БАЛ. Одно из программных прерываний — десятичное переполнение (ДП) может быть замаскировано нулевым значением разряда $\text{РССП}[37]$ (прил. 7, 8). Схемные сигналы поступают в блок режимов работы, где с помощью шифратора пульта управления (ШПУ) формируются начальные адреса микро-

ствующего прерывания (п. 4.3) по сигналу опроса АН42 (3) номер класса прерывания выдается в двоичном коде на магистраль анализов МАН согласно табл. 15.

Сбор триггеров разрядов РФП [1] и РФП [2] происходит по сигналу опроса АН42, РФП [3] — микропрограммно, после окончания обработки таймера РФП [4] — сигналом из каналов ввода-вывода.

Таблица 15

Номер класса прерывания и формирование начального адреса микропрограммы прерывания

| РФПД | МАН[2/3] | Адрес микро-программы прерывания |
|---------|----------|----------------------------------|
| РФПД[1] | 11 | 5.F.0111 |
| РФПД[2] | 10 | 5.F.0110 |
| РФПД[3] | 01 | 5.F.0101 |
| РФПД[4] | 00 | 5.F.0100 |

Обработка прерывания центральным процессором (п. 4.3) заключается в записи в ОП по фиксированному адресу текущего слова состояния программы в качестве старого ССП, в разрядах [16/31] которого хранится код причины прерывания, и выборке из ОП по другому фиксированному адресу нового ССП. Последнее в качестве текущего записывается в РССП (прил. 10). Таким образом, дальнейшая работа системы определяется значением нового ССП.

Программное обслуживание прерываний происходит в порядке, обратном смене ССП. Это означает, что первым обслуживается прерывание от ввода-вывода, затем внешнее, по вызову супервизора и, наконец, последним обслуживается программное прерывание.

Блок обработки прерываний реализован в одном ТЭЗ — регистр прерываний «РПР».

3.3. СРЕДСТВА ПРЯМОГО УПРАВЛЕНИЯ

Средства прямого управления предназначены для обмена информацией между процессорами двух ЭВМ или между процессором и специальными устройствами ввода-вывода (СУВВ) через интерфейс прямого управления. Связь между процессорами двух ЭВМ или между процессором и СУВВ осуществляется с помощью внешних прерываний по командам ПРЯМАЯ ЗАПИСЬ и ПРЯМОЕ ЧТЕНИЕ.

Интерфейс прямого управления ЭВМ ЕС-1033 содержит по 8 входных ВХ-Ш0 — ВХ-Ш7 и выходных ВЫХ-Ш0 — ВЫХ-Ш7 линий информации; по 8 входных и выходных линий синхросигналов ВС-0 — ВС-7 и ЛС-0 — ЛС-7; 4 линии управления ВЫХ-ЛЧТ, ВЫХ-ЛЗП, ВХ-ЛЧТ, ВХ-ЛЗП и по 2 линии (для приема и для выдачи) сигналов ВНЕШНИЙ ПРОГРАММНЫЙ СБРОС, ВНЕШНИЙ ПУСК, ВЫЗОВ СИСТЕМЫ, ВНЕШНЯЯ ОШИБКА, ВНЕШНИЙ АППАРАТНЫЙ СБРОС.

Структурная схема блока прямого управления БПРУ представлена на рис. 52.

Восьмиразрядный регистр приема информации внешних линий (РПИВЛ) предназначен для приема байта информации от ЭВМ-2 или СУВВ.

Регистр приема синхроимпульсов внешних линий (РПСИВЛ) предназначен для приема внешних синхросигналов от ЭВМ-2 или СУВВ в РПСИВЛ [2/7], а также для приема запроса на прерывание от таймера в РПСИВЛ [0] и с пульта управления

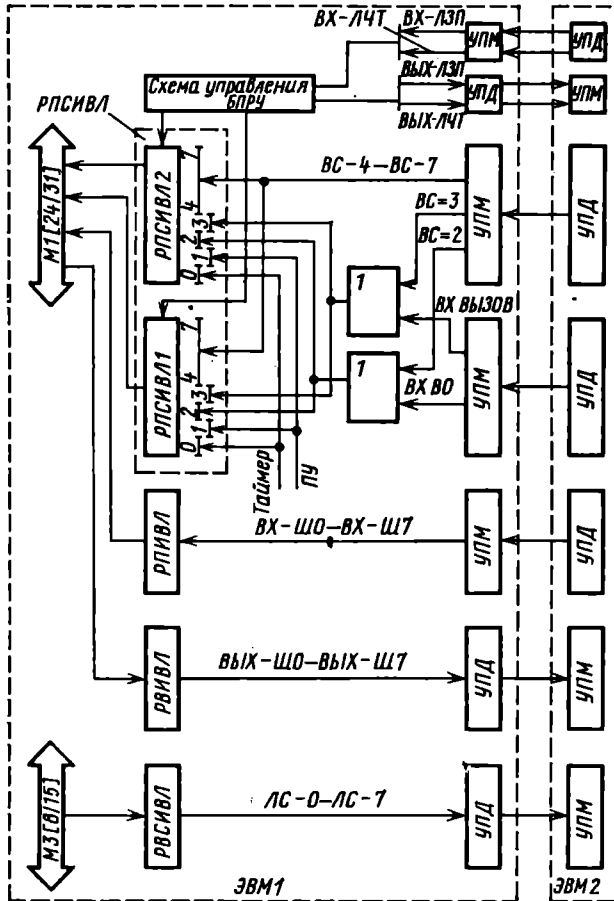


Рис. 52. Структурная схема блока прямого управления

в РПСИВЛ [1]. Так как два из внешних синхросигналов (нулевой и первый) в ЭВМ ЕС-1033 не используются, они подменяются сигналами от таймера и от пульта управления. РПСИВЛ состоит конструктивно из двух восьмиразрядных регистров РПСИВЛ1 и РПСИВЛ2, что позволяет избежать потерь синхросигналов внешних линий при отключении от линий одного регистра и подключении другого. В то время как один из регистров осуществляет прием сигналов извне, другой подключен к магистрали М1 [24/31] для обработки его содержимого.

Восьмиразрядный регистр выдачи информации внешних линий (РВИВЛ) предназначен для выдачи байта информации, считанного из ОП, в ЭВМ-2 или СУВВ.

Восьмиразрядный регистр выдачи синхроимпульсов внешних линий (РВСИВЛ) предназначен для выдачи синхросигналов в ЭВМ-2 или СУВВ при выполнении команды ПРЯМАЯ ЗАПИСЬ.

Схема управления БПРУ обеспечивает поочередное подключение РПСИВЛ1 и РПСИВЛ2 к внешним линиям и к магистрали М1. Она также осуществляет сброс РПСИВЛ1 или РПСИВЛ2 после передачи его содержимого на магистраль. Входные сигналы ВС-0 — ВС-7 через усилители-приемники (УПМ) поступают на РПСИВЛ, причем на тот из них, который в данный момент подключен к внешним линиям. РПСИВЛ в своем составе имеет схему формирования значения контрольного разряда. Это необходимо, так как синхросигналы и информация внешних линий им не сопровождается, а чтение информации на магистраль предполагает обязательное наличие побайтного контроля. После того как содержимое соответствующего РПСИВЛ выдано на магистраль, регистр сбрасывается в нулевое состояние.

Байт синхросигналов, выдаваемых вовне, при выполнении команды ПРЯМАЯ ЗАПИСЬ сопровождается сигналом ВЫХ-ЛЗП, а при выполнении команды ПРЯМОЕ ЧТЕНИЕ — сигналом ВЫХ-ЛЧТ. Байт синхросигналов, поступающих извне, при выполнении команды ПРЯМОЕ ЧТЕНИЕ сопровождается сигналом ВХ-ЛЗП, а при выполнении команды ПРЯМАЯ ЗАПИСЬ — сигналом ВХ-ЛЧТ. Выходная линия записи проходит от одного процессора к другому или к СУВВ. При связи с процессором она присоединяется к входной линии записи другого процессора. Входная линия чтения используется для подключения выходной линии чтения другого процессора.

Байт информации, принимаемой извне по входным шинам ВХ-Ш0 — ВХ-Ш7, поступает в регистр приема информации внешних линий (РПИВЛ) по сигналу ВХ-ЛЗП. Чтение из РПИВЛ на магистраль М1 [24/31] происходит по микроприказу М1 := := РПИВЛ. Регистр РПИВЛ в своем составе также содержит схему формирования значения контрольного разряда.

При выполнении команды ПРЯМАЯ ЗАПИСЬ ее биты [8/15] поступают в РВСИВЛ через магистраль М3 по микроприказу РВСИВЛ := М3 [8/15] и используются в качестве синхросигналов, которые через усилители-передатчики (УПД) поступают на выходные линии синхросигналов ЛС-0 — ЛС-7.

Байт информации, считанной из ОП, через магистраль М1 [24/31] передается в РВИВЛ по микроприказу РВИВЛ := := М1 [24/31], а затем через УПД поступает на выходные линии информации ВЫХ-Ш0—ВЫХ-Ш7.

При выполнении команды ПРЯМОЕ ЧТЕНИЕ появлению данных на входных шинах предшествует появление синхросигналов, сопровождаемых сигналом ВХ-ЛЧТ. Этот сигнал, по-

ступая в схему управления БПРУ, осуществляет формирование сигналов либо РПСИВЛ1 := ВЛ, либо РПСИВЛ2 := ВЛ. По микроприказу М1 := РПСИВЛ (в микропрограмме обработки прерываний) содержимое соответствующего РПСИВЛ поступает на магистраль М1 для идентификации источника прерывания.

По сигналу ВХ-ЛЗП происходит запись данных с линий ВХ-Ш0—ВХ-Ш7 в РПИВЛ. По микроприказу М1 [24/31] := := РПИВЛ информация из РПИВЛ поступает на магистраль М1 для записи ее в ОП.

При выполнении команды ПРЯМАЯ ЗАПИСЬ появлению данных на выходных шинах предшествует появление бит [8/15]

Т а б л и ц а 16

Значение разрядов поля 12 команды ПРЯМАЯ ЗАПИСЬ

| Разряды командного слова | | | | | Сигнал |
|--------------------------|---|----|----|-------|---------|
| 8 | 9 | 10 | 11 | 12/15 | |
| 0 | 0 | 0 | 0 | 0000 | — |
| 1 | 0 | 0 | 0 | 0000 | ВП |
| 0 | 1 | 0 | 0 | 0000 | СБР ПРГ |
| 0 | 0 | 1 | 0 | 0000 | — |
| 0 | 0 | 0 | 1 | 0000 | ВЫЗОВ |

команды, поступающих в качестве синхросигналов в РВСИВЛ по микроприказу РВСИВЛ := := М3 [8/15]. Они сопровождаются сигналом Вых-ЛЗП.

Запись данных в РВИВЛ с магистрали М1 осуществляется по микроприказу РВИВЛ := М1 [24/31].

Если два процессора находятся в режиме «Мультисистема», между ними осуществляется связь посредством специальных сигналов, передаваемых по отдельным линиям.

Эти сигналы называются мультисистемными. Режим «Мультисистема» задается нажатием клавиши «МУЛЬТИСИСТ.» на пульте управления (п. 3.6). Если эта клавиша отжата, то прием и выдача мультисистемных сигналов заблокированы.

Команда ПРЯМАЯ ЗАПИСЬ используется для выработки мультисистемных сигналов внешнего пуска (ВП), внешнего программного сброса (СБР ПРГ) и вызова системы (ВЫЗОВ) (табл. 16).

Сигнал ВП осуществляет пуск без производства сброса процессора, получившего этот сигнал. Если процессор, принимающий данный сигнал, находится в состоянии ОЖИДАНИЕ или СТОП, пуск выполняется сразу же. В противном случае пуск откладывается до окончания выполнения текущей операции. При выполнении внешнего пуска производится загрузка нового ССП и начинает выполняться команда, адрес которой содержится в адресной зоне загружаемого ССП. Новое ССП выбирается из ОП по адресу, равному нулю. Операции ввода-вывода не инициируются, и старое ССП не запоминается.

Сигнал СБР ПРГ предназначен для сброса принимающего процессора и полного останова его каналов обмена. Если сигнал сброса пришел во время обработки запроса к ОП, то выполнение сброса откладывается до завершения обращения к памяти.

Сигнал ВЫЗОВ вызывает внешнее прерывание по входу внешнего сигнала ВС-3 (прил. 8) в принимающем процессоре, если прерывания от внешних источников не замаскированы.

К мультисистемным сигналам относятся также сигналы внешней ошибки (ВО) и внешнего аппаратного сброса (СБР АПП). Сигнал ВО вырабатывается схемами контроля одного процессора и по отдельной линии передается в другой для воздействия на систему прерываний по входу внешнего сигнала ВС-2 на принимающем процессоре. Сигнал СБР АПП, аналогичный по действию сигналу СБР ПРГ, вырабатывается в передающем процессоре при нажатии кнопки «ЗАГРУЗКА».

В режиме МУЛЬТИСИСТЕМА команда ПРЯМАЯ ЗАПИСЬ используется для выработки мультисистемных сигналов, если в разрядах [12/15] командного слова (поле /2) содержатся нули, и для выработки сигналов прямого управления, если имеется «1» только в одном из разрядов [12/15] командного слова; при этом значения разрядов [8/11] не используются.

Команда ПРЯМОЕ ЧТЕНИЕ в режиме «Мультисистема» выполняется так же, как в обычном режиме прямого управления, за исключением того, что разряды [8/11] команды не используются, и синхроимпульсы по линиям ЛС-0—ЛС-3 не выдаются.

В режиме «Мультисистема» выполнение команды УСТАНОВИТЬ МАСКУ СИСТЕМЫ вызывает программное прерывание, если отсутствуют другие программные прерывания. Выполнение команды подавляется, и в старом ССП программного прерывания записывается код 0012 (прил. 8).

Блок прямого управления реализован на следующих ТЭЗ: усилитель интерфейса «УИ» — 3 шт.; регистр блока прямого управления «РБПРУ» — 2 шт.; мультисистема «МС» — 1 шт.

3.4. УПРАВЛЕНИЕ ОПЕРАТИВНОЙ ПАМЯТЬЮ

Центральный процессор и каналы ввода-вывода могут обращаться в ОП независимо и асинхронно. Для сопряжения процессора и каналов с ОП служит блок управления оперативной памятью.

Процессор и каналы по отношению к ОП являются независимыми устройствами и могут формировать свои запросы одновременно. БУ ОП обслуживает эти запросы по принципу «первым пришел — первым обслужен». При одновременном поступлении запросов запрос от канала обладает более высоким приоритетом. При поступлении запроса БУ ОП активизирует ОП, формирует сигналы, определяющие режим работы ОП (запись или чтение), выдает в ОП адрес обращения и обеспечивает прием информации из ОП в режиме чтения либо выдачу информации в ОП в режиме записи. При обслуживании запроса производится контроль с целью обнаружения трех типов программных ошибок: защиты памяти, адресации и спецификации. В состав БУ ОП входят (рис. 53):

узел тактовых импульсов (ТИ), узел запросов и очередности (ЗПР ОЧ), узел режимов (РЖ), узел выбора байт (ВБ), регистр информации процессора (РИП), регистр адреса канала (РАК), коммутатор выдачи адреса и информации (КВАИ), узел обнаружения ошибок обращения к оперативной памяти (ООП), узел памяти ключей защиты (ПКЗ).

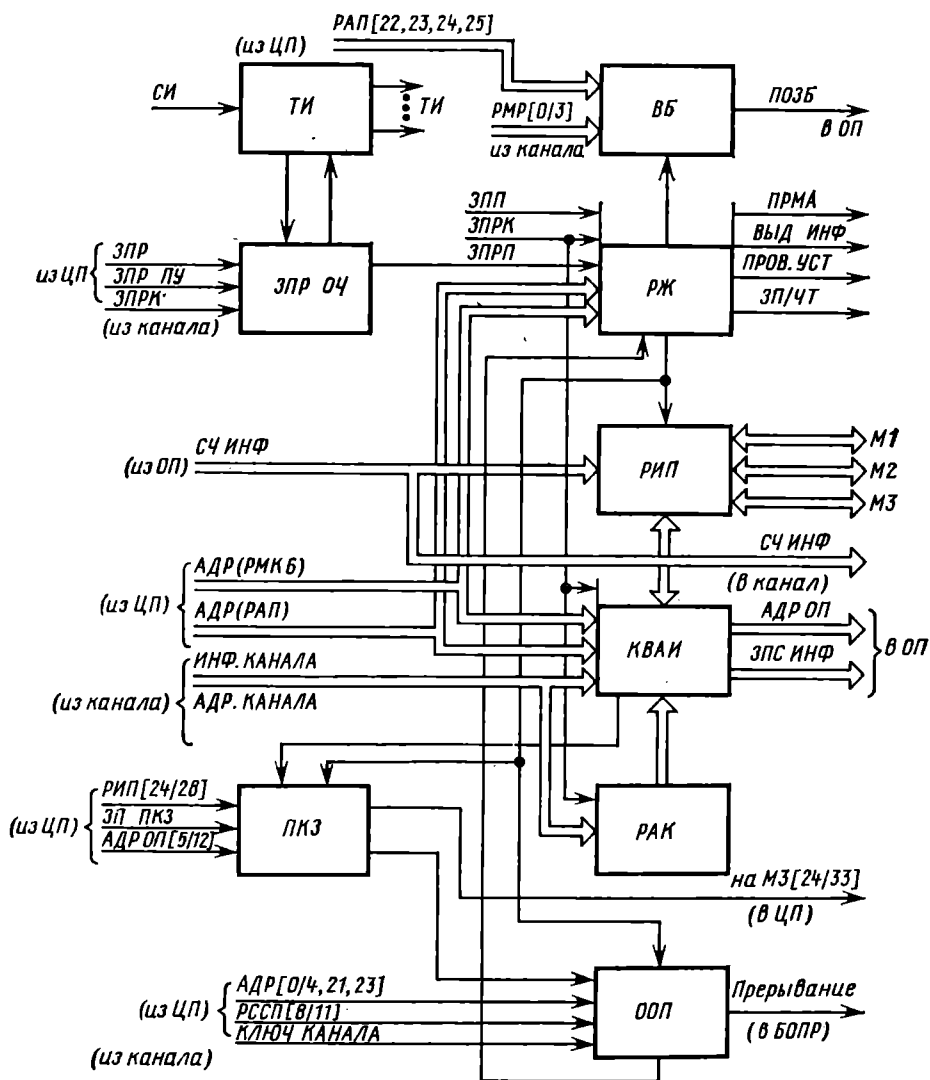


Рис. 53. Структурная схема БУ ОП

ЭВМ ЕС-1033 может работать с устройствами ОП различного типа (п. 1.9). В зависимости от типа и количества этих устройств емкость ОП машины может быть от 256К байт до 2048К байт.

Цикл работы процессора и каналов составляет 400 нс, цикл работы ОП — в четыре раза больше. Поэтому для синхронизации их работы необходимо выработать последовательность сигналов внутри каждого цикла обращения к ОП, привязанных к син-

хросериям машины. Эти же сигналы необходимы и для синхронизации работы всех составных частей БУ ОП (рис. 54).

Узел тактовых импульсов формирует ряд одиночных тактовых импульсов «1» ТИ1* — «1» ТИ4*, И1*—И4*, «1» ТИ2—«1» ТИ4, И2—И4, жестко привязанных во времени к синхросериям С1 и С2. При поступлении по серии С1 запроса от процессора или от общего канала (сигнал ЗПР) происходит установка по серии С2 в единичное состояние одного из триггеров запроса процессора (ТЗПР), либо триггера общего запроса каналов (ТОЗК). Сигналы с выходов всех этих триггеров поступают в схему приоритета, которая формирует сигнал запроса от процессора (ЗПРП). По времени и форме этот сигнал совпадает с выходным сигналом с ТЗПР или ТОЗК и разрешает по серии С1 запуск тактовой цепочки узла ТИ. Тактовые импульсы поступают во все узлы БУ ОП. Тактовая цепочка не может быть запущена вновь, если она еще не отработала до конца, — для этого служит триггер цикла (ТЦИКЛ).

Формирование запроса к ОП производит узел ЗПР ОЧ (рис. 55), который содержит четыре триггера — ТЗПР 000, ТЗПР 001, ТЗПР 011 и ТЗПР 111, предназначенных для запоминания запросов ЦП; триггер общего запроса каналов; триггер цикла (ТЦИКЛ); триггер очередности процессора (ТОЧП); схему приоритета; схему формирования запросов за операндами и триггер аварии питания (ТАВ ПИТ).

Запросы ЦП объединены в четыре группы: запрос информации формата двойного слова ЗПР 111 и ЗПР 111С; запрос информации однословного формата ЗПР 011 и ЗПР 011С; запрос информации, граница которой совпадает с границей полуслова ЗПР 001 и ЗПР 001С; запрос информации, которая может быть расположена в ОП, начиная с любого байта ЗПР 000 и ЗПР 000С. Такое разделение запросов вызвано необходимостью анализировать правильность спецификации при каждом обращении к ОП. Часть запросов является микропрограммными сигналами: ЗПР111, ЗПР011, ЗПР001, ЗПР000 и запрос выборки ЗПРВ, который при анализе спецификации квалифицируется как запрос за полусловом. Остальные сигналы запросов формируются схемно. ЗПРД вырабатывается в блоке диагностики ЦП. ЗПР ПУ вырабатывается при обращении к ОП с пульта управления. Опережающие запросы ЗПР 111С, ЗПР 011С, ЗПР 001С, ЗПР 000С вырабатываются

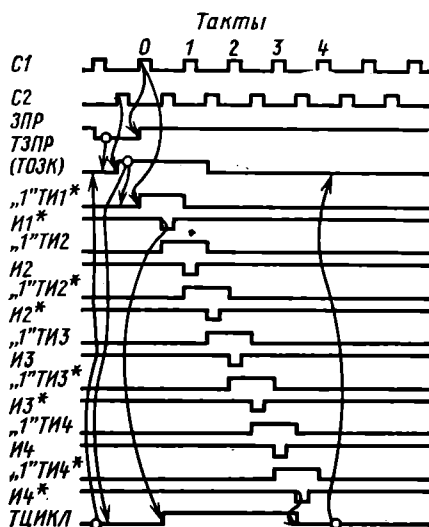


Рис. 54. Синхронизация БУ ОП

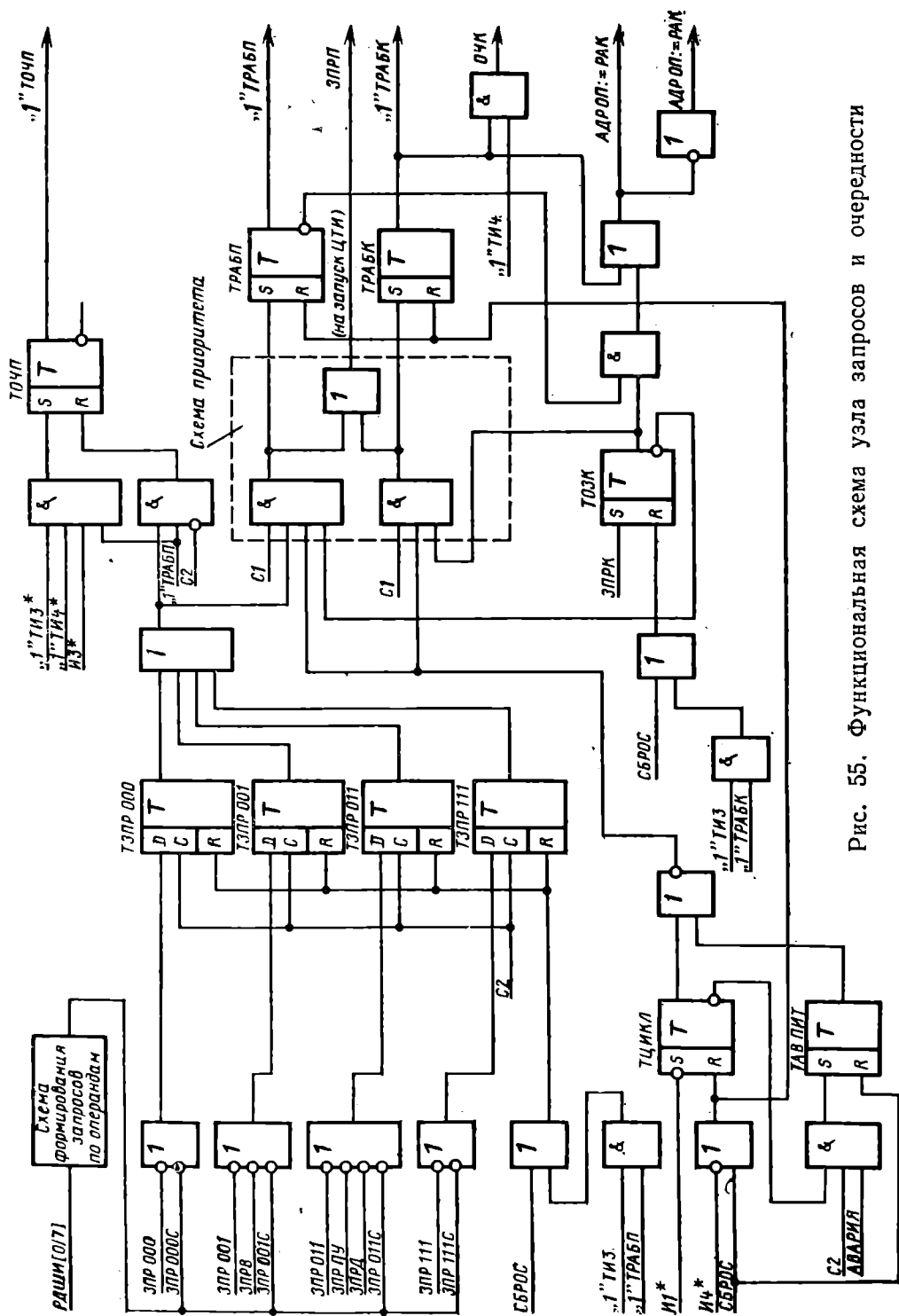


Рис. 55. Функциональная схема узла запросов и очереди

по микроприказу РАБОП := РДШК. Этим микроприказом производится передача кода операции из регистра дешифрации команд (РДШК) в регистр адреса БОП (РА БОП) перед выходом на выполнение тех операций, в которых операнды хранятся в ОП (кроме команд формата SS). Это возможно потому, что в ряде случаев код операции позволяет однозначно определить формат данных. Применение опережающих запросов сокращает время выполнения операции. Для каждой из четырех групп запросов ЦП существует свой триггер, в котором запоминается соответствующий запрос. Два запроса ЦП одновременно возникнуть не могут.

Сигналы запросов к ОП от всех каналов поступают в общий канал (ОК), координирующий работу селекторных и мультиплексного каналов. Он определяет приоритет обращения каналов к ОП, организует их очередность и вырабатывает все необходимые сигналы, поступающие в БУ ОП, который непосредственно связан только с ОК. Поэтому в БУ ОП не могут поступить одновременно сигналы запросов от разных каналов, и в дальнейшем имеет смысл в применении к работе БУ ОП говорить только об одном канале, имея в виду ОК.

Сигнал запроса канала (ЗПРК) запоминается триггером ТОЗК. Потенциалы с выходов триггеров запросов в ЦП поступают в собирающую схему. Сигнал с ее выхода поступает в схему приоритета и является сигналом наличия запроса от ЦП. Туда же, в схему приоритета, поступает сигнал с выхода ТОЗК.

Свободна ли ОП в данный момент времени, указывает состояние ТЦИКЛ. Он предназначен для предотвращения повторного запуска тактовой цепочки от вновь поступившего запроса, когда обслуживание предыдущего запроса еще не окончено.

Запрос от канала принимается к обслуживанию всегда, когда ТЦИКЛ находится в нулевом состоянии, т. е. ОП свободна. Запрос от ЦП принимается к обслуживанию, если ТЦИКЛ и ТОЗК находятся в нулевом состоянии, т. е. ОП не занята обслуживанием какого-либо запроса, и, кроме того, отсутствует запрос канала. При одновременном возникновении запросов от канала и от ЦП вначале обслуживается запрос канала, а запрос от ЦП запоминается на одном из четырех триггеров, соответствующем типу запроса, и ждет своей очереди.

Два триггера, входящие в узел режимов — триггер работы процессора (ТРАБП) и триггер работы канала (ТРАБК) — определяют принадлежность очередного цикла ОП соответственно центральному процессору или каналу. Их установка в единичное состояние происходит сигналами, формируемыми схемой приоритета одновременно с формированием сигнала ЗПРП. Сбрасываются эти триггеры в конце цикла по импульсу И4* тактовой цепочки.

Запрос от ЦП принимается к обслуживанию, если нет запроса от каналов (ТОЗК = 0) и ОП закончила предыдущий цикл (ТЦИКЛ = 0). В этом случае (рис. 56) одновременно в запуске тактовой цепочки устанавливается в единичное значение ТРАБП и формируется

сигнал приема адреса ПРМА, который поступает в ОП и инициирует ее внутренний цикл. Код адреса выдается в ОП постоянно, пока обслуживается данный запрос. Если необходимо произвести запись информации в ОП, то вместе с запросом микропрограммно устанавливается триггер записи процессора (ТЗП). При этом узел режимов выдает в ОП признак записи. Информация, подлежащая записи (ЗПС ИНФ), поступает в ОП через коммутатор выдачи адреса и информации из РИП и присутствует на информационных шинах ОП все время, пока поступает код адреса. В случае возникновения ошибки при записи информации ОП пере-

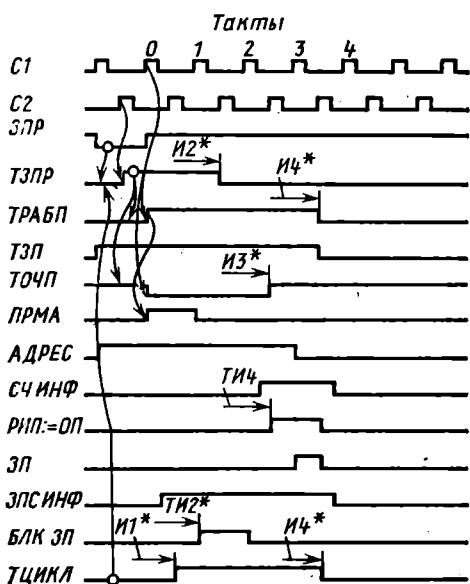


Рис. 56. Временная диаграмма работы процессора с ОП

обслужен, когда подойдет его очередь. О том, что имеется необслуженный запрос от ЦП, сигнализирует триггер процессора ТОЧП. Нормальное состояние его единичное. Он устанавливается в нулевое состояние по сигналу запроса от ЦП при условии, что в данный момент не обслуживается предыдущий запрос ЦП. В единичное состояние ТОЧП вновь устанавливается в цикле обработки запроса ЦП сигналами, формируемыми тактовой цепочкой БУ ОП, тогда, когда считанная из ОП информация будет принята в РИП или произойдет запись информации в ячейку памяти ОП. Состояние ТОЧП может быть проанализировано с помощью микроприказа ТОЧП \neq 0. Все микропрограммы ЦП построены таким образом, что выработке очередного запроса предшествует проверка наличия необслуженного запроса. Если ТОЧП=0, то это означает, что предыдущий запрос еще не обслужен. В этом случае продвижение микропрограммы приостанавливается (п. 1.6). Ее исполнение возобновляется после того, как

переводится из режима записи в режим чтения сигналом блокировки записи БЛК ЗП. Если ТЗП не установлен в единичное состояние, то запрос расценивается как запрос на считывание информации из ОП. В этом случае считанная из ОП информация (СЧ ИНФ) выдается не позднее, чем через 0,7 мкс после окончания сигнала ПРМА. Эта информация микроприказом РИП := ОП принимается в регистр информации процессора.

Если при поступлении очередного запроса от ЦП окажется, что предыдущий запрос от каналов еще не обслужен, то вновь поступивший запоминается в соответствующем триггере. Он будет

ТОЧП будет установлен в единичное состояние, тем самым разрешая использование находящейся в РИП информации, а также смену адреса обращения, находящегося в РАП, и формирование следующего запроса. Таким образом, ТОЧП совместно с ТЦИКЛ предотвращают наложение друг на друга запросов от ЦП.

Канал выдает свой сигнал запроса ЗПРК вместе с кодом адреса обращения к ОП. Последний поступает по информационным шинам канала из блока коммутации числовой информации ОК. По сигналу ЗПРК происходит запись кода адреса в РАК, после чего ОК производит замену кода адреса на информационных шинах канала записываемой в ОП информацией. Кроме того, по сигналу ЗПРК происходит установка в единичное состояние ТОЗК, ТРАБК и формирование сигнала ЗПРП для запуска тактовой цепочки аналогично тому, как это происходит при обслуживании запроса от ЦП. В конце обслуживания запроса от канала по импульсу ТИЗ происходит сброс ТОЗК, разрешая прием следующего запроса от канала, и по импульсу ТИ4 формируется сигнал очередности канала (ОЧК), информируя канал о том, что его запрос обслужен. Этот сигнал разрешает каналу сменить адрес обращения и выработать следующий запрос, а также разрешает прием в блок коммутации числовой информации ОК, считанной из ОП информации. Последняя поступает в канал из ОП по отдельным шинам.

Триггер аварии питания ТАВ ПИТ устанавливается в единичное состояние по сигналу из системы питания и блокирует очередной запуск тактовой цепочки БУ ОП. Текущий запрос обслуживается до конца. Сброс этого триггера происходит сигналом СБРОС при нажатии на ПУ кнопки «СБРОС СИСТЕМЫ».

Узел режимов содержит еще ряд триггеров, определяющих режим работы с ОП.

Триггер записи (ТЗП) и триггер записи байта (ТЗБ) задают режим записи при обращении к ОП со стороны ЦП и со стороны канала, а также режим записи, когда требуется записать в ОП только один байт. Установка этих триггеров в единичное состояние осуществляется микропрограммными сигналами, а сброс — в конце цикла ОП импульсами тактовой цепочки.

Триггер команды (инструкции) ПРОВЕРИТЬ и УСТАНОВИТЬ (ТИПУ) определяет режим записи единиц во все разряды указанного командой байта ОП.

В некоторых микропрограммах бывает необходимо заблокировать возможные программные прерывания в момент обращения к ОП. Этой цели служит триггер фиксированного адреса процессора (ТФАП). Для блокировки программных прерываний во время обращения канала к ОП канал выдает в БУ ОП сигнал «1» ТФАК. Его действие аналогично действию ТФАП при обращении от ЦП.

Триггер блокировки записи ТБЛ ЗП служит для блокировки записи в ОП при наличии какой-либо ошибки (как программной,

так и ошибки по четности). Если ТБЛ ЗП установлен в единичное состояние, по сигналу «1» ТИ2* в ОП выдается сигнал БЛ ЗП, который переводит ОП из режима записи в режим чтения.

Для связи ОП и ЦП в режимах чтения и записи информации служит регистр информации процессора РИП (см. рис. 53). Считанная из ОП информация принимается в РИП по сигналу $РИП := := ОП$. Затем она может быть выдана на любую магистраль по микроприказам $М1 := РИП$, $М2 := РИП$, $М3 := РИП$. Информация для записи в ОП также может быть дана в РИП с любой из магистралей по микроприказам $РИП := М1$, $РИП := М2$, $РИП := М3$.

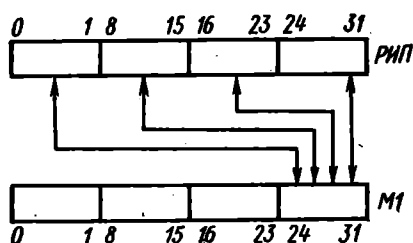


Рис. 57. Логические связи РИП (ЛРИП)

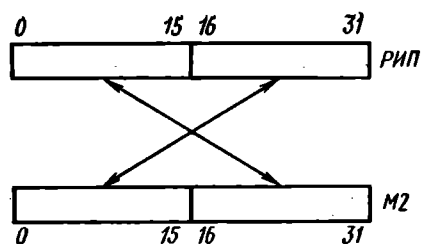


Рис. 58. Перекрестные связи РИП (ПРИП)

С магистралью М1 кроме обычной связи РИП связан еще и таким образом, что крайний правый байт М1 [24/31] может быть помещен на место любого байта РИП по микроприказу $ЛРИП := := М1 [24/31]$ и, наоборот, любой байт РИП может быть считан на М1 [24/31] по микроприказу $М1 [24/31] := ЛРИП$ (рис. 57). Номер нужного байта РИП задается в двоичном коде в буферном регистре сравнения (РСБФ), входящим в состав БФА БОП (п. 3.5), и расшифровывается специальным дешифратором.

Кроме обычной связи с магистралью М2, РИП связан с ней еще и таким образом, что младшее полуслово М2 может быть принято на место старшего полуслова РИП, а старшее полуслово М2 — на место младшего полуслова РИП. Это осуществляется по микроприказу $ПРИП := М2$. Возможно также чтение на М2 из РИП с взаимно перекрестной заменой полуслов по микроприказу $М2 := ПРИП$ (рис. 58). Аппараты ЛРИП и ПРИП широко применяются в микропрограммах ЦП (п. 5.2, 6.1).

Регистр адреса канала (см. рис. 53) предназначен для хранения адреса, по которому канал обращается к ОП. В этот регистр адрес поступает из коммутатора числовой информации общего канала и запоминается по сигналу ЗПРК. После переписи адреса в РАК адрес в коммутаторе замещается информацией, предназначенной для выдачи ее в ОП в режиме записи.

Адрес и информация выдаются в ОП через коммутатор КВАИ. Адрес может поступать в ОП по любому из трех направлений: из ЦП в регистра адреса процессора, с регистра адреса канала

и из блока диагностики ЦП с диагностического регистра микрокоманд РМК6. Информация в ОП поступает либо из РИП, либо из коммутатора числовой информации ОК.

Узел выбора байт предназначен для выработки сигналов-признаков, определяющих позиции записываемых в ОП байт ПОЗБ. В машине предусмотрена возможность записи в ОП как целого слова, так и его части. В последнем случае в ОП выдаются признаки тех байт, куда необходимо записать информацию. При этом в остальных байтах данного слова информация в ОП остается неизменной (п. 1.9).

Признаки — номера байт могут формироваться двояко: для нескольких подряд расположенных в слове байт и для одного отдельного байта. В первом случае задаются номера начального и конечного байтов. Для записи отдельного байта триггер записи байта (ТЗБ) устанавливается в единичное состояние и задается номер байта в качестве начального: номер конечного байта аппаратно формируется равным номеру начального.

Номер начального байта при запросе ЦП задается двумя младшими разрядами регистра адреса центрального процессора РАП[22/23], Номер конечного байта задается двумя дополнительными разрядами РАП[24/25]. Информация в РАП[24/25] поступает с магистрали М1 по микроприказу РАП[24/25] : = := М1 [30/31].

При обращении каналов к ОП номера начального и конечного байт поступают из регистра маркеров РМР, находящегося в канале. При этом разряды РМР[0/1] задают номер начального байта, а разряды РМР [2/3] — конечного.

В режиме чтения в ОП выдаются признаки всех четырех байтов независимо от типа запроса. Для сигналов позиций байта формируется контрольный разряд дополнением до нечетности и также выдается в ОП.

Узел обнаружения ошибок обращения к ОП (ООП, рис. 59) выявляет программные ошибки и формирует соответствующие сигналы, посылаемые в блок прерываний. Кроме того, в режиме чтения из ОП сигнал о любой из ошибок приводит к формированию узлом режимов сигнала БЛ ЗП, который посылается в ОП.

Сигнал ошибки защиты (ОШ ЗШ) формируется при обращении к ОП в случае несовпадения текущего ключа защиты с ключом, хранящимся в узле ПКЗ, входящем в состав БУ ОП (см. рис 53). Этот узел предназначен для записи, хранения и выдачи ключей защиты ОП. Вся оперативная память условно разделяется на страницы по 2048 байт. Каждой странице присваивается свой ключ защиты, хранящийся в ПКЗ. Младшие 11 разрядов кода адреса ОП являются адресом информации внутри страницы, а старшие разряды — адресом страницы и, соответственно, адресом ключа защиты. Ключ защиты имеет пять информационных разрядов и один контрольный. Четыре старших разряда являются собственно кодом ключа, а младший — признаком вида защиты.

Нулевое значение младшего разряда определяет защиту только по записи, единичное — по записи и чтению. Емкость ПКЗ составляет 256 байт для ОП емкостью 512К байт и 1024 байта для ОП емкостью 2048К байт. Конструктивно ПКЗ выполнена на ТЭЗ «ПКЗ», емкость одного ТЭЗ — 128 байтов.

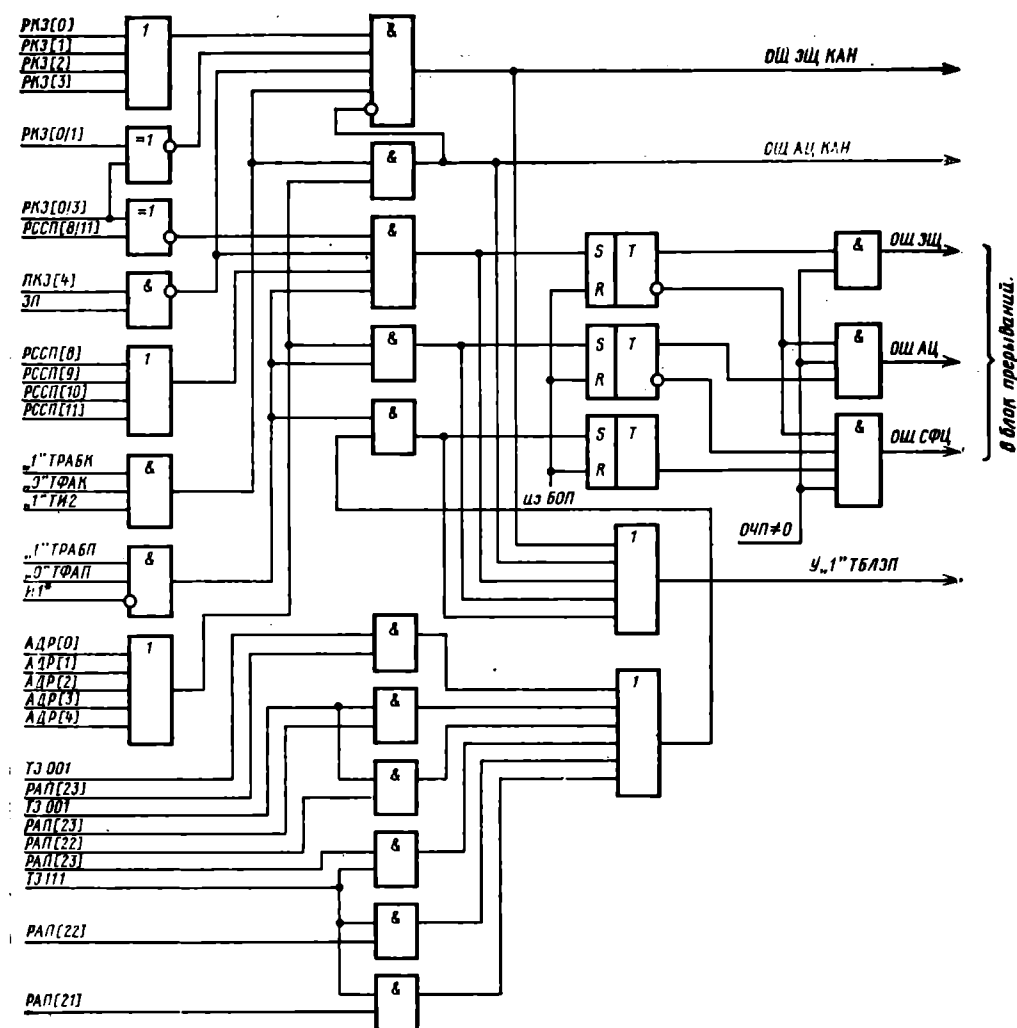


Рис. 59. Функциональная схема узла обнаружения ошибок обращения к ОП (для ОП емкостью 512К байт)

При обращении к ПКЗ формируется такой же запрос, как и при обращении к ОП, но при этом блокируется строб приема информации в РИП. На выходе ПКЗ всегда имеется код ключа, адрес которого поступает с коммутатора адреса ОП. Прием ключей на схему сравнения не строится, поэтому сравнение ключей осуществляется постоянно.

Запись в ПКЗ осуществляется по команде УСТАНОВИТЬ КЛЮЧ ПАМЯТИ. Информация при записи поступает из РИП

[24/28]. Чтение ключа из ПКЗ для записи его в РОН осуществляется по команде ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ. При этом ключ из ПКЗ считывается на магистраль МЗ по микроприказу МЗ [24/28] := РИПКЗ.

Текущий ключ защиты каналов ввода-вывода хранится в четырехразрядном регистре ключей защиты (РКЗ) каналов, куда он помещается из адресного слова канала АСК [0/3]. Текущий ключ защиты ЦП хранится в РССП [8/11].

При обращении каналов к ОП не по фиксированному адресу (ТФАК сброшен), ключ канала считывается из РКЗ [0/3] одновременно с приемом адреса в РАК и сравнивается с ключом из ПКЗ [0/3].

При обращении к ОП со стороны ЦП не по фиксированному адресу (ТФАП сброшен) ключ ЦП считывается из РССП [8/11] и сравнивается с ключом из ПКЗ [0/3].

ПКЗ [4] анализируется только при наличии сигнала записи. Нарушения защиты не происходит, если сравниваемые ключи либо равны, либо ключ каналов или ключ ЦП равен нулю. При несовпадении ключей формируется сигнал ошибки защиты канала или процессора (ОШ ЗАЩ КАН или ОШ ЗАЩ ЦП).

Ошибкой адресации при обращении к ОП считается появление адреса обращения, превышающего максимально возможный для конкретной емкости памяти машины. Схема на рис. 59 соответствует емкости ОП 512К байт. Так как в ЕС ЭВМ для адреса отводится 24 разряда, что соответствует максимально возможной емкости ОП, равной 16М байт, то при емкости 512К байт пять старших разрядов адреса памяти АДРОП [0/4] должны быть равны нулю. Эти разряды проверяются на значимость при каждом обращении к ОП и при наличии единицы хотя бы в одном из них формируются сигналы неправильной адресации процессора (ОШ АЦ) или канала (ОШ АЦ КАН). Каждый из этих сигналов вызывает блокировку записи в ОП.

Ошибкой спецификации (ОШ СФЦ) считается нарушение целочисленной границы информации. В зависимости от типа запроса ЗПР 111, ЗПР 011 или ЗПР 001 проверяются на значимость разряды РАП [21/23], РАП [22/23] или РАП [23] соответственно. Ошибка спецификации при запросе от канала не проверяется, так как эта проверка осуществляется в самом канале.

При обращении центрального процессора к ОП сигналы ошибок выдаются в блок прерываний по микроприказу ТОЧП $\neq 0$, который анализирует состояние триггера ТОЧП. Значение ТОЧП $\neq 0$ свидетельствует о том, что запрос процессора обслужен. Сброс триггеров, фиксирующих ошибки при обращении от ЦП, происходит либо по сигналу СБРОС, либо по микроприказу блокировки ошибок ОП (БЛ ОШ ОП). Так как при одном обращении к ОП могут иметь место ошибки всех трех типов, то для определенности сигналы ошибок выдаются в приоритетном порядке. Наивысшим приоритетом обладает ошибка

защиты, наименьшим — ошибка спецификации. Приоритет ошибок устанавливается схемно.

При обращении канала к ОП ошибка адресации имеет более высокий приоритет, чем ошибка защиты.

Блок управления ОП реализован на следующих ТЭЗ: регистр дешифрации команды РДШК (содержит схему опережающих запросов) — 1 шт.; коммутатор-передатчик КПД (содержит КПД и РАК) — 6 шт.; регистр информации процессора РИП — 8 шт.; контрольные разряды РИП «РИПКР» — 4 шт.; БУОП1, БУОП2, БУОП3, БУОП4, БУОП5 — по 1 шт., а также ТЭЗ ПКЗ, количество которых зависит от емкости ОП.

3.5. БЛОК МИКРОПРОГРАММНОГО УПРАВЛЕНИЯ

Магистральная организация связей между блоками центрального процессора определила и некоторые особенности его микропрограммного управления (МУ), которое в данном случае должно обеспечить:

параллельное независимое управление заданием функций обрабатываемых блоков;

независимое управление подключением каждого блока к различным магистралям;

независимое управление работой информационных магистралей;

управление логической обработкой информации при ее передаче по магистралям;

развитую и гибкую логику выбора дальнейшего пути выполнения микропрограммы.

Выполнение этих требований наложило отпечаток как на аппаратуру блока микропрограммного управления, так и на структуру микрокоманд.

Микропрограммное управление процессором в своей основе имеет традиционную схему и состоит из памяти микропрограмм БОП0 — БОП3 (рис. 60), системы дешифрирования содержимого операционных полей и выработки микроприказов (сигналов микроопераций), входящей в состав блока режимов работы и блока формирования адреса следующей микрокоманды (адреса БОП), который осуществляет выбор пути дальнейшего выполнения микропрограммы.

Память микропрограмм процессора состоит из четырех унифицированных блоков односторонней памяти БОП 2048-33. Каждый блок имеет емкость 2048 33-разрядных слов. Слово содержит 32 информационных разряда и один контрольный (пп. 1.7, 2.1). Очередная микрокоманда считывается по одному и тому же адресу одновременно из всех четырех блоков. Таким образом, микрокоманда ЦП имеет 128 информационных разрядов и разбита на четыре 32-разрядных слова, каждое из которых считывается из своего БОП (БОП0—БОП3) и выдается в регистр микрокоманд

РМК (РМК0—РМК3). Из общего количества в 2048 таких 128-разрядных слов основные микропрограммы для выполнения команд используют около 1600 слов. Остальные слова управляющей памяти используются наладочными, контрольными, пультовыми микропрограммами и процедурами микродиагностики.

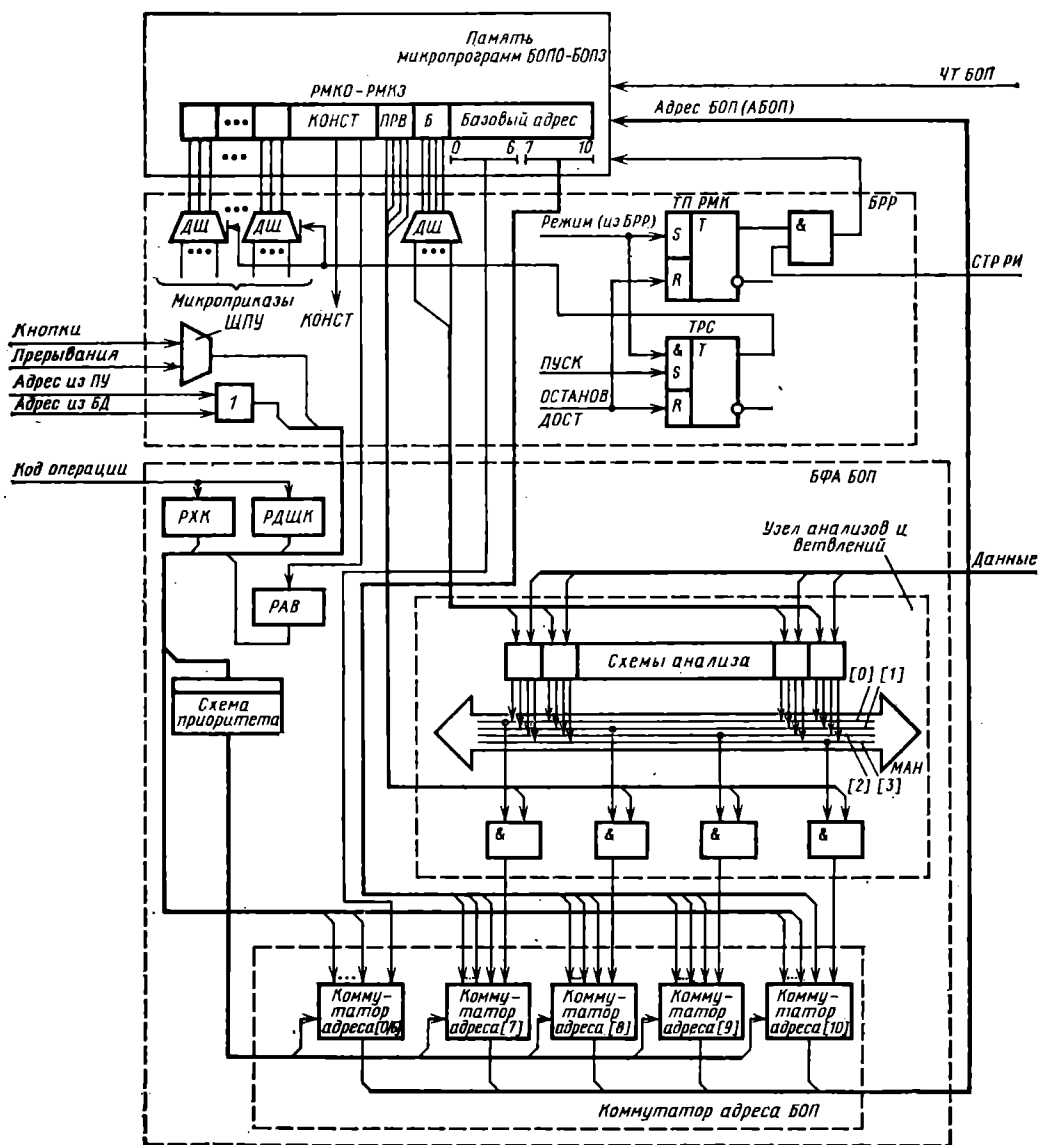


Рис. 60. Структурная схема блока микропрограммного управления

Необходимость обеспечить независимое управление параллельно работающих блоков и возможность организации связей между блоками по трем магистралям обусловили необходимость довольно большого формата управляющего слова. Однако за счет получения высокого параллелизма работы и экономичности алгоритмов количество слов, используемых при этом, удалось значи-

тельно уменьшить. В управляющем слове используется как кодированное, так и позиционное управление.

Нулевое слово микрокоманды, которое считывается в РМК0 (рис. 61), содержит шесть пятиразрядных полей управления чтением информации из регистров ЦП на три информационных магистрали и записью информации в блоки и регистры с магистралей. Два последних разряда не используются. Первые два символа обозначения каждого поля указывают номер магистрали (М1—М3), а два других — выполняемое действие — чтение или запись. Каждое поле адресует либо 16

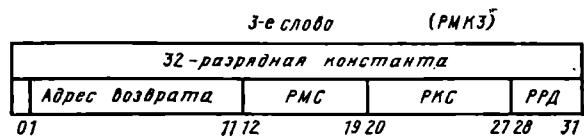
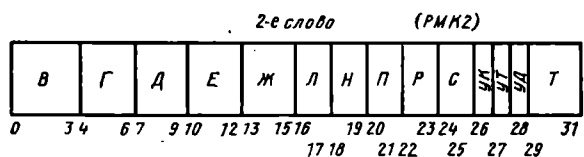
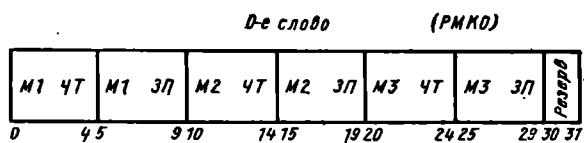


Рис. 61. Структура микрокоманды процессора

что дает возможность осуществить $961 \times 3 = 2883$ различных трактов передачи информации между блоками процессора. При выполнении алгоритмических действий нужны различные сочетания таких связей, общее количество которых равно:

$$C = \frac{n!}{r!(n-r)!} = \frac{961!}{3!(961-3)!} = 0,5 \cdot 10^6,$$

где C — возможное число вариантов настройки структуры; n — число возникших связей по каждой магистрали; r — число магистралей.

В первом слове микрокоманды, которое считывается в РМК1, содержится поле базового адреса следующей микрокоманды (РМК1[0/10]) и поля ПРВ (РМК1[11/14]), А (РМК1[16/18]) и Б (РМК1[19/24]). Последние три используются в БФА БОП для выбора направления продолжения микропрограммы.

В поле S (РМК1 [25/29]) содержатся коды, определяющие режимы работы БАЛ и вид выполняемых в нем действий. В БАЛ могут выполняться 15 логических и 30 арифметических микроопераций. Одна комбинация, соответствующая $S = D$, запрещает выдачу результата из блока на магистраль $M1$. Коды поля S для различных микроопераций, выполняемых БАЛ, приведены в табл. 7.

В первом слове кроме поля $ПРВ$ имеются три позиционных поля, наличие единиц в каждом из которых определяет соответствующий микроприказ. РМК1 [15] (поле $ЗП$) определяет наличие запроса к ОП за байтом информации. РМК1 [30] содержит микроприказ $ТОЧП \neq 0$ анализа триггера очередности (пп. 2.1, 3.4). Единичное состояние этого триггера свидетельствует о том, что выставленный процессором запрос к ОП обслужен. Эти микроприказы были обособлены в связи с асинхронностью работы абонентов с ОП. Запрос к ОП на обслуживание может быть выдан ЦП в любой микрокоманде, после чего продолжается выполнение микропрограммы. Когда необходимо использовать данные из ОП или необходимо убедиться, что запись результата в ОП выполнена, в очередной микрокоманде выставляется микроприказ $ТОЧП \neq 0$. Поскольку цикл работы ОП в четыре раза больше такта ЦП, запрос выдается обычно за 4—5 тактов до использования данных. Вследствие такого построения микропрограмм микроприказы работы с ОП оказываются слабо связанными с другими микроприказами и поэтому они выделены для позиционного кодирования.

Позиционный микроприказ ВКЛ ПА содержится в поле $ПА$ (РМК1 [31]) и предназначен для включения режима переадресации РОП (п. 2.3). Этот режим используется при обработке данных переменной длины и разрешает переадресацию РОП не от полей нулевого слова микрокоманды, а от адресного регистра-счетчика.

Второе слово микрокоманды, которое считывается в РМК2, выполнено в традиционной системе кодированного управления и содержит ряд операционных полей $B, Г, Д, E, Ж, Л, Н, П, P, C, T, УК, УТ, УД$.

Третье слово микрокоманды, которое считывается в РМК3, отражает особенности магистральной организации ЦП. Все 32 разряда этого слова могут быть представлены как полноразрядная константа и поданы на любую из магистралей ЦП. Таким образом, они могут участвовать в вычислениях наряду с данными, а также записываться в ОП, сверхоперативную память и т. п. Наиболее интенсивно полноразрядная константа используется как маска, накладываемая на информацию, участвующую в вычислениях.

Содержимое третьего слова используется и для непосредственной записи констант в регистры ЦП, т. е. минуя информационные магистрали. Содержимое РМК3 [1/11] может быть записано в регистр адреса возврата и таким образом использоваться для

организации выхода в основную микропрограмму из микроподпрограммы. Два байта (поля *PKC* и *PMS*) служат для организации ветвления микропрограмм по анализам сравнения. При этом содержимое поля *PKC* является эталоном, с которым производится сравнение информации, а содержимое поля *PMS* — маской, разрешающей поразрядное сравнение. При ветвлении учитывается результат сравнения части эталона и информации, выделенные маской.

РМКЗ [28/31] используются для установки значения регистра результата деления в микропрограммах команд с фиксированной и плавающей запятой.

Наличие единицы в поле *УК* (РМК2 [26]) свидетельствует о том, что содержимое третьего слова следует воспринимать как константу. Нулевое значение поля *УК* позволяет использовать РМКЗ [1/3] и РМКЗ [4/7] как дополнительные кодированные поля микроприказов *X* и Φ .

Для считывания микрокоманды в блоки БОПО—БОПЗ (см. рис. 60) подаются код адреса микрокоманды (адрес БОП) и сигналы ЧТ БОП и СТР РИ (п. 1.7). Последний дополнительно стробируется сигналом с единичного выхода триггера приема информации РМК—(ТПРМК). Этот триггер переключается в зависимости от режима работы ЦП (автомат или такт), а также блоком диагностики ЦП. Дешифраторы, вырабатывающие микроприказы от кодов полей микрокоманды, стробируются триггером рабочего состояния (ТРС). Этот триггер устанавливается в зависимости от режима работы ЦП по сигналу ПУСК, вырабатываемому от кнопки «ПУСК» с пульта управления. Сброс этих триггеров происходит по сигналу ОСТАНОВ или по сигналу останова ДОСТ из блока диагностики. Дешифратор поля анализов *B* не стробируется (т. е. активизирован постоянно).

В состав БРР входят также следующие схемы: шифратор пульта управления (ШПУ), который формирует адреса начальных микрокоманд микропрограмм обработки прерываний по сигналам прерываний и от кнопки «ПРЕРЫВАНИЕ» с пульта управления (п. 3.2) и адреса начальных микрокоманд пультовых процедур при нажатии соответствующей кнопки (п. 3. б); схема для передачи адреса микрокоманды, набранного на пульте управления, или адреса микрокоманды ЦП, формируемые блоком диагностики.

На рис. 60 128-разрядный РМК показан условно. Выделены лишь операционные поля, подключенные к дешифраторам микроприказов, поле константы, поля *ПРВ* и *B*, принимающие участие в работе аппарата анализа и поле базового адреса следующей микрокоманды.

БФА БОП предназначен для формирования адреса микрокоманды в соответствии с выполняемым алгоритмом и в зависимости от обрабатываемых данных и получаемых результатов [а. с. 613401 (СССР)]. В его состав входят ряд регистров, узел анализа и ветвлений, узел коммутатора адреса.

Последний служит для выдачи в память микропрограмм сформированного адреса следующей микрокоманды. В коммутатор адреса код адреса может поступать с нескольких направлений. При нажатии кнопки на пульте управления сигнал от нее поступает в ШПУ, который формирует код адреса соответствующей микропрограммы. Этот код поступает через коммутаторы в блок памяти, из которого начинает выбираться последовательность микрокоманд, осуществляющая заданное с пульта действие (исполняться пультовая микропрограмма). При выполнении процессором программы начальный адрес последовательности микрокоманд определяется кодом команды (кодом операции). При выборке команды ее код операции заносится в регистр дешифрации команды, а при опережающей подвыборке — в регистр хранения команды. По завершению подготовительных действий начальный адрес микропрограммы данной команды, численно равный ее коду операции, подается через коммутаторы на адресные входы БОП. Начинает выбираться микропрограмма данной команды.

При исполнении микропрограммы очередная микрокоманда считывается из БОП и поступает в регистр микрокоманд. Часть ее разрядов, составляющая поле базового адреса, выдается в коммутатор и через него на адресные входы блока памяти. Следующая микрокоманда будет выбрана по адресу, заданному в текущей микрокоманде. Так осуществляется управление в последовательных ветвях микропрограмм.

Последовательность выборки и исполнения микрокоманд может быть изменена при прерывании. Необходимость срочно прервать нормальное течение микропрограммы возникает при различных программных ошибках, таких, как нарушение защиты памяти, обращение в ОП по несуществующему адресу, неверное задание адресов команд и т. п. Сигнал прерывания поступает также в ШПУ. В зависимости от характера прерывания последний вырабатывает тот или иной адрес, который определяет микропрограмму реакции на соответствующую ситуацию. Через коммутаторы адрес поступает в БОП. В этот момент поступление других адресов через коммутаторы блокируется.

При необходимости вызвать с пульта управления какую-либо микрокоманду из блока памяти ее адрес, набранный на клавиатуре пульта, подается на входы коммутаторов одновременно с управляющим сигналом, и нужная микрокоманда выбирается из блока памяти. Имеется возможность управлять работой центрального процессора с помощью блока диагностики. Задавая адреса микрокоманд процессора, он осуществляет выполнение соответствующей диагностической процедуры (п. 10.4). Для выхода из вспомогательных микропрограмм (микроподпрограмм) в основную используется адрес возврата. В основной микропрограмме он записывается в регистр адреса возврата перед входом в микроподпрограмму. Через коммутатор адрес возврата посту-

пает в БОП. Адрес возврата может быть модифицирован схемами анализа.

Так как адрес следующей микрокоманды может одновременно формироваться в различных блоках и поступать с нескольких направлений, то коммутаторы адреса управляются приоритетной схемой. На выходе коммутатора может быть только один адрес, имеющий высший приоритет из поступающих на вход в данный момент. Адрес с пульта управления имеет наивысший приоритет. Поэтому оператор может вмешиваться в работу машины в любое время. Следующими по приоритету идут: адрес БОП, поступающий из блока диагностики, адреса из ШПУ, РАВ, РХК или РДШК, с магистрали анализов и, наконец, из РМК (базовый адрес).

Узел анализа и ветвлений предназначен для модификации четырех последних разрядов базового адреса следующей микрокоманды, заданного в текущей микрокоманде. Модификация осуществляется в зависимости от обрабатываемых данных и получаемых результатов. Возможна также модификация содержимого РАВ.

Высокая скорость выполнения операций процессором обусловлена применением компактных и эффективных алгоритмов. Однако небольшая глубина алгоритма требует высокой его разветвленности и быстрой реакции на возникшие в процессе вычислений ситуации. В большинстве микрокоманд осуществляется анализ данных и результатов с целью выбора пути продолжения алгоритма. Схемы анализа и ветвлений дают возможность разветвления алгоритма по 16 направлениям. Сложность реализации таких процессов заключается в том, что результат анализа должен быть учтен при формировании адреса следующей микрокоманды, которое заканчивается к началу синхросерии С2 текущего такта. Если к нужному сроку получить результат анализа не удается (например, учесть полученный на выходе БАЛ результат), то принимаются специальные меры для запоминания условий анализа с возможностью последующего использования, а алгоритм строится таким образом, чтобы темп обработки не замедлялся.

В каждом конкретном случае проверяется свой набор условий. Хотя общее число таких проверок для разнообразных условий достаточно велико, набор условий, проверяемых одновременно в одной микрокоманде, ограничен. Это позволило скомпоновать проверяемые условия в стандартные наборы. Результаты их анализа выдаются на четырехразрядную шину, называемую магистралью анализов (МАН). Номер анализа задается полем Б РМК1. Этот номер дешифрируется, и сигнал с выхода дешифратора подключает к МАН выход соответствующей схемы анализа.

Через МАН результат анализа поступает в четыре младших разряда коммутатора адреса, подменяя одноименные разряды базового адреса, приходящего из РМК. Старшие разряды базового адреса выдаются в БОП через коммутатор, не претерпевая

изменений. Таким образом возможно разветвление на 16 направлений в пределах «страницы» блока памяти (группы ячеек, адресуемых неизменной частью адреса). Однако ветвление каждый раз на 16 направлений нерационально, так как требует большого количества ячеек памяти, информация в которых должна повторяться. Поэтому количество направлений ветвления ограничивается с помощью других разрядов микрокоманды — полем признака ветвления — РМК1 [11/14]. Наличие единицы в каком-либо разряде этого поля разрешает подключение через схемы совпадения в качестве значения разряда следующего адреса содержимого соответствующего разряда магистрали анализов. Отсутствие признака ветвления в данном разряде блокирует прием разряда адреса с МАН. В этом случае используется значение разряда адреса, поступающего из РМК. При единице в одном разряде поля ПРВ возможно разветвление на два направления, при единицах в двух разрядах — на четыре направления и т. д. Таким образом, использование одного анализа в сочетании с применением признаков ветвления позволяет, по сути дела, выполнять от двух до 15 видов проверок (нулевое значение признака ветвления есть отсутствие анализа).

Группу основных анализов составляют анализы с 16 по 45 (табл. 17). Номер анализа задается кодом поля Б в РМК1. При этом формируется соответствующий микроприказ (например, АН16). По микроприказу анализа в разряды МАН [0/3] считывается либо содержимое разрядов регистра или магистрали, которые указаны в строке анализа, либо в соответствующем разряде МАН устанавливается единица при выполнении условий, указанных в строке анализа.

При выполнении анализов с 16 по 26 на МАН [0/3] считывается значение четырех разрядов РДШК, РАВ, счетчика РОН или магистралей М1, М2 и М3. При выполнении анализа 27 на МАН [0] выдается значение РДШК [3]. На МАН [1], МАН [2] и МАН [3] устанавливаются единицы (МАН [1] = 1, МАН [2] = 1, МАН [3] = 1), если соответственно значения всех разрядов М1, М2 и М3 будут равны нулю. При выполнении анализа 28 МАН [0] и МАН [1] равны единице, когда выполняются условия $M2[1/7] = M3[1/7]$ и $M2[0] \neq M3[0]$; МАН [2] принимает значение $M3[0]$, а МАН [3] — значение переноса сумматора.

Условия остальных анализов аналогичны и указаны в соответствующих строках табл. 17; ниже будут пояснены некоторые из них.

Анализ 33. МАН [1] = 1, если на ПУ нажата клавиша «МУЛЬТИСИСТ»;

Анализ 38. Применяется при выборке команд и, в отличие от остальных, является приоритетным (п. 4.4);

Анализ 40. МАН [1] = 1, если на М2 [24/27] находится код цифры, большей 9; МАН [2] = 1, если на М2 [28/31] находится код знака «плюс» и МАН [2] = 0, если там любой другой код;

Основные анализы

| Номер анализа | Условия анализа по разрядам МАН | | | | Шестнадцатиричный код поля Б |
|---------------|--|--------------------------|----------|------------|------------------------------|
| | 0 | 1 | 2 | 3 | |
| 16 | РДШК[0/3] | | | | 10 |
| 17 | РДШК[4/7] | | | | 11 |
| 18 | РАВ[7/10] | | | | 12 |
| 19 | СТ1РОН | | | | 13 |
| 20 | СТ2РОН | | | | 14 |
| 21 | М1[0/3] | | | | 15 |
| 22 | М2[0/3] | | | | 16 |
| 23 | М3[0/3] | | | | 17 |
| 24 | М1[28/31] | | | | 18 |
| 25 | М2[28/31] | | | | 19 |
| 26 | М3[28/31] | | | | 1A |
| 27 | РДШК[3] | М1=0 | М2=0 | М3=0 | 1B |
| 28 | М2[1/7]= =М3[1/7] | М2[0]≠М3[0] | М3[0] | ПСМ | 1C |
| 29 | Анализ нормализации в операциях сложения и вычитания | | | | 1D |
| 30 | М1[0] | М3=0 | М2[0] | М3[0] | 1E |
| 31 | Анализ нормализации в операциях умножения и деления | | | | 1F |
| 33 | М3[0] | Клавиша «Мультисист.» | М2[0] | М2[1/31]=0 | 21 |
| 34 | СТРОН=0 | М3=0 | СТ1РОН=0 | СТ2РОН=0 | 22 |
| 35 | М2[0] | СТЦ=0 | СТЦ1=0 | СТЦ2=0 | 23 |
| 36 | Анализ цикла деления | | | | 24 |
| 37 | Анализ цикла умножения | | | | 25 |

| Номер анализа | Условия анализа по разрядам МАН | | | | Шестнадцатиричный код поля Б |
|---------------|---------------------------------------|-----------------|------------------------|---------------------|------------------------------|
| | 0 | 1 | 2 | 3 | |
| 38 | РССП[62]=1 | РХК[0/1]=00 | ТЗРХК=1 | РФПД≠0 | 26 |
| 39 | Анализ неуспешной передачи управления | | | | 27 |
| 40 | СТ1РОН= =СТ2РОН | М2[24/27] >9 | М2[28/31]= =ЗН «+» | М2[28/31]= =ЗНАК | 28 |
| 41 | Микроприказ РАБОП: = РДШК | | | | 29 |
| 42 | Анализ РФПД | | | | 2А |
| 43 | М1[28/31]= =ЗН « » | ЗНМ1=ЗНМ2 | Идентификатор действия | | 2В |
| 44 | СТ2РОН=0 | ПСМ(Д) | СДВ ЛЕВ | | 2С |
| 45 | «0» СМ | СТ1РОН=0 | СТ2РОН=0 | ПСМ | 2D |
| 54 | Передача управления БД | | | | 36 |
| 55 | Передача управления каналам | | | | 37 |
| 56 | КОМАНДЫ 08, 09 | | | | 38 |

МАН [3] = 1, если на М2 [28/31] находится код знака «плюс» или «минус». Этот анализ применяется в операциях редактирования (п. 8.4);

Анализ 43. МАН [1] = 1, если равны коды знаков операндов, расположенных на магистралях М1 и М2; на МАН [2/3] устанавливается код, значение которого зависит от кода операции (РДШК [6/7]) и знаков операндов ЗНМ1 и ЗНМ2 и идентифицирует действия сложения и вычитания. Этот анализ применяется в операциях десятичной арифметики (п. 9.1);

Анализ 44. МАН [1] принимает значение переноса сумматора при выполнении операций десятичной арифметики (п. 9.3); на МАН [2/3] считывается признак направления сдвига, который формируется по содержимому М2 [30/31] и М3 [30/31] (п. 2.5);

Анализ 45. МАН [0] = 1, если содержимое сумматора равно нулю.

Анализы 29, 31, 36, 37, 39, 42 отличаются тем, что по микроприказу любого из них выполняется сразу комплекс проверок. Первые четыре анализа используются в операциях с плавающей

запятой. Результат анализа 29 показывает, на сколько тетрад надо нормализовать операнд двойной длины, находящийся на M2 и M3, или операнд длиной в слово, находящийся на M2 (п. 7.2). Результат анализа 31 дает сравнительную оценку нормализации двух операндов, расположенных на магистралях M2 и M3 (пп. 7.3, 7.4). Результат анализа 36 дает возможность анализировать ход деления и обеспечивает своевременный выход из этого цикла (пп. 7.3, 7.4). Результат анализа 37 определяет состояние сомножителей и промежуточного произведения, а также обеспечивает выход из цикла (п. 7.3). Анализ 39 применяется при выборке команд для определения правильности задания команд или данных, для определения неуспешности переходов и для выбора направления ветвления по формату команд (п. 4.4). Анализ 42 используется для определения класса прерываний (пп. 3.2, 4.3).

Анализы 41, 54, 55 и 56 отсутствуют. По соответствующим им микроприказам выполняются следующие действия: АН41 — передача в РА БОП кода операции при выходе на микропрограмму операции после выборки команды; АН54 — передача управления блоку диагностики ЦП при выполнении команды ДИАГНОСТИКА (пп. 10.3, 10.6); АН55 — передача управления каналам ввода-вывода при выполнении команды ДИАГНОСТИКА (п. 13.5); АН56 — блокировка ошибок защиты памяти при выполнении команд УСТАНОВИТЬ КЛЮЧ ПАМЯТИ и ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ.

Чтобы проиллюстрировать формирование адреса следующей микрокоманды с помощью аппарата анализов, рассмотрим примеры выполнения некоторых из них.

Пример. Допустим, что необходимо произвести ветвление микропрограммы по содержимому РДШК [0/3]. Для этого используется анализ 16 с признаком ветвления, равным 1111, т. е. F. Микроприказ, по которому выполняется этот анализ, записывается так: АН16(F). Базовый адрес следующей микрокоманды должен быть следующего вида: X.Y.0000, где X и Y — трех- и четырехразрядные двоичные коды, а последние четыре модифицируемых разряда должны быть равны нулю. При РДШК[0/3] = KLMN, где K, L, M и N — любые значения двоичных разрядов, образующие коды от 0 до F, схема модификации адреса будет выглядеть следующим образом:

| | |
|---------------|-----------|
| базовый адрес | X.Y.0000, |
| РДШК[0/3] | KLMN, |
| МАН[0/3] | KLMN, |
| ПРВ | 1111, |
| адрес БОП | X.Y.KLMN, |

т. е. возможно формирование шестнадцати адресов от X.Y.0000 до X.Y.1111.

Если необходимо произвести ветвление микропрограммы на четыре направления по значению РДШК[1/2], то признак ветвления должен быть равен 0110, т. е. 6. В этом случае используется микроприказ АН16(6). Базовый адрес должен быть такого вида: X.Y.P00Q, а значение РДШК[0/3] — Z00Z, Z01Z, Z10Z, или Z11Z, где P, Q и Z — любые значения двоичных разрядов. По микроприказу АН16(6) на МАН[0/3] считывается значение РДШК [0/3],

но в модификации участвуют только 1-й и 2-й разряды. Модификация происходит по схеме:

| | | | | |
|---------------|----------------|----------------|----------------|----------------|
| базовый адрес | а) X. Y. P00Q, | б) X. Y. P00Q, | в) X. Y. P00Q, | г) X. Y. P00Q, |
| РДШК [0/3] | Z00Z, | Z01Z, | Z10Z, | Z11Z, |
| МАН [0/3] | Z00Z, | Z01Z, | Z10Z, | Z11Z, |
| ПРВ | 0110, | 0110, | 0110, | 0110, |
| адрес БОП | X. Y. P00Q, | X. Y. P01Q, | X. Y. P10Q, | X. Y. P11Q. |

Пример. Допустим, что необходимо произвести ветвление микропрограммы в зависимости от позиции байта в слове. Адрес байта обычно хранится в РОП15. В микропрограмме производится его считывание из РОП15 на магистраль, например на М1. Так как код позиции байта оказывается на М1[30/31], можно использовать микроприказ АН24(3). При обозначениях, принятых выше, модификация будет выполняться по следующей схеме:

| | | | | |
|---------------|----------------|----------------|----------------|----------------|
| базовый адрес | а) X. Y. PQ00, | б) X. Y. PQ00, | в) X. Y. PQ00, | г) X. Y. PQ00, |
| РОП15 [28/31] | ZZ00, | ZZ01, | ZZ10, | ZZ11, |
| МАН [0/3] | ZZ00, | ZZ01, | ZZ10, | ZZ11, |
| ПРВ | 0011, | 0011, | 0011, | 0011, |
| адрес БОП | X. Y. PQ00; | X. Y. PQ01; | X. Y. PQ10; | X. Y. PQ11. |

По результатам анализа 18 возможен переход по адресу возврата; при этом последняя его тетрада, находящаяся в РАВ [7/10], может быть модифицирована обычным образом. Возможен также безусловный переход по адресу возврата. Это происходит при наличии в микрокоманде двух микроприказов: АН18 и ВБАВ (ветвление безусловное по адресу возврата, пп. 4.2, 6.1). Существует еще одна возможность перехода по содержимому РАВ—при использовании анализов сравнением АН СР2.

Табличные регистры анализа ТРА1—ТРА7 (табл. 18) представляют собой шесть четырехразрядных регистров [а. с. 591075 (СССР)]. Регистр ТРА5 для анализа не используется, его применение описано в пп. 2.3 и 3.5. Состояние всех четырех триггеров, входящих в регистр, может быть проанализировано одним из

Т а б л и ц а 18

Табличные регистры анализа

| Номер анализа | Условия анализа по разрядам МАН | | | | Код поля Б |
|---------------|---------------------------------|-------|-------|---------|------------|
| | 0 | 1 | 2 | 3 | |
| 1 | Отсроченный анализ | | | | 01 |
| 2 | Отсроченный анализ | | | | 02 |
| 3 | ТИПУ | ТФАП | ТЗПКЗ | ТИВ | 03 |
| 4 | ТНЗ | ТЗПРВ | ТВ/В | ТБ | 04 |
| 5 | Специальное применение | | | | 05 |
| 6 | ТР | ТКЛ1 | ТДК | ТПС | 06 |
| 7 | ТОР | ТЗНАЧ | ТЗН1 | ТЗН «+» | 07 |

микроприказов АН1—АН7 (кроме АН5). При этом разряды МАН принимают значения, соответствующие состояниям триггеров регистра, т. е. происходит чтение содержимого регистра на МАН. В регистры ТРА3 и ТРА4 объединены системные триггеры. О некоторых из них: ТИПУ, ТФАП, ТЗ ПКЗ упоминалось в п. 3.4. Состояние триггера интервалов времени (ТИБ) анализируется по микроприказу АН3 (1) в микропрограмме обработки таймера (п. 4.3). Триггер команды ввода-вывода (ТВ/В) анализируется по микроприказу АН4 (2) при выполнении команды ввода-вывода процессором (п. 5.3). Триггер блокировки (ТБ) анализируется по микроприказу АН4 (1) при обработке процессором прерываний от ввода-вывода.

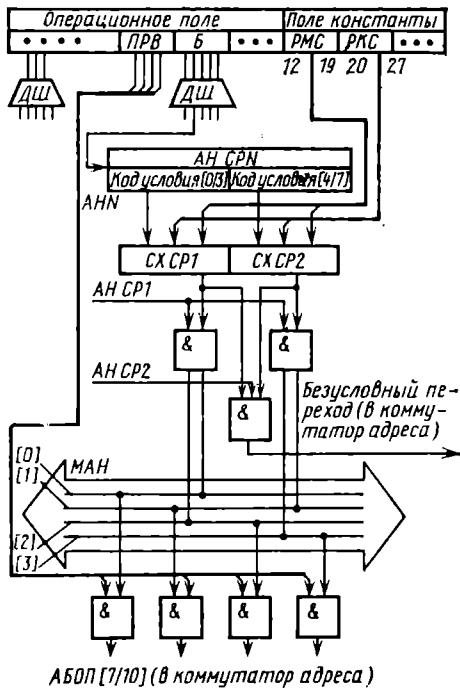


Рис. 62. Выполнение анализов сравнением

выдается на МАН, для немедленного ветвления не используется, а запоминается в ТРА1 или ТРА2 по микроприказу ТРА1 := МАН (ТРА2 := МАН). В нужный момент времени запомненный результат читается на МАН микроприказом АН1 (X) или АН2 (X), где X = ПРВ, в результате чего происходит соответствующее ветвление микропрограммы (п. 6.2).

В процессоре применяется еще один эффективный метод выполнения проверок — анализы путем сравнения с эталоном с применением маски [а. с. 615538 (СССР)]. Сравнению подвергаются 8-разрядные группы информации, поступающие из различных источников (рис. 62). Эталон и маска задаются микропрограммно полями РКС и РМС в такте выполнения анализа. Регистр микрокоманд показан условно в виде операционного поля и поля константы. В операционном поле выделены поля ПРВ и Б; последнее содержит коды анализов. В поле константы выделены поля РКС и РМС.

По одному из микроприказов выполнения анализа сравнением (АН8—АН15, табл. 19) содержимое двух проверяемых 4-разрядных групп в качестве кодов условия поступает в схемы сравнения СХ СР1 и СХ СР2. В схеме сравнения осуществляется по-разрядное сравнение кода условия с эталоном, заданным полем РКС. Маска сравнения, заданная полем РМС, разрешает сравнение в тех разрядах, где она содержит единицы. В тех разрядах, где в маске нули, сравнение считается выполненным независимо от значения проверяемого кода и эталона. Имеются два вида анализов путем сравнения. Первый из них — анализ сравнением первый — осуществляет модификацию базового адреса по результату анализа. Если все открытые маской разряды сравнились с эталоном, то по микроприказу АН СР1 схема сравнения СХ СР1 выдает единицы в разряды МАН [0, 2], а схема сравнения СХ СР2 — единицы в разряды МАН [1, 3]. Если сравнения не происходит, то соответствующие разряды МАН принимают нулевое значение. Таким образом, в результате анализа первого вида возможно формирование четырех комбинаций последней тетрады адреса БОП: 1010 — когда произошло сравнение только разрядов [0/3] кода условия с разрядами [0/3] кода сравнения; 0101 — когда произошло сравнение только разрядов [4/7] кода условия с разрядами [4/7] кода сравнения и 1111 либо 0000 — когда сравнение обеих тетрад этих кодов произошло либо не произошло. Разряды, на которые наложена нулевая маска, всегда считаются сравнившимися.

Т а б л и ц а 19

Анализы сравнением

| Номер анализа | Сравниваемая информация по разрядам кода условия | | | | | | | | Шестнадцатиричный код поля Б |
|---------------|--|------------|---|---|---------------------|------|------|------|------------------------------|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| 8 | РДШК[0/7] | | | | | | | | 08 |
| 9 | | | | | | | | | 09 |
| 10 | М3[0/3] | | | | М2[0/3] | | | | 0A |
| 11 | М3[28/31] | | | | М2[28/31] | | | | 0B |
| 12 | СТ1РОН | | | | СТ2РОН ₂ | | | | 0C |
| 13 | СТЦ1 | | | | СТЦ2 | | | | 0D |
| 14 | М3[0] | М2[1/31]=0 | | | М1[0] | М1≠0 | М2≠0 | М3≠0 | 0E |
| 15 | РСБФ | | | | | | | | 0F |

Пример. Выяснить, не является ли код операции в РДШК кодом 1С. Используется анализ 8 вида АН СР1. Код сравнения должен быть 1С, маска сравнения может быть FF или 1С. Для кодов в РДШК, равных 1С, и допустим, 4С (соответственно случаи *a* и *b*), формирование кода на МАН происходит по следующей схеме:

| | | |
|------------------------------------|---|---|
| код сравнения (поле <i>PKC</i>) | а) 0001 1100, | б) 0001 1100, |
| маска сравнения (поле <i>PMC</i>) | 1111 1111, | 1111 1111, |
| код в РДШК (код условия) | 0001 1100, | 0100 1100, |
| МАН[0/3] | <hr style="width: 100%; border: 0.5px solid black;"/> 1111; | <hr style="width: 100%; border: 0.5px solid black;"/> 0101. |

В случае *a* в результате сравнения обеих тетрад на МАН[0/3] формируется код 1111; в случае *b* в результате сравнения только правой тетрады на МАН[0/3] формируется код 0101. Один из этих кодов модифицирует младшую тетраду базового адреса БОП, позволяя разветвиться на два направления: РДШК[0/7] = 1С и РДШК[0/7] ≠ 1С.

Пример. На РДШК может находиться один из кодов операции: 1С, 4С, 5С; требуется выяснить наличие кода 4С. Используется анализ 8 вида АН СР1. Так как сравнению подлежит только первая тетрада кода операции, то код сравнения можно выбрать равным 0100, а маска сравнения F0, так как единичное значение маски разрешает сравнение соответствующих разрядов кодов условий и сравнения, а нулевое всегда означает их сравнение при любых значениях этих кодов. Для кодов в РДШК, равных 4С и, допустим, 5С (соответственно случаи *a* и *b*), формирование кода на МАН происходит по схеме:

| | | |
|------------------------------------|---|---|
| код сравнения (поле <i>PKC</i>) | а) 0100 0000, | б) 0100 0000, |
| маска сравнения (поле <i>PMC</i>) | 1111 0000, | 1111 0000, |
| код в РДШК (код условия) | 0100 1100, | 0101 1100, |
| МАН[0/3] | <hr style="width: 100%; border: 0.5px solid black;"/> 1111; | <hr style="width: 100%; border: 0.5px solid black;"/> 0101. |

В случае *a* произошло сравнение обеих тетрад, в случае *b* правые тетрады считаются сравнившимися. Один из кодов МАН модифицирует младшую тетраду базового адреса БОП.

Второй вид анализа путем сравнения — анализ сравнением второй — осуществляет по результату анализа условный переход по адресу возврата. В регистр РАВ из поля констант заранее записывается желаемый адрес. При сравнении кода условия с эталоном схемы сравнения СХ СР1 и СХ СР2 формируют сигнал, который по микроприказу АН СР2 поступает в коммутатор адреса. Последний выдает в БОП адрес возврата в качестве адреса следующей микрокоманды. При несравнении в качестве адреса следующей микрокоманды принимается базовый адрес.

Применение анализов АН12 (АН СР1) и АН14 (АН СР2) рассматривается в п. 5.1 АН10 (АН СР2) — в п. 6.1, АН14 и АН15 — в п. 8.1.

Метод ветвления по результату сравнения широко использован в реализованных алгоритмах. Он удобен для микропрограммирования, так как дает возможность выполнять несколько близких по типу операций по единой микропрограмме. Например, команда умножения полуслов выполняется по микропрограмме умножения полноразрядных слов. Процесс умножения останавливается по достижении счетчиком циклов определенного состояния (п. 6.2). Операции сравнения (где отсутствует запись результата) выполняются по микропрограмме вычитания, ход которой прерывается перед записью результата. Для этого в РАВ заранее заносится

адрес начала выборки, а код операции, находящийся в РДШК, подвергается сравнению с эталоном (кодом операции сравнения).

При выполнении анализа путем сравнения в качестве кода условия могут быть приняты код операции в РДШК, старшие и младшие тетрады слов, находящихся на магистралях М2 и М3, содержимое счетчиков СТ РОН и СТЦ, а также набор специальных условий (анализ 14). Особое место занимает регистр РСБФ (анализ 15). В нем сравнению может быть подвержена любая ранее поступившая информация, в том числе и принятая с МАН. Это означает, что сравниваться может результат обычного анализа или даже двух анализов, так как четырехразрядный код с МАН может быть независимо в разных тактах записан в правую или левую половины восьмиразрядного регистра РСБФ либо в обе половины одновременно. Последнее обстоятельство дает возможность осуществлять разветвление по трем направлениям.

Применение маски дает возможность проверять не весь байт, а только его часть. Таким образом, например, можно выделять класс команд, проверять знаки результатов и т. п. Один блок сравнения на 8 разрядов обеспечивает $256 \times 256 = 64\,386$ видов проверок путем задания любых из возможных кодов и масок. Этот метод не ограничен какой-либо системой команд или конкретными алгоритмами микропрограмм. Он может быть рекомендован для широкого использования в системах, для которых микропрограммы разрабатываются самими пользователями. Его применение позволяет также значительно экономить ячейки памяти, так как в данном случае всегда имеет место ветвление только на 2 направления: либо по найденному искомому коду, либо по всем остальным кодовым комбинациям, в результате чего занимают всего 2 ячейки памяти. Применение других методов, например, метода функционального перехода, используемого в моделях IBM/360, ведет к большому расходу ячеек, так как в этом случае в одном из них кодируется реакция на искомый код, а во всех других — одно и то же противоположное действие. Пусть, например, необходимо выполнить одно действие при состоянии счетчика 0101 и другое при всех прочих его состояниях. Если ветвление производится путем модификации адреса присвоением значения счетчика некоторым четырем разрядам кода адреса БОП, то занимается 16 ячеек, из которых 15 служат для организации выполнения одного и того же действия, и только ячейка с адресом Х.Х.0101 предназначена для запланированного.

Схемы микропрограммного управления размещены в стойке процессора следующим образом.

Блок режимов работы: реализован на ТЭЗ дешифратор ДШ—9 шт.; дешифратор полей ДШП — 2 шт. и частично в ТЭЗ пульт управления ПУ1, ПУ2, ПУ3 и ПУ4;

Блок формирования адреса БОП: коммутатор адреса БОП реализован на четырех ТЭЗ адрес БОП АБОП; узел анализов и ветвлений на ТЭЗ анализ 1 АН1 — 4 шт., анализ 2 АН2 —

1 шт., анализ деления АНД — 4 шт., анализ циклов АНЦ — 1 шт., анализ неуспешной передачи управления АН НПУ — 1 шт., регистр анализов РАН — 4 шт., а также часть узлов в ТЭЗ регистр сравнения буферный РСБФ, регистр прерываний РПР регистр дешифрации команды РДШК.

3.6. ПУЛЬТ УПРАВЛЕНИЯ

Пульт управления (рис. 63) предназначен для управления работой машины и индикации ее состояния в различных режимах функционирования. С пульта управления производятся включение и отключение питания, сброс системы, занесение информации в ОП и регистры и ее индикация, загрузка начальной программы и другие операции. Для этого он содержит необходимые переключатели и элементы световой индикации. В ЭВМ ЕС, благодаря развитой структуре устройства управления центрального процессора, каналов ввода-вывода и внешних устройств, необходимость в ручном управлении машиной сведена к минимуму.

Пульт управления ЭВМ ЕС-1033 крепится к торцовой части стойки процессор — каналы. Конструктивно он состоит из трех панелей: верхней, средней и нижней. Эти три панели содержат органы управления и индикации по сути дела трех пультов, предусмотренных «Принципами работы ЕС ЭВМ»: пульта оператора, пульта ручной работы и инженерного пульта. Рассмотрим назначение основных переключателей и элементов индикации.

Пульт оператора — это часть ПУ машины, которая содержит средства, необходимые оператору в случае, когда управление работой обеспечивается супервизором. С помощью этого пульта производятся включение и отключение питания, загрузка начальной программы, вызов внешнего прерывания, индикация состояния системы, а также индикация системы питания.

В верхней части верхней панели расположена группа лампочек К, П, ОП1, ОП2 зеленого цвета с надписью «ДИСТАНЦИОННО». Они загораются тогда, когда на систему полагается питание и переключатели «МЕСТНО—ДИСТАНЦИОННО» на блоках управления системой электропитания соответственно процессора, ОП1, ОП2 и каналов ввода-вывода находятся в положении «ДИСТАНЦИОННО». В этом случае система может быть включена с ПУ.

Нажатие кнопки «ПИТ. ВКЛ.», расположенной в верхней правой части ПУ, вызывает последовательность действий по включению питания в системе. Блоком управления системы питания производится последовательное включение питания каналов, процессора, ОП1 и ОП2; затем под управлением блока включения внешних устройств последовательно включаются внешние устройства селекторного канала 1, селекторного канала 2, селекторного канала 3, мультиплексного канала. При нажатии кнопки «ПИТ. ВКЛ.» в ШПУ процессора аппаратурно вырабатывается

сигнал СБРОС, поступающий в ЦП, блок диагностики и каналы ввода-вывода. Одновременно ШПУ формирует фиксированный адрес БОП 0. А. 7 микропрограммы сброса, которая производит восстановление четности всех регистров процессора и ОП (п. 4.2). После завершения последовательности действий по включению питания процессор переходит в состояние СТОП (TRC устанавливается в единичное состояние). Загорается лампочка внутренней подсветки кнопки «ПИТ. ВКЛ.», а также группа лампочек К, П, ОП1, ОП2 зеленого цвета с общей надписью «ПИТАНИЕ ВКЛЮЧЕНО». Это говорит о том, что произошло включение питания соответствующих устройств.

Нажатие кнопки «ПИТ. ОТКЛ.» вызывает последовательность действий по выключению питания в системе. После отключения питания загорается внутренняя подсветка кнопки.

Кнопка «АВАР. ОТКЛ.» предназначена для экстренного отключения питания одновременно всех устройств системы блоком управления системы питания. Об аварии питания в отдельном устройстве свидетельствует загорание одной из красных лампочек в группе К, П, ОП1, ОП2 с общей надписью «АВАРИЯ».

Загорание лампочки «ПРОФ. КОНТР.» свидетельствует о том, что на одно из устройств К, П, ОП1, ОП2 в целях профилактического контроля подано повышенное или пониженное напряжение питания.

Клавиши «АДРЕС КАНАЛА», «АДРЕС УСТРОЙСТВА», расположенные в правой нижней части ПУ, позволяют оператору задать тип и номер канала, а также адрес устройства ввода-вывода, которое будет использовано для загрузки начальной программы. Положения двух клавиш «АДРЕС КАНАЛА» означают:

- 00 — мультиплексный канал,
- 01 — селекторный канал 1,
- 10 — селекторный канал 2,
- 11 — селекторный канал 3.

С помощью восьми клавиш «АДРЕС УСТРОЙСТВА» может быть набран любой адрес в пределах от 0 до 255.

Кнопка «ЗАГРУЗКА» предназначена для того, чтобы начать загрузку начальной программы после выбора канала и устройства ввода-вывода. При ее нажатии в ШПУ аппаратурно вырабатывается сигнал СБРОС и формируется фиксированный адрес БОП 0. А. 5 микропрограммы начальной загрузки. В начале выполнения производится восстановление четности регистров РИП, РАП, РОН, РПЗ; а затем в РИП [24/31] заносятся номера канала и устройства, с которого производилась загрузка (п. 4.2). При наличии питания в системе кнопка постоянно подсвечивается.

Лампочка «ЗАГР.» включается во время ввода начальной программы. Она индицирует состояние триггера загрузки (ТЗ) пульта управления, который устанавливается в единичное состояние при

нажатию кнопки «ЗАГРУЗКА», а сбрасывается сигналом ТЗ := 0, приходящим из канала, или сигналом СБРОС.

Кнопка «ПРЕРЫВ.» служит для вызова внешнего прерывания. При нажатии этой кнопки устанавливается в единичное состояние РФП [3]. Если процессор находится в состоянии РАБОТА (лампочка «РУЧ.» не горит), прерывание обрабатывается в том случае, если оно не замаскировано. В противном случае запрос на прерывание хранится в ожидании обработки. При обработке прерывания устанавливается в единичное состояние РССП [25], указывая, что источником внешнего прерывания является кнопка на пульте управления.

Лампочка «СИСТ.» включена, если: процессор находится в состоянии СЧЕТ (РССП [14] = 0), а состояние устройств ввода-вывода не определено либо работает одно из устройств ввода-вывода, а состояние процессора безразлично (ОЖИДАНИЕ, СТОП).

Лампочка ручного управления «РУЧ.» включена, если процессор находится в состоянии СТОП. При этом индицируется нулевое состояние триггера рабочего состояния (ТРС=0). Кнопки пульта ручной работы — «ЗАПИСЬ», «ЧТЕНИЕ», «ПОВТ. РССП», «УСТ. РССП [40/63]», «ПУСК» и кнопки инженерного пульта — «СБР. М», «ВЫЗ. МК» действуют только тогда, когда процессор находится в состоянии СТОП (лампочка «РУЧ.» горит). При этом также горят лампочки внутренней подсветки перечисленных кнопок. Из них кнопка «ПУСК» подсвечивается постоянно при наличии питания в системе.

Лампочка «ОЖИД.» включена, если процессор находится в состоянии ОЖИДАНИЕ (РССП [14] = 1).

Пульт ручной работы содержит средства управления, необходимые для того, чтобы оператор мог вмешиваться в нормальное выполнение программы, а также для того, чтобы обеспечить оператору работу с машиной при техническом обслуживании, наладке программ, анализе неисправностей и т. д.

В правой нижней части нижней панели расположена кнопка сброса системы «СБРОС СИСТ.». При нажатии на эту кнопку аппаратно вырабатывается сигнал СБРОС, который поступает в каналы ввода-вывода, блок диагностики и узлы процессора. Одновременно ШПУ формирует фиксированный адрес БОП 0.А.6 микропрограммы сброса системы (п. 4.2). Сигнал сброса устанавливает в исходное состояние аппаратуру процессора и каналов, а микропрограмма сброса системы восстанавливает четность ОП, РОН, РПЗ, РОП, ПКЗ, РССП, РИП, РАП и регистров блока прямого управления. Кнопка подсвечивается постоянно при наличии питания в системе. На состояние автономных или разделенных устройств УВУ сброс системы не влияет.

Рядом с кнопкой «СБРОС СИСТ.» расположена кнопка «ОСТАНОВ». При нажатии на эту кнопку процессор останавливается после завершения команды.

Клавиши «АВТ.», «ШАГ», «ТАКТ» в линейке клавиш под общим названием «РЕЖИМ РАБОТЫ» и расположенные над кнопками «СБРОС СИСТ.» и «ОСТАНОВ», указывают, каким образом происходит выполнение команд. Если нажаты клавиша «АВТ.» (автоматический режим) и кнопка «ПУСК» (расположена рядом с кнопкой «ОСТАНОВ»), то выполняется обычная последовательность команд по программе. При этом лампочка «РУЧ.» не горит. Если нажаты клавиша «ШАГ» и кнопка «ПУСК», то будет выполнена одна команда и последовательно обработаны все незамаскированные прерывания, ожидающие обработки. Затем произойдет останов процессора. Если нажаты клавиша «ТАКТ» и кнопка «ПУСК», будет выполнена одна микрокоманда.

При нажатии кнопки «ПУСК» в соответствии с положением клавиш «РЕЖИМ РАБОТЫ» начинается выполнение операции. Если кнопка пуска нажимается после нормального останова и нажата клавиша «АВТ.» или «ШАГ», то выполнение последовательности команд возобновляется так, если бы этого останова не было. При нажатии кнопки «ПУСК» после сброса системы без предварительного введения нового адреса команды в РССП [40/63] выполнение программы начнется с команды, адрес которой находился в РССП [40/63] в момент останова.

При нажатии кнопки «УСТ. РССП [40/63]» (расположена рядом с кнопкой «СБРОС СИСТ.»), адрес, набранный на РАПУ [0/23] (см. ниже), засылается в поле адреса команды (разряды [40/63]) текущего ССП. При этом происходит пересылка $РАП := РАПУ$ с присвоением контрольных разрядов, а ШПУ формирует фиксированный адрес БОП 0. А. 1 микропрограммы УСТАНОВИТЬ РССП [40/63].

При нажатии кнопки «ПОВТ. РССП», расположенной в ряду кнопок в средней части нижней панели, ШПУ формирует фиксированный адрес БОП 0. А. 2 и микропрограммно из нулевой ячейки ОП считывается двойное слово и пересылается в РССП. Таким образом ССП начальной программы становится текущим и фактически программа повторяется. В режиме «Шаг» происходит останов по началу выборки команды.

В верхней части нижней панели под надписью «АДРЕС» находятся 24-разрядный клавишный регистр адреса ПУ (РА ПУ) и соответствующие лампочки индикации. РА ПУ используется для задания адресов различных видов памяти, а также для занесения кода адреса в РССП [40/63]. Нажатое состояние клавиши записывает единицу в одноименный разряд, отжатое — ноль.

Адреса памяти разных типов на РА ПУ распределены следующим образом:

РАПУ [0/3] — адрес РОН, РПЗ, РОП;

РАПУ [0/12] — адрес ПКЗ;

РАПУ [13/23] — адрес БОП, БОП6;

РАПУ [0/23] — адрес ОП и РССП [40/63].

Клавиши, объединенные общим названием «ВЫБОР ПАМЯТИ» («ОП», «РОН», «РПЗ», «ПКЗ», «РОП», расположенные ниже РА ПУ), служат для задания вида памяти, к которому относится адрес, набранный на РА ПУ.

При нажатии кнопки «ЧТЕНИЕ» в ряду кнопок, расположенных в средней части нижней панели, информация из ячейки, определяемой клавишами «ВЫБОР ПАМЯТИ» и клавишами РА ПУ, индицируется на пульте. Индикация производится на пяти линейках индикации — четырех 36-разрядных информационных линейках средней панели и одной 27-разрядной адресной линейке нижней панели. У каждой восьмиразрядной группы индицируется значение ее контрольного разряда. Выдача на индикацию состояния регистров производится согласно табл. 20.

Т а б л и ц а 20

Таблица «ИНДИКАЦИЯ» пульта управления

| Линейка индикации | Клавиши | | | |
|-------------------|------------|-------|-------------------|-------|
| | «ИНФ» | «РМК» | «РССП» | «РМО» |
| М1 | РАКБ | РМК0 | РССП | РМО |
| М2 | РФПД, РСБФ | РМК1 | РССП | РМО |
| М3 | ТРА | РМК2 | РДШК, РХК, РПС | РМО |
| 4-я линейка | РИП | РМК3 | | РМК6 |
| АДРЕС | РАП | — | РССП[40/63] | — |

Когда все клавиши под общим названием «ИНДИКАЦИЯ» (в правой нижней части средней панели) отжаты, на пульте индицируется состояние магистралей М1, М2, М3, а на четвертой линейке — состояние триггеров: занятости РХК (ТЗ РХК), запроса выборки ТЗВ, плавающей запятой ТПЗ, приема в РМК (ТП РМК), сброса магистралей ТСБР. М, тактовых импульсов И2, И3, И4 тактовой цепочки БУ ОП, по которым происходит обращение к ОП, значение переноса АЛУ (ПАЛУ) регистра формирования маски (РФМ), а также состояние счетчиков СТ1 РОН, СТ2 РОН, СТЦ1, СТЦ2. На линейке индикации под общим названием «АДРЕС» (в нижней панели) при отжатых клавишах «ИНДИКАЦИЯ» постоянно отображается состояние буферного регистра адреса БОП (РА БОПБ).

Для индикации регистров, указанных в таблице, необходимо нажать на соответствующую клавишу «ИНДИКАЦИЯ». Состояние регистров согласно таблице будет индицироваться на линейках индикации (М1, М2, М3, четвертая линейка индикации,

АДРЕС). Индикация регистров при нажатии клавиши «ИНФ.» или «РМО» (красного цвета) возможна только при нахождении процессора в состоянии СТОП, когда горит лампочка «РУЧ.». При нажатии клавиши «РМК» или «РССП» (белого цвета) индикация регистров возможна и в автоматическом режиме.

Для индикации содержимого ячеек ОП необходимо после нажатия кнопки «ЧТЕНИЕ» в ряду «ИНДИКАЦИЯ» нажать клавишу «ИНФ.» Информация из ОП (содержимое РИП) будет индцироваться на четвертой линейке индикации, а адрес ОП (содержимое РАП) — на линейке «АДРЕС». При индикации содержимого остальных видов памяти все клавиши «ИНДИКАЦИЯ» должны быть отжаты.

Индикация содержимого РОН происходит микропрограммно. При нажатии кнопки «ЧТЕНИЕ» ШПУ формирует фиксированный адрес БОП 0. А. 4 пультовой микропрограммы, которая производит чтение содержимого выбранного регистра на линейку индикации. Причем на М2 индицируется РОН, адрес которого набран на РА ПУ [0/3], а на М3—РОН, адрес которого на единицу больше.

Индикация РПЗ происходит аналогично. На М2 индицируется первое слово выбранного РПЗ, на М3 — второе слово.

Содержимое РОП индицируется на линейке М2, причем перед нажатием кнопки «ЧТЕНИЕ» необходимо нажать кнопку «СБР.М».

В нижней части средней панели находится 32-разрядный клавишный регистр информации пульта управления (РИ ПУ) и соответствующие лампочки индикации. Эти клавиши служат для набора информации, которая затем записывается в ячейку, определяемую клавишами «ВЫБОР ПАМЯТИ» («ОП», «РОН», «РПЗ», «РОП») и адресом, набранным на РА ПУ. На РИПУ [24/28] набирается информация для записи в ПКЗ. При вызове в РМК микрокоманды из БОП адрес микрокоманды набирается на РИПУ [21/31].

При нажатии кнопки «ЗАПИСЬ» информация из РИ ПУ заносится в память. Вид памяти определяется клавишами «ВЫБОР ПАМЯТИ» и адресом, набранным на РА ПУ.

Часть ПУ, образующая пульт инженера, содержит средства управления, которые используются при инженерном обслуживании. Она условно разделена на две части: инженерный пульт процессора и инженерный пульт каналов. Последний рассматривается в п. 13.1.

Кнопка («ВЫЗ. МК.») служит для вызова в РМК микрокоманды, адрес которой набран на РИПУ [21/31]. При нажатии этой кнопки триггер вызова микрокоманд устанавливается в единичное состояние и стробирует на коммутаторе адреса прохождения кода из РИПУ [21/31] в качестве адреса БОП.

Клавиши «ЦИКЛ» и «АВТОН.» в линейке клавиш «РЕЖИМ РАБОТЫ» служат для задания специальных режимов: При нажатии клавиши «ЦИКЛ» действие кнопок «УСТ. РССП [40/63]»,

«ПОВТ. РССП», «ЗАПИСЬ», «ЧТЕНИЕ», «ЗАГРУЗКА», «СБРОС СИСТ.» (а также действие сигнала СБРОС по включению питания), «ПУСК», «ОСТ. МК.», «ПРЕРЫВ», «ВЫЗ. МК» зацикливается. Если нажата клавиша «АВТОН», то при чтении из ОП информация ОП подменяется информацией с РИ ПУ.

Кнопка сброса магистралей «СБР. М» в ряду кнопок, находящихся на нижней панели, служит для установки в единичное состояние всех разрядов магистралей М1, М2, М3.

С помощью инженерного пульта можно зациклить участок микропрограммы, задавая его начальный и конечный адреса. Для этого необходимо выполнить следующие действия:

на РИПУ [21/31] набрать начальный адрес микропрограммы; нажать на кнопку «ВЫЗ. МК.»; на инженерном пульте каналов нажать клавишу «ЦИКЛ»; на РАПУ [13/23] набрать конечный адрес микропрограммы; в ряду клавиш «РЕЖИМ РАБОТЫ» нажать клавишу «АВТ.»; нажать на кнопку «ПУСК».

Рядом с клавишами «ВЫБОР ПАМЯТИ» на нижней панели ПУ находятся объединенные общей надписью клавиши блокировки записи «БЛ. ЗП» в «ОП», «РОН», «РОП». Нажав одну из этих клавиш, при выполнении микропрограммы можно заблокировать запись в соответствующий нажатой клавише вид памяти.

При нажатии клавиши блокировки диагностики «БЛ. Д.» возникающие ошибки фиксируются в регистре машинных ошибок блока контроля процессора, но обработка их блоком диагностики не производится.

При нажатии клавиши «ОСТ. ПО ОШ» происходит останов процессора в случае возникновения ошибки.

Клавиша диагностики «ДИАГ.» коммутирует действие кнопок «ПУСК» и «ВЫЗ. МК». При нажатии клавиши «ДИАГ.» этими кнопками производятся вызов и выполнение в соответствующем режиме (АВТ., ТАКТ, ЦИКЛ) диагностических микрокоманд. Сами диагностические микрокоманды, извлекаемые из БОП6 в РМК6, индицируются на четвертой линейке при нажатой клавише «РМО». При отжатой клавише «ДИАГ.» этими кнопками производятся вызов и выполнение микрокоманд процессора.

С помощью клавиши «ИНСТР.», «ДАН», «БОП», «БОП6», объединенных надписью «ОСТАНОВ ПО АДРЕСУ», осуществляется останов процессора при совпадении текущих адресов команды, данных, БОП0—БОП3 или БОП6 с адресом, набранным на клавиатуре РАПУ [0/23]. Если нажата клавиша «ИНСТР.», то при совпадении текущего адреса очередной команды (инструкции) вырабатывается сигнал, который сбрасывает в нулевое состояние триггеры ТРС, ТП РМК и блокирует сброс магистралей. Текущая микрокоманда завершается, следующая микрокоманда выбирается в РМК, но не выполняется. После выполнения последней микрокоманды информационные магистрали не сбрасываются. Это дает возможность осуществления визуального контроля информации в момент останова.

Если нажата клавиша «ДАН», то при совпадении текущего адреса с адресом обращения в ОП за данными, произойдет останов аналогично останову по адресу команды. При этом в РССП [40/63] будет уже находиться адрес следующей команды.

Кнопка останова по микрокоманде «ОСТ. МК.» предназначена для останова ЭВМ, когда процессор не может закончить выполнение операции из-за машинного сбоя или имеется непрерывный поток прерываний.

Клавиша останова по адресу БОП6 используется в режиме «Диагностика» при отладке различных сегментов диагностических микропрограмм и при поиске места неисправности. При совпадении кода адреса текущей диагностической микрокоманды ДМК и кода адреса, набранного на РАПУ [13/23], происходит останов процессора. При возникновении ошибки в БОП6 загорается лампочка ошибки диагностического оборудования «ОШ. Д. ОБОР.» (в группе из трех лампочек в левом верхнем углу нижней панели).

Лампочка «ОТК. ОБОР.» загорается при появлении отказа в основном оборудовании ЦП, локализованном блоком диагностики в результате выполнения тестов локализации неисправностей (ТЛН).

ГЛАВА 4

СЛУЖЕБНЫЕ И ОБСЛУЖИВАЮЩИЕ МИКРОПРОГРАММЫ

Процессор начинает работать при нажатии оператором на ПУ любой из кнопок, возбуждающих ту или иную пультовую процедуру. Это могут быть кнопки «ПУСК», «ЗАГРУЗКА», «ПРЕ-РЫВ» и др. Управление действиями, обеспечивающими выполнение пультовых процедур, может осуществляться аппаратными или микропрограммными средствами. С начала работы процессора при выполнении команд исполняемой программы в нем всегда в строго определенной последовательности осуществляются:

проверка на наличие запросов разрешенных прерываний. Если имеется зафиксированный запрос прерывания, разрешенного к обработке, то происходит переход на микропрограмму обработки этого прерывания. После завершения этих действий вновь проверяется наличие запросов прерываний. Такие циклы повторных проверок будут продолжаться до тех пор, пока не будут обслужены заявки на все разрешенные прерывания (рис. 64);
выборка команды. Микропрограмма выборки одина для всех команд;

выполнение операции, определяемой кодом операции команды.

На этапах выборки и выполнения команды могут возникать ситуации, при которых оказывается нецелесообразно или невозможно продолжение исполняемой микропрограммы. В этих случаях осуществляется переход к микропрограмме обработки прерываний.

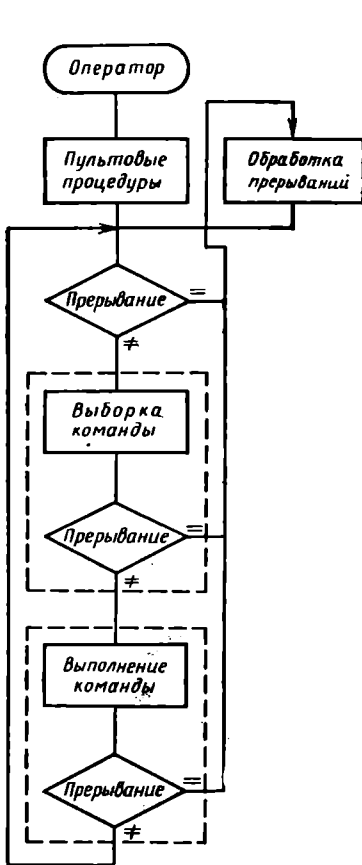


Рис. 64. Последовательность выполнения команд процессором

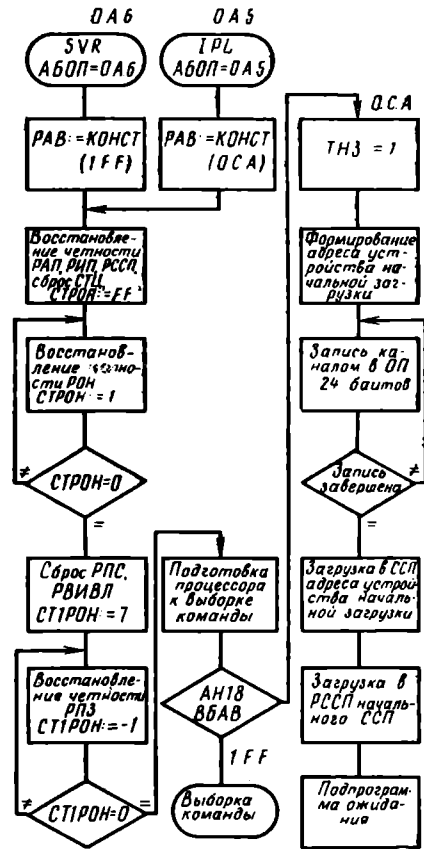


Рис. 65. Схема микропрограмм СБРОС СИСТЕМЫ (SVR) и НАЧАЛЬНАЯ ЗАГРУЗКА ПРОГРАММЫ (IPL)

Все микропрограммы центрального процессора условно разбиты на три группы: микропрограммы пультных процедур, микропрограммы обработки прерываний и продвижения таймера, микропрограммы выполнения команд.

4.1. МИКРОПРОГРАММЫ ПУЛЬТОВЫХ ПРОЦЕДУР

Ряд пультных процедур реализован аппаратурно — микропрограммным способом. К ним относятся: сброс системы; включение питания; начальная загрузка программы; загрузка начального ССП; установка РССП; запись в РОН, РПЗ; чтение РОН, РПЗ.

После нажатия той или иной кнопки управление входом в микропрограммы пультовых процедур осуществляется аппаратно путем формирования соответствующего адреса БОП. Адрес АБОП вырабатывается шифратором пульта управления, расположенным в блоке режимов работы (см. рис. 60).

СБРОС СИСТЕМЫ (SVR). ШПУ вырабатывает адрес БОП 0. А. 6 (рис. 65). В первой же микрокоманде происходит запись в регистр адреса возврата РАВ константы 1.F.F, которая в конце микропрограммы используется как адрес БОП для перехода на микропрограмму выборки команд (ВХОД 1, п. 4.4). Далее, по сигналу СБРОС, формируемому БРР, в исходное состояние устанавливается ряд триггеров и регистров, в том числе регистр машинных ошибок. Микропрограммно восстанавливается, в случае нарушения, четность регистров РОН, РПЗ, РССП, РИП, РАП и обнуляются регистры РХК, РВИВЛ, РПС. При сбросе системы запись в РМО блокируется, что позволяет восстановить четность регистров, используя магистральное оборудование и БАЛ.

В конце микропрограммы имеется микрокоманда, содержащая микроприказы АН18 и ВБАВ (п. 3.5). Их сочетание вызывает безусловный переход по адресу возврата, который находится в РАВ. В данном случае происходит переход на микрокоманду с адресом 1.F.F, которая является начальной микрокомандой микропрограммы выборки команд.

ВКЛЮЧЕНИЕ ПИТАНИЯ (SPS). Микропрограммная часть процедуры ВКЛЮЧЕНИЕ ПИТАНИЯ осуществляет сброс системы, который происходит аналогично рассмотренной выше процедуре. Отличие заключается лишь в том, что восстановление четности РОН, РПЗ, РССП производится путем установки в нуль указанных регистров и дополнительно восстанавливается четность РОП, ПКЗ и ОП.

НАЧАЛЬНАЯ ЗАГРУЗКА ПРОГРАММЫ (IRL). Она производится в случаях, когда состояние ЦП не позволяет системе начать работу самостоятельно. При нажатии кнопки «ЗАГРУЗКА» на пульте управления канал считывает с устройства начальной загрузки, номер которого набран на линейке «АДРЕС УСТРОЙСТВА» пульта управления (п. 3.6), 24 байта и записывает их в оперативную память. Информация, записанная в ОП, определяет начальное ССП и два управляющих слова канала, которые обеспечивают последующий ввод информации и работу системы.

После нажатия кнопки ШПУ формирует адрес БОП 0.А.5. В первой же микрокоманде в РАВ заносится адрес перехода 0.С.А, после чего исполняется микропрограмма сброса. В ее конце по микроприказам АН18 и ВБАВ управление передается микрокоманде с адресом 0.С.А, с которой начинается собственно микропрограмма начальной загрузки. По этой микрокоманде устанавливается в единичное состояние триггер начальной загрузки (ТНЗ = 1). Сигнал с единичного выхода этого триггера поступает в канал. При получении этого сигнала канал формирует команду

ЧИТАТЬ для устройства, определяемого адресами канала и устройства, набранными на клавиатуре пульта управления. На время начальной загрузки с заданного устройства работа ЦП зацикливается на одной микрокоманде в ожидании завершения записи каналом в ОП первых 24 байт. Из них первые восемь байт представляют собой начальное слово состояния программы. В зону кода прерывания ССП заносятся номер канала и номер устройства начальной загрузки. Сформированное таким образом начальное ССП записывается в РССП, определяя дальнейшую работу системы. Процедура завершается подпрограммой ожидания.

ЗАГРУЗКА НАЧАЛЬНОГО ССП (RPSW). Процедура установки начального ССП заключается в передаче в РССП в качестве нового ССП двойного слова из нулевой ячейки оперативной памяти.

УСТАНОВКА РССП (SPSW). Процедура установки РССП заключается в занесении информации, набранной на РА ПУ, в зону адреса команды РССП [40/63]. Передача кода адреса происходит после нажатия соответствующей кнопки. Эта процедура позволяет оператору вручную изменить последовательность выполнения команд.

ЗАПИСЬ В РОН, РПЗ (LGPR). Процедура записи в РОН, РПЗ заключается в занесении информации, набранной на клавиатуре РИ ПУ, в регистр РОН или РПЗ, номер которого определяется разрядами [0/3] клавиатуры РА ПУ. Запись осуществляется при нажатии кнопки **ЗАПИСЬ** на пульте управления. Конкретный регистр (РОН или РПЗ), в который записывается информация, определяется соответствующим переключателем ПУ.

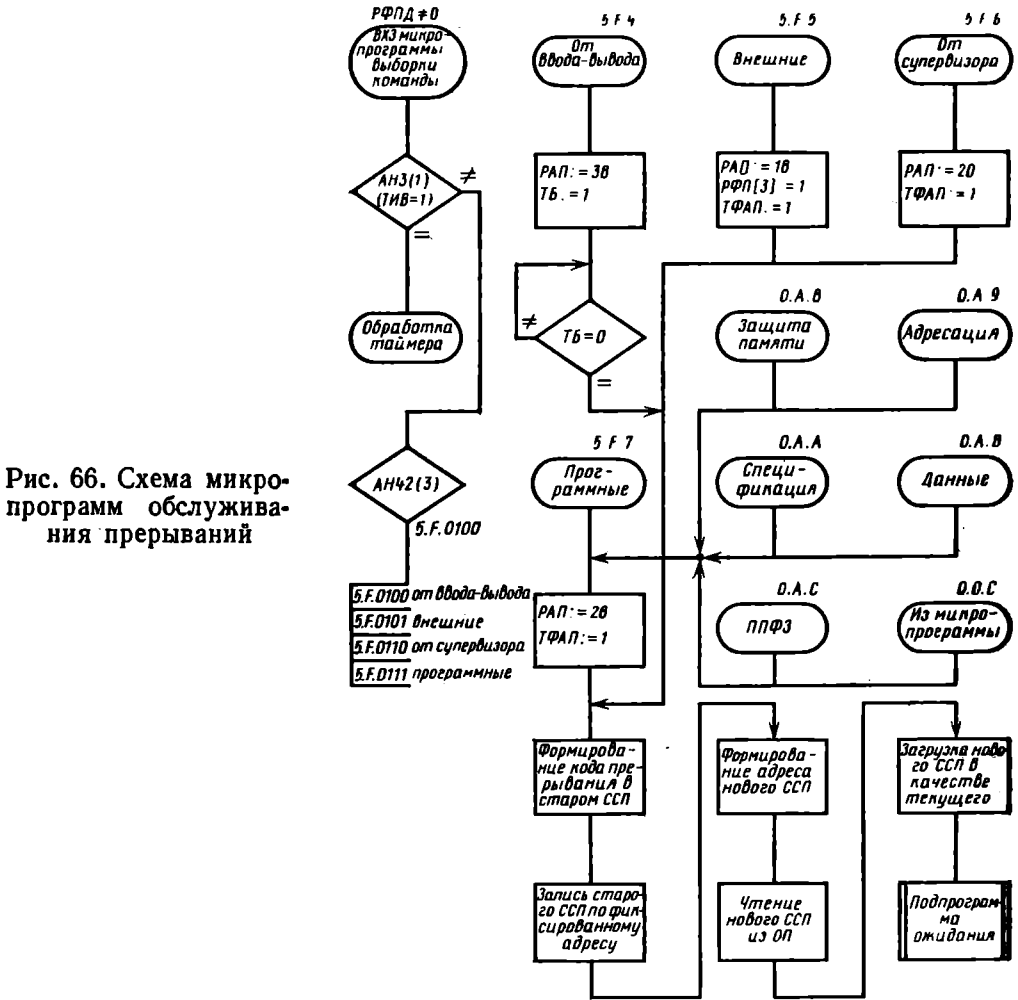
ЧТЕНИЕ РОН, РПЗ (RGPR). Эта процедура используется для индикации содержимого РОН или РПЗ на линейках М2 и М3 ПУ. При этом индицируется два последовательно расположенных РОН или РПЗ. Адрес старшего регистра РОН (РПЗ) определяется клавиатурой ПУ, а адрес следующего формируется микропрограммно.

4.2. ОБСЛУЖИВАНИЕ ПРЕРЫВАНИЙ, КОРРЕКТИРОВКА ТАЙМЕРА

Система прерываний позволяет процессору изменять свое состояние при возникновении определенных условий как внутри ЦП, так и вне его (п. 3.3). Прерывания происходят только в том случае, если они разрешены, т. е. не замаскированы (прил. 7), и принимаются к обработке после завершения выполнения текущей команды, но до начала выполнения следующей. Однако при возникновении некоторых программных прерываний и прерываниях от схем контроля становится нецелесообразным или даже невозможным продолжение выполнения текущей команды. В этих случаях осуществляется переход на обработку прерываний до ее завершения.

После выполнения какой-либо команды в начале выборки следующей проводится анализ 38 (п. 4.4). Составной частью этого анализа является проверка состояния разрядов регистра фиксации прерываний РФПД (п. 3.2). При РФПД $\neq 0$ происходит переход на подпрограмму обработки прерываний (вх. 3, п. 4.4).

Вначале по микроприказу АНЗ (1) (см. табл. 18, п. 3.5) анализируется состояние триггера интервалов времени (ТИВ). Если



ТИВ = 1, то происходит переход на обработку таймера (рис. 66). Если же ТИВ = 0, то в данный момент обработки таймера не требуется, и анализируется состояние следующих разрядов РФПД для определения класса прерываний, запросы которых в нем зафиксированы.

Класс прерываний определяется по микроприказу АНЗ2 (3). Кодированная схема выдает на МАН код класса прерывания, который модифицирует базовый адрес 5.F.0100 (см. табл. 15) следующей микрокоманды. Модифицированный адрес определяет начальный адрес микропрограммы обработки соответствующего пре-

рывания. При входе в микропрограмму обслуживания прерывания любого класса в начале всегда выполняются действия по формированию фиксированного адреса старого ССП (прил. 10), который засылается в регистр адреса процессора РАП, а также запоминается для дальнейшего использования во втором операционном регистре РОП2. Установленный в единичное состояние триггер ТФАП блокирует ошибки, которые могут возникнуть при обращении к ОП в момент записи старых ССП. При обслуживании прерываний от ввода-вывода блокировка ошибок обращения к ОП произойдет триггером ТФАК, входящего в состав каналов ввода-вывода. В этом случае для сообщения каналу о начале обслуживания прерывания устанавливается в «1» триггер блокировки ТБ. Он блокирует продвижение микропрограммы ЦП на промежуток времени, в течение которого канал записывает в ОП слово состояния канала. По окончании записи ССК канал передает в ЦП код прерывания и сбрасывает триггер блокировки, разрешая тем самым дальнейшее продвижение микропрограммы. Центральный процессор записывает полученный из канала код прерывания в разряды [16/31] текущего ССП, после чего происходит выход на общую для всех классов прерываний часть микропрограммы. По ней осуществляется запись текущего ССП в качестве старого, формирование адресов новых ССП, чтение их из ОП и запись в РССП в качестве текущего слова состояния. При обслуживании прерываний от внешнего источника РФП [3] сбрасывается по отдельному микроприказу РФП [3] := 0. Это объясняется тем, что запросы внешних прерываний аппаратно не гасятся (п. 3.2). Микропрограмма обработки прерывания завершается подпрограммой ожидания.

Если в ходе выполнения команды возникают ошибки защиты памяти, адресации, спецификации (п. 3.4), данных и переполнение в операциях с фиксированной запятой (ППФЗ) (п. 2.4), то аппаратно вырабатывается сигнал, по которому шифратор пульта управления формирует один из фиксированных адресов 0.А.8, 0.А.9, 0.А.А, 0.А.В или 0.А.С соответственно. При этом возникает немедленное прерывание программы (п. 3.2), а названные адреса являются начальными адресами соответствующей подпрограммы обработки прерываний.

В ходе выполнения микропрограммы команды возможно возникновение условия программного прерывания. В этом случае микропрограмма формирует адрес БОП 0.0.С, который обеспечивает вход в микропрограмму обработки прерывания. Например, при выполнении операции в команде ЗАГРУЗКА ССП (п. 5.2) в ее начале анализируется состояние процессора ЗАДАЧА/СУПЕРВИЗОР), поскольку эта команда является привилегированной и может выполняться только в состоянии СУПЕРВИЗОР. Если процессор находится в состоянии ЗАДАЧА, то следует переход на микрокоманду 1.6.А, по которой устанавливается в единичное состояние РФП [1] (РФП [1] := 1), в РОП10 зано-

сится код 2 ($M1 := 0000\ 0002$, $POPI0 := M1$), означающий прерывание по привилегированной операции (прил. 8), и происходит переход по адресу 0.0.C на подпрограмму обработки прерывания.

Таймер служит для измерения интервалов времени. Код текущего времени представляет полное слово, хранящееся в ОП по фиксированному адресу 50₁₆. Слово таймера по указанному адресу записывается операционной системой. Во время работы процессора периодически, с частотой 50 Гц, происходит корректировка (обработка) содержимого слова таймера по специальной микропрограмме. Это происходит в промежутках между выполнением команд. Таймерный триггер интервалов времени устанавливается в состояние «1» с частотой 50 Гц специальным генератором. Перед началом очередной команды, когда анализируется РФПД для проверки наличия необработанных запросов прерываний (см. выше), единичное состояние этого триггера вызывает переход на микропрограмму обработки таймера (рис. 67). В начале микропрограммы происходит сброс $TIB := 0$, формируется и записывается в РАП адрес слова таймера, устанавливается в единичное состояние ТФАП. Затем слово таймера выбирается из ОП. Оно рассматривается как целое число со знаком, разряды [24/31] которого не используются. По правилам операций с фиксированной запятой производится вычитание из этого числа единиц в 21-м и 22-м разрядах. Иными словами, из 32-разрядного слова таймера вычитается операнд 0000 0600. При переходе значения слова таймера от положительного к отрицательному (переполнение таймера) формируется внешнее прерывание, которое фиксируется в РФП по микроприказу $RFP[3] := 1$, и происходит переход к микропрограмме обслуживания прерывания. Если переполнение отсутствует, то происходит переход к подпрограмме ожидания.

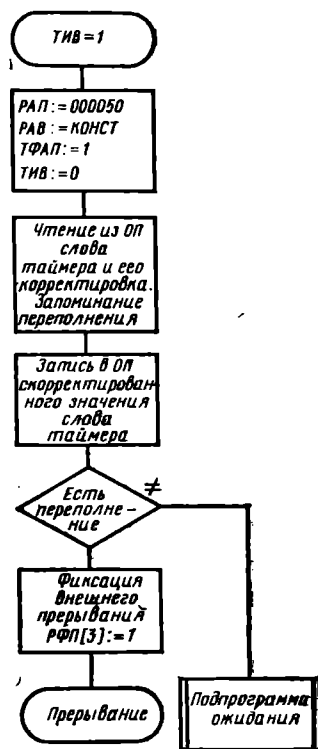


Рис. 67. Схема микропрограммы обработки таймера

Подпрограмма ожидания исследует значение РССП [14], определяющее состояние процессора: ОЖИДАНИЕ или СЧЕТ (прил. 7). Если РССП [14] = 0, то процессор находится в состоянии СЧЕТ и поэтому производится выборка очередной команды. В противном случае организуется цикл ожидания с анализом РССП [14] и анализом появления запроса на прерывание, которое может возникнуть. В состоянии ОЖИДАНИЕ команды не выполняются и обращения к ОП со стороны процессора отсутствуют.

Состояние ОЖИДАНИЕ может быть изменено внешним прерыванием или прерыванием от ввода-вывода.

Подпрограммой ожидания оканчиваются микропрограммы тех операций или пультовых процедур, в которых возможна смена текущего ССП или его части, ведущая к изменению состояния РССП [14].

4.3. ВЫБОРКА КОМАНД

В общем случае выполнение центральным процессором любой команды складывается из следующих этапов:

- 1) считывание команды из оперативной памяти;
- 2) подготовка адресов операндов;
- 3) выборка операндов;
- 4) обработка операндов;
- 5) запись результата.

Эта последовательность может быть нарушена в случаях, если на одном из этапов возникли определенные программные прерывания. В этом случае последующие этапы не выполняются, а управление передается микропрограмме обслуживания прерываний (п. 4.3);

если отсутствуют необходимые условия для успешного перехода в командах переходов. В этом случае этапы 2—5 не выполняются, а производится выборка очередной команды;

если при исполнении этапов 4 и 5 центральный процессор освободил ОП или для данной команды на этих этапах ОП не требуется. В этом случае происходит обращение к ОП для выборки очередной команды. Это позволяет полностью или частично совместить во времени исполнение этапа 1 с исполнением этапов 4 и 5;

если очередная считанная из ОП команда оканчивается на полуслове. Тогда считанное вместе с левым, правое полуслово будет представлять собой либо часть, либо полностью следующую команду. Эта информация засылается в 16 разрядный регистр РХК и используется затем в качестве очередной команды, что позволяет либо частично, либо полностью, в зависимости от формата, сократить этап 1. Признаком заполнения РХК служит единичное состояние триггера ТЗ РХК.

Так как этапы 1 и 2 в той или иной форме присутствуют при выполнении любой команды и являются общими для всего списка команд, то время их выполнения существенно сказывается на производительности процессора. Поэтому были предприняты меры по обеспечению выполнения этапов 1 и 2 с минимальными временными затратами. О двух из них уже упоминалось: это формирование опережающего запроса к ОП для выборки очередной команды на фоне выполнения этапов 4 и 5 (п. 3.4) и введение специального регистра РХК. Кроме этого, часть действий по выборке команды осуществляется аппаратурно-микропрограммными средствами. К ним относится проведение двух специальных анализов:

выборки — анализ 38 и неуспешной передачи управления — анализ 39 (см. табл. 17).

Микропрограмма выборки имеет ряд входов (рис. 68). Их использование зависит от расположения команды в слове (команда начинается с границы слова или с границы полуслова), от состояния оборудования ЦП (имеется ли информация в РХК, был ли

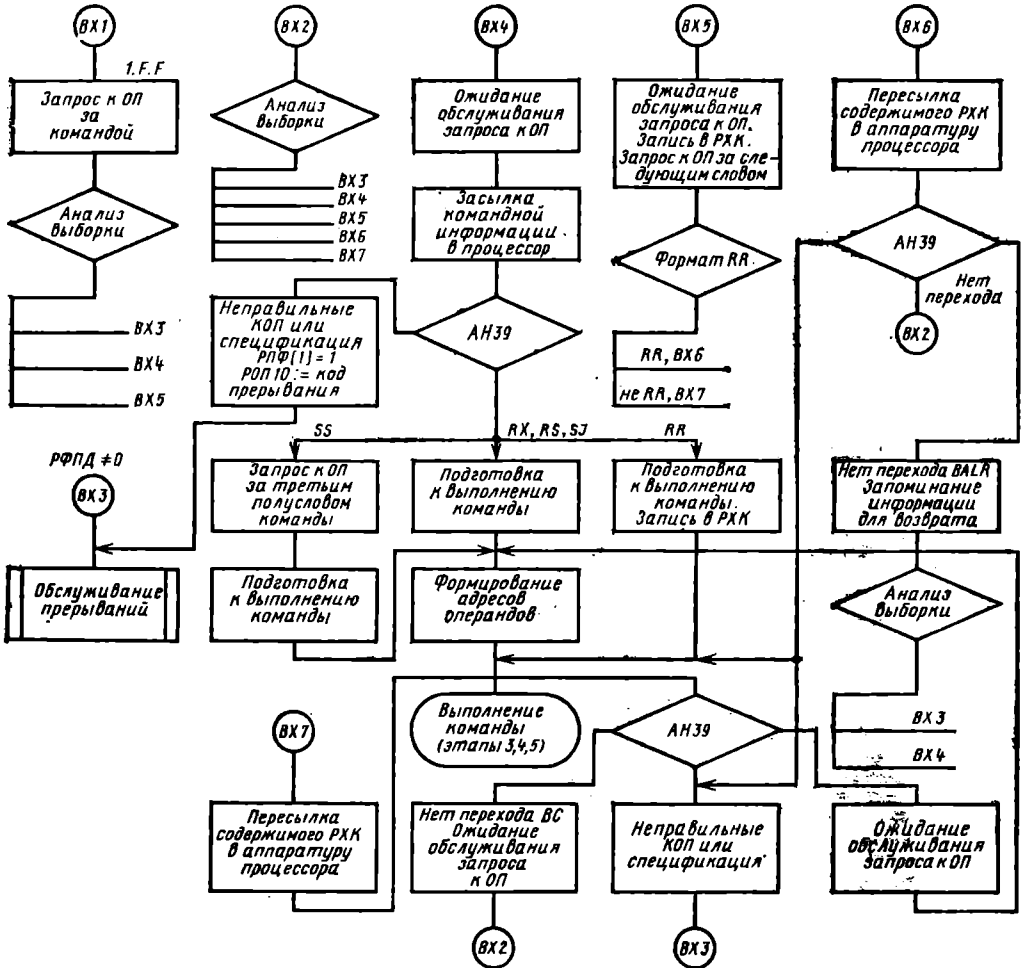


Рис. 68. Схема микропрограммы выборки команды

сделан запрос для чтения команды из ОП, имеются ли запросы на прерывания). Два входа VX1 и VX2 используются в зависимости от наличия запроса к ОП. Первый — когда для чтения команды запроса заранее сформировано не было, второй — когда запрос был сделан в предыдущей команде. Все остальные входы используются после анализа выборки (рис. 69), целью которого является исследование исходных данных. Анализ выполняется только в том случае, если запрос к ОП за командой был произведен заранее. Он является приоритетным и реализован аппаратно таким образом, что выполнение условий анализа, относящихся к младшим разрядам МАН, блокирует анализ условий,

относящихся к старшим разрядам. Так, например, наличие ненулевого состояния РФПД блокирует подключение к МАН других анализируемых условий выборки и т. д. Такое проведение анализа выборки дает возможность сократить количество разветвлений до пяти из возможных шестнадцати. Кроме этого сокращаются затраты времени на выборку команды. Например, при наличии информации в РХК оказывается возможным сразу выделить команду формата RR. Исследование информации осуществляется в такой последовательности:

— РФПД $\neq 0$, используется ВХ3 для перехода к обработке прерываний (п. 4.3);

| [0] | [1] | [2] | [3] |
|------------|-------------|----------|---------------|
| РССП[62]=1 | РХК[0/1]=00 | ТЗ РХК=1 | РФПД $\neq 0$ |

Рис. 69. Информация, исследуемая при проведении анализа выборки

— РФПД = 0, ТЗРХК = 0, РССП [62] = 0 означает, что команда начинается с границы слова и в РХК информация отсутствует, используется ВХ4;

— РФПД = 0, ТЗРХК = 0, РССП [62] = 1 означает, что команда начинается с правой половины слова ОП, используется ВХ5;

— РФПД = 0, ТЗРХК = 1, РХК [0/1] = 00, т. е. очередная команда полностью имеется в РХК, так как состояние «00» двух старших разрядов кода операции определяет формат RR. В этом случае этап I полностью исключается, используется ВХ6;

— РФПД = 0, ТЗРХК = 1, РХК [0/1] $\neq 00$, т. е. в РХК хранится только часть команды и требуется дополнительное считывание команды из оперативной памяти, используется ВХ7.

По микроприказу АН38 на МАН формируются коды 0001, 0000, 1000, 0110, 0010, определяющие входы ВХ3, ВХ4, ВХ5, ВХ6, ВХ7 соответственно.

Эти коды используются для модификации младшей тетрады базового адреса. Сформированный адрес и определяет соответствующий вход в микропрограмму выборки.

Анализ выборки, сделанный при входе в ВХ2, определяет все перечисленные выше входы ВХ3—ВХ7. Этот же анализ, сделанный при входе в ВХ1, не может определять ВХ6 и ВХ7, поскольку, если заранее не было дано запроса за командой, информация в РХК игнорируется и из ОП выбирается полное слово, содержащее команду.

Анализ неуспешной передачи управления предназначен для определения некорректности задания команд и данных, для определения условий перехода в командах переходов и для выбора направления разветвления по формату команды. По микроприказу АН39 на МАН могут сформироваться следующие коды: 1111 — в зоне кода операции текущей команды встретился код, отсут-

ствующий в списке команд (неправильный КОП); 0111 — адреса операндов текущей команды заданы некорректно (неправильная спецификация); 0101 — отсутствие перехода в командах условного перехода BC; 0100 — отсутствие перехода в командах условного перехода BCR; 0110 — отсутствие перехода в команде перехода с возвратом BALR; коды 0000, 0001, 0010, 0011 определяют формат текущей команды RR, RX, RS, SI или SS соответственно (см. прил. 5).

Анализы 38 и 39 выполняются за один такт работы процессора. Они располагаются в различных ветвях микропрограммы выборки команды.

Вход в микропрограмму по VX1 осуществляется после начальной загрузки, работы с пульта, записи в ОП при исполнении предыдущей команды, т. е. в тех случаях, когда заранее не был сформирован запрос к ОП для выборки команды. В любом из этих случаев запрос к ОП формируется микропрограммой выборки и по результату анализа 38 определяется наличие запросов прерываний и расположение команды в слове оперативной памяти. Содержимое РХК не проверяется, так как в этом случае оно игнорируется.

При входе через VX2, поскольку запрос за командой был сделан заранее (в микропрограмме, предшествующей выборке команд), производится анализ выборки.

Переход на VX3 осуществляется после АН38. Это имеет место в том случае, если РФПД $\neq 0$, т. е. имеются необработанные запросы на прерывание. Процедура обслуживания прерываний, включающая в себя также корректировку, рассмотрена в п. 4.3.

Ветвь микропрограммы, начиная с VX4, функционирует в том случае, если предыдущая команда завершилась анализом выборки и в ней был сделан опережающий запрос за текущей командой, которая начинается с границы слова оперативной памяти. Командная информация, считанная из ОП, может быть командой форматов RX, RS, SI частью команды формата SS или текущей командой формата RR и частью следующей за ней команды. После считывания из ОП информации она засылается в соответствующие блоки процессора. Так, биты команды, определяющие код операции, поступают в РДШК, биты [8/15], являющиеся либо адресами РОН или РПЗ, либо непосредственным операндом, либо кодом длины операндов, в зависимости от формата команды, передаются в СТ РОН. Вся командная информация помещается также в операционные регистры РОП1 и РОП3 и производится анализ 39.

При появлении несуществующего кода операции или нарушения спецификации в задании адресов операндов микропрограмма фиксирует программное прерывание; устанавливается в единичное состояние РФП [1], формируется соответствующий код прерывания, который записывается в РОП10, и управление через VX3 передается подпрограмме обслуживания прерываний.

При невыполнении условий перехода для команд BC, BCR и BALR, формируется запрос к ОП за очередной командой. При наличии команд типа BCR и BALR оставшееся неиспользованным правое полуслово засылается в РХК и управление передается ветви ВХ2, где производится анализ выборки для очередной команды. В остальных случаях в результате анализа 39 определяется формат команды и осуществляется соответствующее разветвление микропрограммы:

при формате RR, когда длина команды равна одному полуслову, оставшееся неиспользованным правое полуслово засылается в РХК, устанавливается в единичное состояние ТЗ РХК и микроприказом РАБОП := РДШК (табл. 17) управление передается микропрограмме выполнения конкретной команды. По этому микроприказу содержимое РДШК используется в качестве адреса блока односторонней памяти АБОП;

при форматах RX, RS, SI один из операндов хранится в оперативной памяти. Адрес операнда неявно задается смещением и базой, а для формата RX еще и индексом. Поэтому вначале осуществляется формирование фактического адреса операнда, хра-

Т а б л и ц а 21

Состояние регистров ЦП после выборки при различных форматах команд (микропрограмма исполнялась, начиная с ВХ4)

| Оборудование | RR | RX | RS | SI | SS |
|--------------|--------------|-----------|-----------|-----------|-----------|
| РОП0 | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 | 0000 0000 |
| РОП1 | 0Ч.К. Т.К. | Т.К. | Т.К. | Т.К. | — 3П.С. |
| РОП3 | Т.К. 0Ч.К. | Т.К. | Т.К. | Т.К. | Т.К. |
| РМС1 | 0000 000F | 0000 000F | 0000 000F | 0000 000F | 0000 000F |
| РМС2 | 00FF FFFF | 00FF FFFF | 00FF FFFF | 00FF FFFF | 00FF FFFF |
| РДШК | КОП | КОП | КОП | КОП | КОП |
| СТ РОИ | R1, R2 | R1, X2 | R2, R3 | I2 | L1, L2 |
| РОП14 | — | — | — | — | A1 |
| РОП15 | — | A2 | A2 | A1 | A2 |
| ТЗВ | 1 | 1 | 1 | 1 | 1 |
| ТЗ РХК | 1 | 0 | 0 | 0 | 0 |
| РХК | 0Ч.К. | — | — | — | — |

нящегося в ОП, после чего происходит переход на выполнение операции. Адрес операнда запоминается в РОП15;

при формате SS длина команды равна трем полусловам. Поэтому необходимо еще одно обращение к ОП для считывания третьего полуслова команды. Правая половина дополнительно считанного слова в РХК не заносится, так как в командах этого формата результат всегда записывается в ОП, где он может быть помещен в зону размещения очередной команды. После формирования адресов операндов переход на микропрограмму выполнения команды осуществляется по микроприказу РАБОП := РДШК, а адреса операндов запоминаются в РОП14 и РОП15.

Как отмечалось выше, третий этап (выборка операндов) является принадлежностью микропрограмм выполнения непосредственно операций. Однако в микропрограмме выборки, при переходе по коду операции, сигнал РАБОП := РДШК аппаратурно

Т а б л и ц а 22

Состояние регистров ЦП после выборки при различных форматах команд (микропрограмма исполнялась, начиная с ВХ6 или ВХ7)

| Оборудование | RR | RX | | RS | | SI | | SS | |
|--------------|-----------|-----------|------|-----------|------|-----------|------|-----------|------|
| РОПО | 0000 0000 | 0000 0000 | | 0000 0000 | | 0000 0000 | | 0000 0000 | |
| РОП1 | — | — | 2П.С | — | 2П.С | — | 2П.С | — | 2П.С |
| РОП3 | Т.К | — | 1П.С | — | 1П.С | — | 1П.С | — | 1П.С |
| РМС1 | — | — | | — | | — | | 0000 000F | |
| РМС2 | 00FF FFFF | 00FF FFFF | | 00FF FFFF | | 00FF FFFF | | 00FF FFFF | |
| РДШК | КОП | КОП | | КОП | | КОП | | КОП | |
| СТ РОИ | R1, R2 | R1, R2 | | R2, R3 | | I2 | | L1, L2 | |
| РОП14 | — | — | | — | | — | | A1 | |
| РОП15 | — | A2 | | A2 | | A1 | | A2 | |
| ТЗВ | 0 | 1 | | 1 | | 1 | | 1 | |
| ТЗ РХК | 0 | 1 | | 1 | | 1 | | 0 | |
| РХК | — | ОЧ.К | | ОЧ.К | | ОЧ.К | | — | |

Примечание: Т.К. — текущая команда; ОЧ.К — очередная команда; 1 П.С. — первое полуслово текущей команды; 2 П.С. — второе полуслово текущей команды; 3 П.С. — третье полуслово текущей команды; A1 — адрес первого операнда; A2 — адрес второго операнда; ТЗВ — триггер запроса выборки, единичное состояние которого разрешает формирование запроса к ОП для считывания команды.

формирует для большинства команд запрос ЗПРС к ОП для считывания операнда. Тип запроса определяется кодом операции (п. 3.4).

Исполнение ветвей микропрограммы выборки, начинающихся со входов ВХ5, ВХ6, ВХ7, осуществляется аналогично рассмотренным выше. Однако здесь учитывается, имеется ли в РХК подвыбранная команда, какого она формата и каким образом она располагается в слове ОП.

Микропрограмма выборки команд, кроме того, подготавливает оборудование ЦП для выполнения команд, размещая в регистрах различные константы, необходимые для выполнения тех или иных операций (табл. 21, 22).

ГЛАВА 5

КОМАНДЫ УПРАВЛЕНИЯ

Команды, исполняемые центральным процессором, выбираются из последовательно расположенных ячеек оперативной памяти. Адреса этих ячеек определяются значениями разрядов [40/63] текущего ССП. Для организации разветвляющихся и циклических вычислительных процессов необходимо иметь возможность изменять линейную последовательность выполнения команды, т. е. по результатам анализа определенных программных условий выбирать то или иное направление продолжения программы или повторять какую-то часть программы. Кроме того, возникает необходимость маскировать на некоторое время различные прерывания, переводить ЦП из одного состояния в другое, организовывать обмен между устройствами ввода-вывода и оперативной памятью и т. д. Все эти действия выполняются с помощью команд управления, которые подразделяются на команды переходов, команды переключения состояния и команды ввода-вывода.

5.1. ПЕРЕХОДЫ

При выполнении команд переходов (прил. 4) продвинутый адрес очередной команды, расположенный в разрядах [40/63] текущего ССП, замещается адресом перехода, с которого программа и будет продолжаться. Переходы могут быть условные, когда замещение продвинутого адреса производится только при выполнении определенных условий, и безусловными, в которых продвинутый адрес всегда замещается адресом перехода.

Для команд условного перехода ВС, ВСR и перехода с возвратом BALR выполнение условий для реализации передачи управления анализируется в процессе выборки команды (п. 4.4). При от-

существовании условий перехода выполнение указанных команд прекращается, т. е. обращение к микропрограммам выполнения этих команд происходит только при однозначном замещении продвинутого адреса адресом перехода. В этом случае микропрограмма выборки обязательно гасит ТЗ РХК, так как при переходе информация, хранящаяся в РХК, не нужна.

Алгоритмы выполнения микропрограмм команд перехода по содержимому счетчика ВСТР и ВСТ также просты и заключаются в вычитании «1» из содержимого регистра РОН по адресу, определяемому полем $R1$, и анализе результата вычитания на нуль. Если результат не равен нулю, то продвинутый адрес замещается адресом перехода, а если равен нулю, то переход отсутствует и выполняется следующая по порядку команда. Здесь, в случае успешного перехода, также сбрасывается в нуль ТЗ РХК. При выполнении команды ВСТР дополнительно анализируется на нуль поле $R2$, и если оно равно нулю, то вычитание «1» из содержимого регистра, определяемого полем $R1$, производится, но переход отсутствует.

С точки зрения организации микропрограмм определенный интерес представляют очень простые микропрограммы команд переходов по индексу. В качестве примера рассмотрим микропрограмму выполнения команды ПЕРЕХОД ЕСЛИ ИНДЕКС БОЛЬШЕ — ВХН (86) формата RS. Из всех команд переходов в ней наиболее полно используется аппарат ветвлений микропрограмм.

При исполнении этой команды приращение (третий операнд) складывается с первым операндом. Сумма алгебраически сравнивается со значением индекса, определяемого содержимым поля $R3$, если оно нечетное, или полем $R3 + 1$, если содержимое $R3$ четное. Чаще оно четное, поэтому сравнивается сумма содержимого регистров, определяемых полями $R1$ (первый операнд) и $R3$ (приращение) с содержимым регистра, определяемым полем $R3 + 1$ (индекс). Если сумма больше значения индекса, продвинутый адрес замещается адресом перехода, если меньше или равна, переход отсутствует. Сумма приращения и первого операнда всегда помещается на место первого операнда. Если адреса первого операнда и индекса совпадают, то для сравнения берется первоначальное значение первого операнда.

Схемы микропрограммной логики изображают последовательность выполнения алгоритма в виде последовательности выполнения микрокоманд. В дальнейшем такие схемы будут называться просто микропрограммой (рис. 70, 71, 74, 75). При этом используются два блочных символа: «Процесс» и «Решение», которые изображаются в виде прямоугольника и ромба соответственно (рис. 70). В поле блочного символа описывается действие микрокоманды, представленное в виде микроприказов. Поле блочного символа «Процесс» разделено горизонтальной чертой на две части. Микроприказы, расположенные в верхней части символа,

выполняются в первом полутакте по синхросерии С1 или являются потенциальными и действуют в течение всего такта. Микроприказы, записанные в нижней части, выполняются во втором полутакте по синхросерии С2.

Цифры над правым верхним углом любого блочного символа указывают адрес памяти микропрограмм отображаемой микрокоманды. Блочный символ «Решение» всегда относится к символу «Процесс», после которого он следует и составляет вместе с ним единую микрокоманду. В поле этого символа записан номер анализа, который определяет подлежащие анализу условия согласно табл. 17—19. Анализы всегда выполняются в первом полутакте по синхросерии С1. Цифра, стоящая в скобках после номера анализа, означает соответствующий признак ветвления ПРВ. Он позиционно определяет те разряды последней тетрады базового адреса, которые могут быть модифицированы по результатам анализа. Базовый адрес записывается внизу справа от блочного символа «Решение». При ветвлении по результату анализа на два направления (например, микрокоманда 0.8.6) последние указываются как при выполнении, так и при невыполнении условия анализа. Эти случаи обозначаются знаками = и ≠ соответственно. При ветвлении более чем на два направления (например, микрокоманды 6.8.8 и 6.8.С) на полке под блочным символом «Решение» указываются значения модифицируемых разрядов базового адреса.

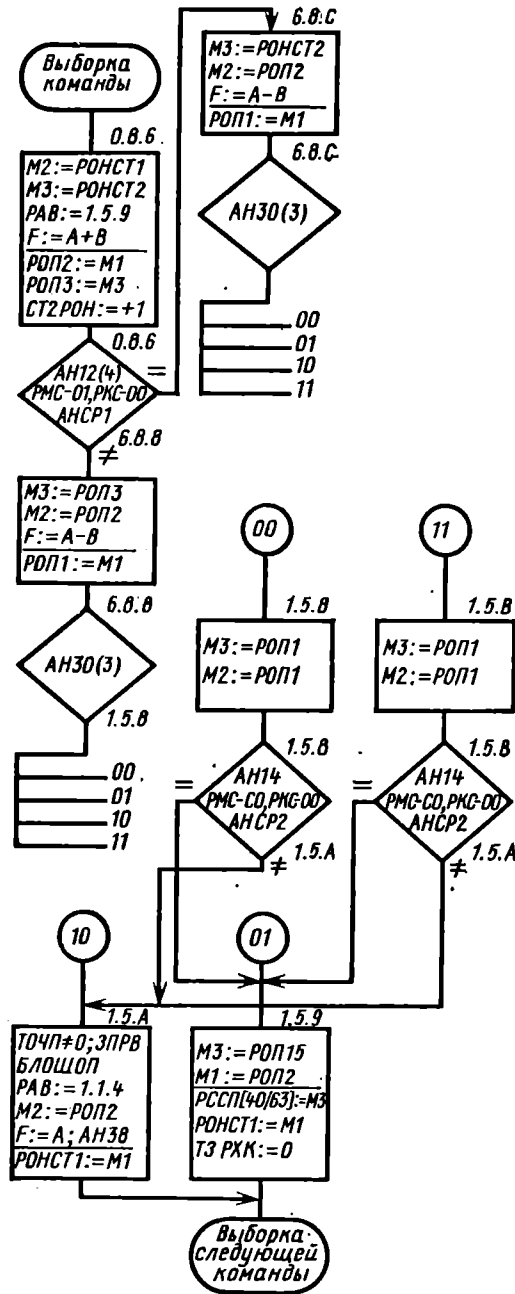


Рис. 70. Микропрограмма команды ПЕРЕХОД ЕСЛИ ИНДЕКС БОЛЬШЕ — ВХН (86)

зываются значения модифицируемых разрядов базового адреса.

Начальным адресом микрокоманды выполнения соответствующей команды является код операций. В первой микрокоманде с адресом 0.8.6 первый операнд и приращение читаются из РОН

на магистрали М2 и М3. Номера РОН определяются содержимым его счетчиков: соответственно СТ1 и СТ2 ($M2 := \text{РОНСТ1}$, $M3 := \text{РОНСТ2}$). Для АЛУ задается функция сложения ($F := A + B$). Результат выдается на М1 и с нее записывается в РОП2 ($M1 := \text{РОП2}$). В РОП3 с М3 записывается ($\text{РОП3} := M3$) содержимое регистра, адрес которого определяется полем *R3* команды (хранится в СТ2 РОН). Оно может быть использовано в дальнейшем, если окажется, что содержимое поля *R3* нечетное. Поскольку индекс всегда хранится в нечетном регистре, то на случай, если содержимое поля *R3* окажется четным, значение СТ2 РОН увеличивается на единицу ($\text{СТ2 РОН} := +1$). В РАВ засылается адрес возврата для осуществления последующего разветвления. В этой же микрокоманде производится анализ на четность содержимого поля *R3*, которое на этапе выборки команды было помещено в СТ2 РОН (п. 4.4). Для этой цели используется один из анализов, выполняемых сравнением — анализ 12 (см. табл. 19) вида АН СР1. По результату этого анализа осуществляется переход по базовому адресу (напомним, что по результату анализа вида АН СР2 происходит переход по адресу возврата из РАВ). Так как при проведении анализа 12 проверяется состояние двух счетчиков — СТ1 РОН и СТ2 РОН, то на значение первого должна быть наложена нулевая маска. Поскольку интерес представляет значение четности содержимого второго, т. е. значение СТ2 [3], то должна быть выбрана маска с единицей в последнем разряде и с нулями в остальных. Четное значение СТ1 РОН означает, что в его последнем разряде — нуль. Этим и определяется выбор константы сравнения. Ветвление должно произойти на два направления (четное и нечетное значение счетчика), следовательно, необходимо модифицировать только один разряд базового адреса следующей микрокоманды. Исходя из этих условий при базовом адресе 6.8.8 и значениях *РМС* — 0000 0001, *РКС* — 0000 0000, *ПРВ* — 0100 по микроприказу АН12 (4) формирование следующего адреса будет происходить по схеме:

| | | |
|------------------------------------|---------------|---------------|
| код сравнения (поле <i>РКС</i>) | а) 0000 0000, | б) 0000 0000, |
| маска сравнения (поле <i>РМС</i>) | 0000 0001, | 0000 0001, |
| код счетчиков (код условия) | xxxx xxx1, | xxxx xxx0, |
| МАН[0/3] | 1010, | 1111, |
| <i>ПРВ</i> | 0100, | 0100, |
| базовый адрес | 6.8.1000, | 6.8.1000, |
| адрес БОП | 6.8.1000; | 6.8.1100, |

где *x* — любое содержимое разряда. В случае *a* только левая тетрада кода условия равна соответствующей тетраде кода сравнения, в результате чего формируется следующий адрес 6.8.8: в случае *б* обе тетрады кода условия равны соответствующим тетрадам кода сравнения и формируется адрес 6.8.С.

Если код в поле *R3* нечетный, модификации адреса не будет. Поэтому следующей будет исполняться микрокоманда 6.8.8.

По ней значение индекса, которое хранится в РОПЗ, вычитается из значения суммы первого операнда и приращения, находящегося в РОП2, и разность записывается в РОП1. Одновременно необходимо исследовать знаки суммы и индекса, т. е. в данном случае — значения знаковых разрядов М2 и М3. Для этого используются две последние проверки анализа З0 (см. табл. 17) с признаком ПРВ = 0011 (микроприказ, АНЗ0 (3)). По результату анализа возможны такие случаи модификации двух последних разрядов базового адреса 1.5.1000 следующей микрокоманды:

Модификация 01 (переход на микрокоманду 1.5.9) — знаки различны и сумма положительна. Так как она больше отрицательного индекса, то продвинутый адрес должен быть замещен адресом перехода. Этим адресом является адрес второго операнда, хранящийся после выборки команды в РОП15 (см. табл. 21, 22). Через магистраль М3 он пересылается в РССП [40/63]. Одновременно сумма приращения и первого операнда через магистраль М1 из РОП2 записывается на место первого операнда (РОНСТ1 := М1). Поскольку переход успешный, то выбранная информация о следующей команде, хранящаяся в РХК, становится недействительной и для сообщения об этом сбрасывается ТЗ РХК. Так как микропрограммой выборки команды для команды ВХН был сделан запрос к ОП для чтения второго операнда, который в этом случае будет являться очередной командой, то завершение микропрограммы данной команды осуществляется передачей управления на соответствующую ветвь микропрограммы выборки команды.

Модификация 10 (переход на микрокоманду 1.5.А) — знаки различны и сумма отрицательна. Так как она меньше положительного индекса, то переход отсутствует. В этом случае ожидается завершение обслуживания запроса к ОП (ТОЧП \neq 0), сформированного микропрограммой выборки команд. Так как второй операнд, считанный из ОП, не используется, то сигналом блокировки ошибок оперативной памяти БЛ ОШ ОП осуществляется сброс триггеров, которые хранят сигналы ошибок адресации, защиты и спецификации. Эти ошибки могли возникнуть при обращении за вторым операндом. Одновременно формируется запрос к ОП для выборки следующей команды — ЗПРВ. Из РОП2 через магистраль М2 и БАЛ сумма приращения и первого операнда помещается на место первого операнда и производится анализ выборки (анализ З8), который и определяет точку входа в микропрограмму выборки следующей команды.

Модификация 00 (переход на микрокоманду 1.5.8) — сумма и индекс имеют одинаковые знаки — положительные. В этом случае необходимо провести дополнительный анализ для определения выполнения условий перехода. Ими является знак разности суммы и индекса, которая хранится в РОП1, и неравенство нулю целой части этой разности. Необходимый операнд из РОП1 читается одновременно на М2 и М3 и путем проведения анализа 14 проверяется значение знакового разряда М3 (М3 [0]) и равенство

нулю содержимого M2 [1/31]. При этом возможны четыре варианта сочетаний условий анализа и соответствующих им кодов:

| Условия анализа | Код условия |
|----------------------|-------------|
| M3[0]=0 ∧ M2[1/31]≠0 | 00, |
| M3[0]=0 ∧ M2[1/31]=0 | 01, |
| M3[0]=1 ∧ M2[1/31]≠0 | 10, |
| M3[0]=1 ∧ M2[1/31]=0 | 11. |

Анализ выполняется по схеме:

| | | |
|----------------------------|---------------|---------------|
| код сравнения (поле РКС) | а) 0000 0000, | б) 0000 0000, |
| маска сравнения (поле РМС) | 1100 0000, | 1100 0000, |
| код условия | 00XX XXXX, | YZXX XXXX, |
| МАН[0/3] | 1111; | 0101. |

где X означает, что значение разряда игнорируется; YZ — значения кода условия, равные 01, 10 или 11.

Чтобы обеспечить возможность разветвления по двум направлениям (код условия 00 и все остальные) вместо возможных четырех, применяется анализ путем сравнения вида АН СР2. При этом в случае сравнения двух тетрад кодов сравнения и условия происходит безусловный переход по адресу возврата 1.5.9, который ранее был помещен в РАВ. В противном случае происходит переход по базовому адресу 1.5.А.

Модификация 11 (переход на микрокоманду 1.5.В) — сумма и индекс имеют одинаковые знаки — отрицательные. Все действия аналогичны рассмотренным в микрокоманде 1.5.8.

Если код поля R3 четный, то выполняется микрокоманда 6.8.С, по которой индекс считывается из регистра с номером R3 + 1, хранящимся в СТ2 РОИ. Все последующие действия аналогичны рассмотренным в микрокоманде 6.8.8.

Алгоритм выполнения микропрограммы команды ПЕРЕХОД ЕСЛИ ИНДЕКС МЕНЬШЕ ИЛИ РАВЕН — ВХЛЕ (87) в основном аналогичен рассмотренному. Отличие состоит в том, что при анализе суммы и индекса условия обратные, т. е. продвинутый адрес замещается адресом перехода тогда, когда сумма приращения и первого операнда меньше или равна значению индекса.

5.2. ПЕРЕКЛЮЧЕНИЯ СОСТОЯНИЙ

Работа процессора как элемента системы в любой момент времени определяется некоторой совокупностью признаков, к числу которых, в первую очередь, относятся несколько пар взаимно исключающих программных состояний:

ЗАДАЧА или СУПЕРВИЗОР;

ОЖИДАНИЕ или СЧЕТ;

ПРЕРЫВАНИЕ ЗАМАСКИРОВАНО или ПРЕРЫВАНИЕ РАЗРЕШЕНО.

Кроме того, состояние ЦП в ходе исполнения любой программы характеризуется адресом команды, признаком результата, кодом длины команды, кодом прерывания и ключом защиты памяти. Вся информация о перечисленных признаках размещается в определенных полях и разрядах ССП (прил. 7). В каждый очередной

момент работы ЭВМ то или иное программное состояние может быть задано в определенной степени независимо. Например, состояния ЗАДАЧА или СУПЕРВИЗОР определяют, все ли команды списка (прил. 4) являются допустимыми. В состоянии СУПЕРВИЗОР все команды допустимы, в состоянии ЗАДАЧА только часть из них. Состояния ОЖИДАНИЕ или СЧЕТ определяют: выполняются ли команды, что соответствует состоянию СЧЕТ, или нет — в состоянии ОЖИДАНИЕ.

Для обеспечения возможности оперативного изменения состояний и осуществления связи между машинами в системе ЕС ЭВМ предусмотрены специальные команды (прил. 4). Рассмотрим эти команды и особенности их выполнения.

ЗАГРУЗКА ССП — LPSW (82), формат SI. По этой команде слово двойной длины из ячейки, указанной адресом операнда, замещает текущее ССП. При этом содержимое зоны кода прерывания (разряды [16/31]) и кода длины (разряды [32/33]) не изменяются.

Адрес первой микрокоманды 0.8.2 определяется кодом операции (рис. 71). В начале операции необходимо определить состояние процессора, поскольку команда LPSW привилегированная и выполняется только

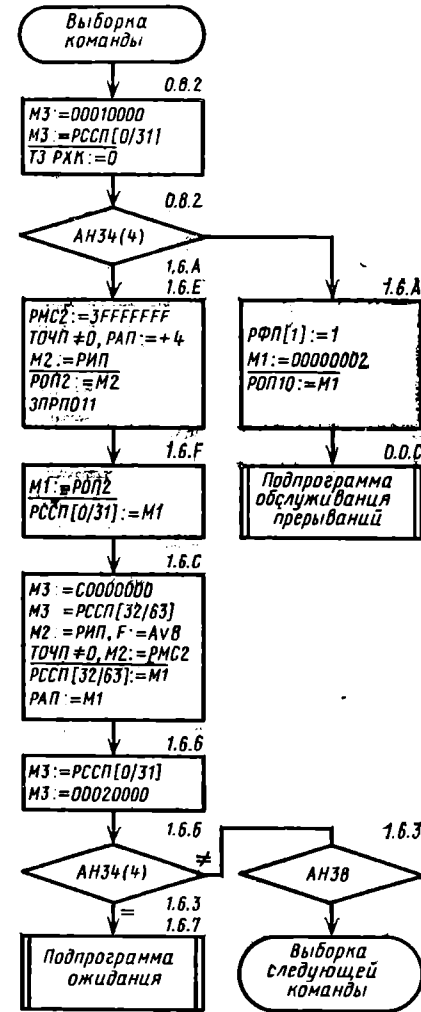


Рис. 71. Микропрограмма команды ЗАГРУЗКА ССП—LPSW (82)

в состоянии СУПЕРВИЗОР. Для этого необходимо определить значение РССП[15] текущего ССП (прил. 7). С этой целью на М3 считывается первое слово РССП. Одновременно на М3 из БОПЗ считывается маска, содержащая нули во всех разрядах, кроме 15-го. В результате все разряды М3 принимают нулевое значение, а разряд 15 — значение РССП [15]. Далее исследуется на нуль содержимое М3. Это делается с помощью анализа 34 (см. табл. 17), причем для выделения нужной проверки

(M3 = 0) ПРВ должен быть равен 0100. Наличие одной единицы в ПРВ обеспечивает ветвление на два направления путем модификации базового адреса 1.6.A. Одновременно сбрасывается ТЗ РХК (ТЗРХК := 0), поскольку при смене ССП подвыбранная команда оказывается не нужна. При РССП [15] ≠ 0 в микрокоманде 1.6.A фиксируется программное прерывание с кодом 02, так как привилегированная команда недопустима к исполнению в состоянии ЗАДАЧА. При этом формируется адрес БОП 0.0.C и происходит переход на подпрограмму обработки прерывания (см. рис. 66), которое обрабатывается немедленно. Если же ЦП находится в состоянии СУПЕРВИЗОР, эта команда допустима.

По микрокоманде 1.6.E ожидается конец обслуживания оперативной памятью запроса процессора (ТОЧП ≠ 0, п. 2.1) за первым словом нового ССП. В качестве запроса за операндом он был сделан на этапе выборки команды. После того как запрос обслужен и информация из ОП получена в РИП, она (в данном случае — первое слово ССП) через M2 записывается в РОП2 (M2 := РИП, РОП2 := M2). Одновременно адрес обращения к ОП увеличивается на 4 по микроприказу РАП := +4 и дается запрос к ОП для считывания старшего слова ССП (ЗПРП011). В регистр маски РМС2 заносится константа из БОП3 (РМС2 := 3FFF FFFF), которая в дальнейшем используется для обнуления значения РССП [32/33], т. е. зоны кода длины команды в ССП. В микрокоманде 1.6.F в разряды [0/31] текущего ССП поступает информация, считанная из ОП. Затем ожидается завершение считывания младшего слова ССП (ТОЧП ≠ 0, 1.6.C), после чего оно выдается на магистраль M2 (M2 := РИП). Одновременно на M2 читается маска из РМС2 (M2 := РМС2). Тем самым на M2 формируется младшее слово ССП с обнуленной зоной кода длины команды. Для окончательного формирования первого слова ССП на M3 из БОП3 читаются маска с единицами в двух старших разрядах и нулями во всех остальных (M3 := C000 0000) и одновременно второе слово старого ССП (M3 := РССП [32/63]). Задавая БАЛ функцию ИЛИ (F := A ∨ B), получим в результате на M1 младшее слово ССП, считанное из ОП, в разрядах [32/33] которого находится код длины, соответствующий коду длины команды старого ССП. Это слово записывается в РССП [32/63] (РССП [32/63] := M1) и одновременно значение РССП [40/63], определяющее адрес очередной команды, помещаются в РАП (РАП := M1). Поскольку в новом ССП бит 14, определяющий состояние ОЖИДАНИЕ или СЧЕТ, может быть изменен, то по микрокоманде 1.6.6 анализируется его значение. Если оно равно единице, то осуществляется переход к подпрограмме ожидания (п. 4.3), если же нулю — производится анализ выборки АН38 и команда завершается.

УСТАНОВИТЬ МАСКУ ПРОГРАММЫ — SPM (04), формат RR. Значение разрядов [2/7] регистра, определяемого полем R1, замещает содержимое разрядов [34/39] текущего ССП. Выделение

значений разрядов РОН [2/7] и РССП [34/39] осуществляется путем маскирования магистралей, аналогично рассмотренному в микропрограмме команды ЗАГРУЗКА ССП.

УСТАНОВИТЬ МАСКУ СИСТЕМЫ — SSM (80) формат S1. Байт, определяемый адресом операнда в ОП, замещает биты ССП [0/7], т. е. биты маски системы. Эта команда привилегированная и выполняется только в состоянии СУПЕРВИЗОР. Анализ состояния ЗАДАЧА или СУПЕРВИЗОР и формирование кодов замещения аналогичны рассмотренному в микропрограмме команды ЗАГРУЗКА ССП. Поэтому рассмотрим только

часть процесса выполнения команды, связанную с перемещением байта информации, считанного из ОП (он может быть расположен в позиции любого байта слова) в позицию первого байта слова, где размещена маска системы. Это производится с помощью аппарата ЛРИП (рис. 72), рассмотренного в п. 3.4. По микроприказу M1 := ЛРИП в считанном из ОП и переданном в РИП слове, байт, код позиции (10) которого был заранее занесен в РСБФ, перемещается на место младшего байта магистрали M1, откуда он записывается в младший байт промежуточного операционного регистра. Затем через M2 перемещаемый байт записывается на место младшего байта старшей половины двухсловного сдвигателя (в РСДВ1) и сдвигается вправо на 8 разрядов. После этого он оказывается на месте старшего байта младшего слова сдвигателя (в РСДВ2). Из РСДВ2 рассматриваемый байт считывается на магистраль M3, откуда он заносится в РССП [0/7], замещая прежнее значение маски системы.

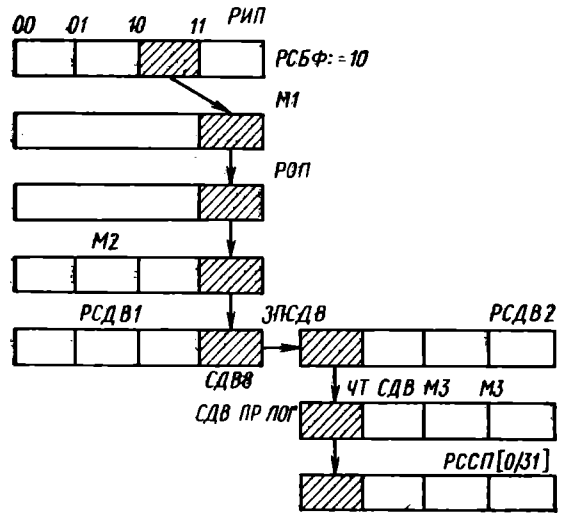


Рис. 72. Перемещение байта информации с использованием ЛРИП и сдвигателя

ного операционного регистра. Затем через M2 перемещаемый байт записывается на место младшего байта старшей половины двухсловного сдвигателя (в РСДВ1) и сдвигается вправо на 8 разрядов. После этого он оказывается на месте старшего байта младшего слова сдвигателя (в РСДВ2). Из РСДВ2 рассматриваемый байт считывается на магистраль M3, откуда он заносится в РССП [0/7], замещая прежнее значение маски системы.

ОБРАЩЕНИЕ К СУПЕРВИЗОРУ — SVC (0A), формат RR. При выполнении команды формируется прерывание соответствующего класса (пп. 3.2, 4.3), а содержимое зоны [8/15] команды заносится в разряды [24/31] текущего ССП в качестве кода прерывания. После этого управление передается микропрограмме обслуживания прерываний.

УСТАНОВИТЬ КЛЮЧ ПАМЯТИ — SSK (08), формат RR. Содержимое разрядов [24/28] регистра, определяемого полем R1, замещает в ПКЗ ключ блока (страницы) памяти, адрес которого определяется разрядами [13/20] регистра, определяемого полем R2.

ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ — ISK (09), формат RR. Эта команда вызывает действия, обратные действиям команды **УСТАНОВИТЬ КЛЮЧ ПАМЯТИ**. Ключ блока (страницы) памяти, который адресуется регистром, определяемым полем R2, помещается в регистр, определяемый полем R1. Команды SSK и ISK выполняются по единой микропрограмме. Для управления записью в ПКЗ имеется специальный триггер записи — ТЗ ПКЗ, который устанавливается в 1 по микроприказу ТЗПКЗ := 1 только при РДШК[7] = 0, т. е. при наличии в РДШК кода операции команды **УСТАНОВИТЬ КЛЮЧ ПАМЯТИ**.

ПРЯМАЯ ЗАПИСЬ — WRD (84), формат SI. Байт из ОП, определяемый адресом первого операнда, передается в регистр выдачи информации внешних линий РВИВЛ, а восемь бит непосредственного операнда (содержимое поля I2) пересылаются в регистр выдачи синхроимпульсов внешних линий РВСИВЛ для использования их в качестве синхросигналов передаваемой информации. Команда привилегированная и может исполняться только тогда, когда ЦП находится в состоянии СУПЕРВИЗОР. Для сообщения другому процессору о том, что во внешние линии выдается байт информации, байт синхросигналов помещается в РВСИВЛ. Через 600 нс синхросигналы снимаются, что говорит о том, что данные, находящиеся на внешних линиях, действительны и могут быть приняты другим процессором.

ПРЯМОЕ ЧТЕНИЕ — RDD (85), формат SI. Байт поля I2 посылается во внешние линии в качестве синхросигналов. При отсутствии на входной линии блокирующего сигнала из другого ЦП процессор, выполняющий команду, в течение 600 нс принимает байт информации из внешних линий. Принятый байт помещает в ячейку ОП, определяемую адресом первого операнда. Одновременно с синхросигналами во внешнюю линию посылается сигнал чтения. Байт синхросигналов, посылаемый во внешние линии в командах WRD и RDD, используется в качестве кода внешнего прерывания. Эти команды дают возможность организовывать связи в многомашинных системах, используя средства прямого управления (п. 3.3) на уровне межпроцессорного обмена. Следует заметить, что в качестве внешнего источника, передающего информацию по шинам прямого управления, не обязательно должен быть аналогичный ЦП. Источником информации могут быть и какие-либо специализированные устройства. Важно лишь то, чтобы они обеспечивали необходимую последовательность передачи сигналов.

ПРОВЕРИТЬ И УСТАНОВИТЬ — TS (93), формат SI. Старший бит байта, определяемого адресом первого операнда, используется для формирования признака результата в ССП [34/35], а именно, если названный бит равен нулю, признак результата устанавливается равным 0, и если бит равен единице, признак результата устанавливается равным 1.

В результате выполнения операции в адресованный байт ОП заносится код FF (все единицы). Между моментами чтения байта и записью в него всех единиц другие обращения в эту ячейку недопустимы. Необходимо отметить, что запись всех единиц в выбранный байт производится самой ОП, а ЦП только сообщает, что выполняется команда ПРОВЕРИТЬ И УСТАНОВИТЬ, устанавливая в единицу триггер инструкции (команды) ПРОВЕРИТЬ И УСТАНОВИТЬ ТИПУ, состояние которого передается в управление ОП.

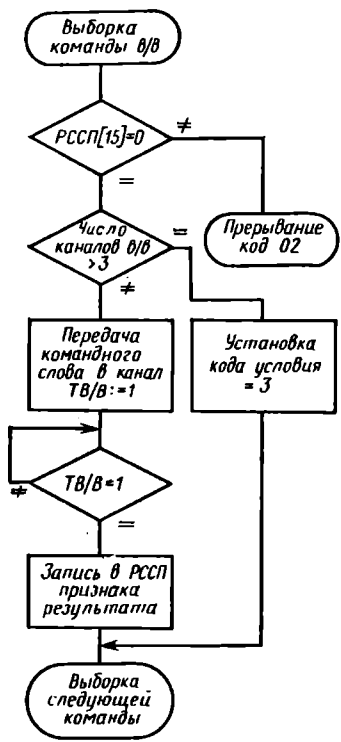


Рис. 73. Схема алгоритма выполнения команд ввода-вывода процессором

РБСП. После этого ЦП закидывается на одной микрокоманде, постоянно анализируя состояние ТВ/В, и ожидает, когда канал установит связь с выбранным устройством. По окончании связи с устройством канал помещает признак результата в РБСП, сбрасывает ТВ/В и отключается от ЦП. Информация из РБСП помещается процессором в зону признака результата РССП. Выполнение команды ввода-вывода процессором на этом заканчивается.

5.3. ВВОД-ВЫВОД

Для управления вводом-выводом в списке команд имеются четыре специальные команды (прил. 4).

Эти команды являются привилегированными, используются только тогда, когда ЦП находится в состоянии СУПЕРВИЗОР, и выполняются по единой микропрограмме. При выполнении этих команд основным в действиях процессора (рис. 73) является сообщение каналу о появлении команды ввода-вывода, для чего устанавливается в единичное состояние триггер ввода-вывода ТВ/В (см. табл. 18), и передача кода операции, номеров канала и устройства в общий канал через регистр буферный связи с процессором

ГЛАВА 6

КОМАНДЫ ОБРАБОТКИ ДАННЫХ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

С помощью этих команд выполняются арифметические и логические действия над двоичными операндами, которые могут быть либо адресами или индексами, либо представлять собой данные

с фиксированной запятой. Один операнд всегда находится в одном из РОН, другой может находиться или в РОН или в оперативной памяти.

Набор команд предусматривает возможность выполнения операций загрузки, сложения, вычитания, умножения, деления и записи в память, а также действий над знаком, преобразование чисел из одной системы счисления в другую и сдвиги операндов с фиксированной запятой. После выполнения операций над знаком, а также после операций сложения, вычитания, сравнения и сдвига устанавливается признак результата (см. прил. 9).

В общем случае все операнды с фиксированной запятой рассматриваются как целые числа со знаками. Обычно длина операнда составляет 32 разряда. Отрицательные числа представлены в дополнительном коде с единицей в знаковом разряде. Увеличение длины операнда в сторону старших разрядов достигается расширением поля целой части в сторону старших разрядов и присвоением каждому новому разряду значения, равного значению знакового разряда исходного числа (например, при расширении полуслова до полного слова, арифметические сдвиги).

6.1. ЗАГРУЗКА, СЛОЖЕНИЕ, СРАВНЕНИЕ, ЗАПИСЬ В ПАМЯТЬ, СДВИГ

Оборудование процессора в максимальной степени ориентировано на возможно более быструю обработку операндов фиксированного формата (см. п. 2.3). Большинство команд данного типа выполняется за один-два такта. В первом полутакте операнды через магистрали вызываются в БАЛ на обработку, во втором происходит запись результата. Одновременно с обработкой операндов в БАЛ вырабатывается признак результата (см. табл. 8), который во втором полутакте помещается в РССП. В этом же такте дается запрос на выборку следующей команды и производится анализ 38. Выполнение команд можно проиллюстрировать на примерах микропрограмм команд сложения (форматы RR и RX, рис. 74).

Команда СЛОЖЕНИЕ — AR (1A) формата RR выполняется за один такт. После выборки команды операнды из РОН считываются на M2 и M3. Номера РОН определяются соответственно содержимым счетчиков СТ1 РОН и СТ2 РОН ($M2 := \text{РОНСТ1}$, $M3 := \text{РОНСТ2}$). На этапе выборки команды (см. п. 4.4) в счетчики были записаны коды из полей R1 и R2 команды. Для БАЛ задается функция сложения ($F := A + B$). Результат операции во втором полутакте с M1 записывается на место первого операнда ($\text{РОНСТ1} := M1$) в РОН, номер которого определяется содержимым счетчика СТ1 РОН, т. е. полем R1 команды. В этом же полутакте по микроприказу РССП [34/35] := ПРА устанавливается признак результата операции (п. 2.4). В первом полутакте дается

запрос на выборку следующей микрокоманды ЗПРВ и производится анализ выборки.

Команда СЛОЖЕНИЕ — А (5А) формат RХ. Вначале ожидается конец обслуживания запроса к ОП за вторым операндом (ТОЧП $\neq 0$, см. пп, 1.6, 2.1). Этот запрос был сделан на этапе выборки команды. После того как операнд из ОП будет получен в РИП, он считывается на М2 (М2 := РОИСТ1). В остальном выполнение этой команды аналогично выполнению команды СЛОЖЕНИЕ формата RR.

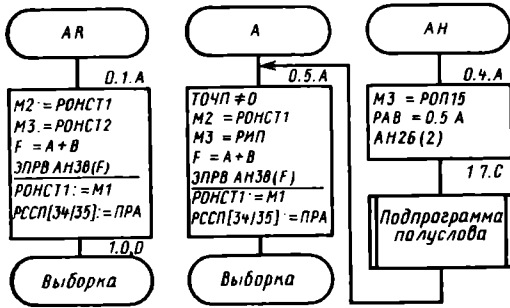
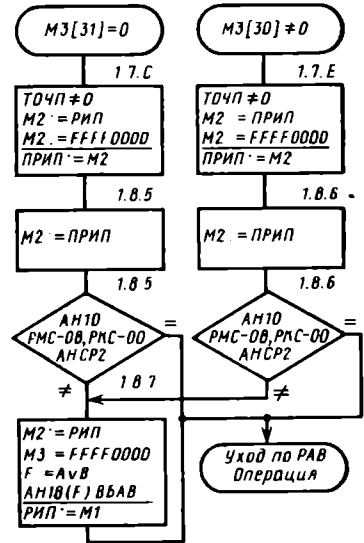


Рис. 74. Микропрограмма команд СЛОЖЕНИЕ—AR (1А), СЛОЖЕНИЕ—А(5А), СЛОЖЕНИЕ ПОЛУСЛОВА—АН (4А)

Рис. 75. Подпрограмма полуслова



Команда СЛОЖЕНИЕ ПОЛУСЛОВА — АН (4А), формат RХ. Слагаемый операнд — полуслово — может быть расположен в ОП, начиная с границы слова либо полуслова. Перед сложением он должен быть помещен в младшую половину слова слагаемого и расширен распространением знакового разряда в старшие разряды слова.

По первой микрокоманде 0.4.А в РАВ помещается адрес врата (РАВ := 0.5.А) для выхода на операцию сложения из подпрограммы полуслова и анализируется расположение операнда в слове ОП. Анализ выполняется путем определения значения предпоследнего разряда кода адреса операнда. Для этого код адреса считывается из РОП15, куда он был помещен на этапе выборки команды, на М3 (М3 := РОП15). Для определения значения М3 [30] используется анализ 26 с признаком ветвления ПРВ = 0010 (микроприказ АН26 (2)). Базовый адрес следующей микрокоманды 1.7.С. Далее происходит вход в одну из двух ветвей подпрограммы полуслова (рис. 75).

Если М3 [30] = 0, то операнд расположен начиная с границы слова, справа от него — посторонняя информация. Ожидается конец обслуживания запроса в ОП за операндом, который был сделан на этапе выборки команды. После этого операнд из РИП

считывается на $M2$. Одновременно на $M2$ накладывается маска ($M2 := FFFF\ 0000$) для освобождения от посторонней информации, расположенной справа от операнда. Во втором полутакте операнд с помощью перекрестной передачи с $M2$ (см. рис. 51) помещается в правую часть РИП, а в его левую часть помещаются нули. Далее необходимо проанализировать значение знакового разряда операнда. Для этой цели применяется анализ 10 вида АН СР2.

Операнд из РИП (см. рис. 75) считывается на $M2$ с помощью перекрестной передачи $M2 := ПРИП$ для того, чтобы знаковый разряд занял позицию $M2\ [0]$. Анализ происходит по следующей схеме:

| | | |
|---------------------------------------|--------------|--------------|
| | а) $M2[0]=0$ | б) $M2[0]=1$ |
| код сравнения (поле PKC) | 0000 0000, | 0000 0000, |
| маска сравнения (поле PMC) | 0000 1000, | 0000 1000, |
| код условия ($M3[0/3]$, $M2[0/3]$) | xxxx 0xxx, | xxxx 1xxx, |
| $MAN[0/3]$ | 1111; | 1010. |

где x — значение разряда не играет роли.

В случае *a* знак операнда положительный, поэтому обе тетрады кода условия равны соответствующим тетрадам кода сравнения. По микроприказу АН СР2 осуществляется переход по адресу возврата 0.5.A, т. е. непосредственно на выполнение операции сложения. Нули в левой части слова операнда означают распространение значения знакового разряда. В случае *б* знак операнда отрицательный, поэтому правые тетрады кодов условия и сравнения не равны и происходит переход по базовому адресу следующей микрокоманды 1.8.7. Так как знак операнда отрицательный, то необходимо в левую часть слова операнда записать единицы в качестве распространенного значения знакового разряда. Операнд считывается из РИП на $M2$ ($M2 := РИП$), на $M3$ накладывается маска ($M3 := FFFF\ 0000$), для БАЛ задается функция ИЛИ ($F := A \vee B$). Сформированное таким образом слово отрицательного операнда помещается в РИП ($РИП := M1$). По микроприказам АН18 (F) и ВБАВ (см. п. 3.5) происходит переход по адресу возврата 0.5.A на выполнение операции.

Если $M3\ [30] \neq 0$, то это означает, что операнд расположен начиная с границы полуслова, слева от него посторонняя информация. Перекрестной передачей $M2 := ПРИП$ операнд вместе с маской ($M2 := FFFF\ 0000$) считывается на $M2$, где освобождается от посторонней информации (на $M2$ она будет слева от операнда), после чего с $M2$ перекрестной передачей $ПРИП := M2$ операнд с $M2$ вновь помещается в правую половину РИП. Дальнейшие действия аналогичны рассмотренному выше.

Команды загрузки для операций с фиксированной и плавающей запятой выполняются одинаково. Микропрограммы этих команд описаны в п. 7.2, рис. 79.

При выполнении команд сдвига адрес второго операнда как адрес ОП не используется, а шесть его младших разрядов определяют величину сдвига. Для сдвига операнд предварительно записывается в РОП (рис. 76), а код величины сдвига — в РПС (п. 2.5). Причем в РПС помещаются лишь младшие пять бит кода величины сдвига, определяющие сдвиг не более чем на 31 разряд, а старший разряд анализируется с целью ветвления микропрограмм. При сдвигах больше, чем на 31 разряд при длине опе-

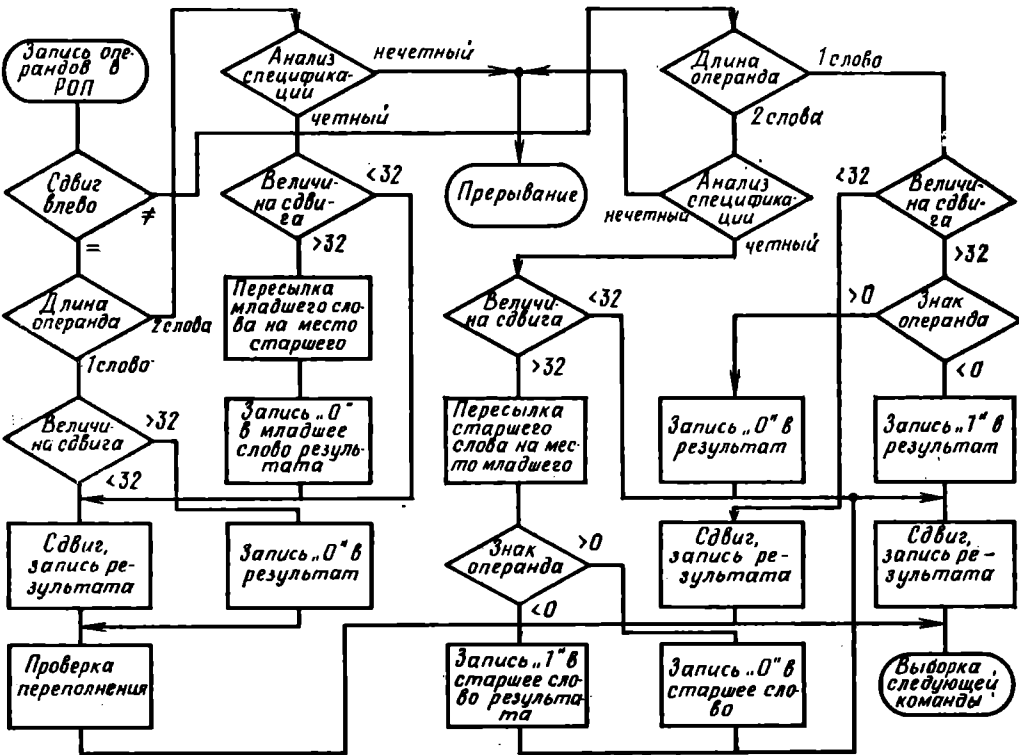


Рис. 76. Схема обобщенного алгоритма операций сдвига

ранда в одно слово, для сокращения времени выполнения команды собственно сдвига не происходит. В результат сразу записываются нули или единицы, в зависимости от направления сдвига и знака операнда. При длине операнда в два слова при сдвигах вправо старшее слово пересылается на место младшего и сдвигается по параметру сдвига. В старшее слово результата записываются нули или единицы, в зависимости от знака операнда. При сдвигах влево младшее слово пересылается на место старшего и сдвигается, а в младшее слово результата записываются нули. Для операндов двойной длины при сдвиге используются РСДВ1 и РСДВ2 (для коротких — РСДВ1) и, кроме того, производится анализ спецификации.

6.2. УМНОЖЕНИЕ

При выполнении команд этого типа в ЕС-1033 применяется способ одновременного умножения на два разряда множителя с учетом старшего разряда предыдущей пары разрядов [а. с. 651341 (СССР)]. Умножение выполняется, начиная с младших разрядов множителя. Обычно при умножении на 2 разряда, когда пара разрядов множителя, на которую производится очередное умножение, равна 00, к сумме частичных произведений прибавляется нулевое значение множимого (мнм) и новая сумма частичных произведений сдвигается на два разряда вправо. Для пары разрядов 10 прибавляется удвоенное значение множимого (2 мнм), для пары разрядов 11 — утроенное (3 мнм). Однако использование удвоенного и утроенного значений множимого усложняет алгоритм. Кроме того, применение этого метода в чистом виде требует преобразования отрицательных чисел, представленных в дополнительном коде, в прямой и обратного преобразования результата.

Эти недостатки устраняются при использовании следующего приема. Пусть C — некоторое частичное произведение; при сдвиге на два разряда вправо оно уменьшается в 4 раза. Если очередная пара разрядов множителя — 11, то новое частичное произведение

$$C' = \frac{C + 3 \text{ мнм}}{4}.$$

Преобразуем это соотношение следующим образом:

$$C' = \frac{C + 3 \text{ мнм}}{4} = \frac{C - 1 \text{ мнм}}{4} + 1 \text{ мнм}.$$

Распространение этого приема на другие комбинации пар разрядов множителя путем учета наличия единицы в старшем разряде пары прибавлением единичного значения множимого, которое можно сделать на следующем шаге, позволило реализовать в машине умножение без использования удвоенных и утроенных значений множимого. При одновременном умножении на пару разрядов множителя с учетом значения старшего разряда предыдущей пары частичные произведения на каждом шаге умножения формируются следующим образом:

1. Комбинация разрядов 00.0

$$C' = \frac{C + 0 \text{ мнм}}{4},$$

действия — сдвиг на два разряда вправо.

2. Комбинация разрядов 01.0

$$C' = \frac{C + 1 \text{ мнм}}{4},$$

действия — прибавление одинарного множимого, сдвиг на два разряда вправо.

3. Комбинация разрядов 10.0

$$C = \frac{C+2 \text{ мнм}}{4} = \frac{C-2 \text{ мнм}+4 \text{ мнм}}{4} = \frac{\frac{C}{2}-1 \text{ мнм}}{2} + 1 \text{ мнм},$$

действия — сдвиг на один разряд вправо, вычитание одинарного множимого, еще сдвиг на один разряд вправо; учет единицы старшего разряда пары разрядов множителя производится прибавлением одинарного множимого на следующем шаге.

4. Комбинация разрядов 11.0

$$C' = \frac{C-1 \text{ мнм}}{4} - 1 \text{ мнм},$$

действия — вычитание единичного множимого, сдвиг вправо на два разряда; единица старшего разряда пары разрядов множителя учитывается на следующем шаге.

5. Комбинация разрядов 00.1, действия — с учетом единицы старшего разряда предыдущей пары разрядов множителя, в текущем шаге формирование очередной суммы частичных произведений происходит как для комбинации 01.0.

6. Комбинация разрядов 01.1, действия — обрабатывается как 10.0, но поскольку в данной паре разрядов множителя нет единицы в старшем разряде, коррекции в следующем шаге также не будет:

$$C' = \frac{C+2 \text{ мнм}}{4} = \frac{\frac{C}{2}+1 \text{ мнм}}{2},$$

т. е. сдвиг на один разряд вправо, прибавление одинарного множимого, еще сдвиг на один разряд вправо.

7. Комбинация разрядов 10.1, действия — обрабатывается как 11.0.

8. Комбинация разрядов 11.1, действия — обрабатывается как 00.0.

Правила обработки пар разрядов множителя сведены в табл. 23.

Алгоритм позволяет умножать числа, которые, в зависимости от знака, могут быть представлены как в прямом, так и в дополнительном кодах. Результат автоматически получается в прямом коде, если он положительный, и в дополнительном коде, если он отрицательный. Никаких действий над знаками выполнять не требуется.

Количество циклов умножения равно количеству пар разрядов множителя, т. е. 16. Циклической частью микропрограммы выполняется 15 циклов умножения. Последний цикл умножения выполняется отдельно, так как он совмещен с записью результата. Параметр цикла помещается в СТЦ перед выходом микропрограммы на циклический участок (рис. 77).

**Обработка очередной пары разрядов множителя
в операции умножения с фиксированной запятой**

| Пара разрядов множителя | Старший разряд предыдущей пары | Действия |
|-------------------------|--------------------------------|---|
| 00 | 0 | Сдвиг на два разряда вправо |
| 01 | 0 | Прибавление, сдвиг на два разряда вправо |
| 10 | 0 | Сдвиг на один разряд вправо, вычитание, сдвиг на один разряд вправо |
| 11 | 0 | Вычитание, сдвиг на два разряда вправо |
| 00 | 1 | Прибавление, сдвиг на два разряда вправо |
| 01 | 1 | Сдвиг на один разряд вправо, прибавление, сдвиг на один разряд вправо |
| 10 | 1 | Вычитание, сдвиг на два разряда вправо |
| 11 | 1 | Сдвиг на два разряда вправо |

В цикле умножения действия согласно табл. 23 выполняются с помощью БАЛ, РСДВ и операционных регистров. Последние играют роль буферной памяти. Запись информации в РСДВ для сдвига, чтение сдвинутой информации из РСДВ и запись ее в регистры ЦП происходит через информационные магистрали, которые, для упрощения схемы, на рисунке не показаны.

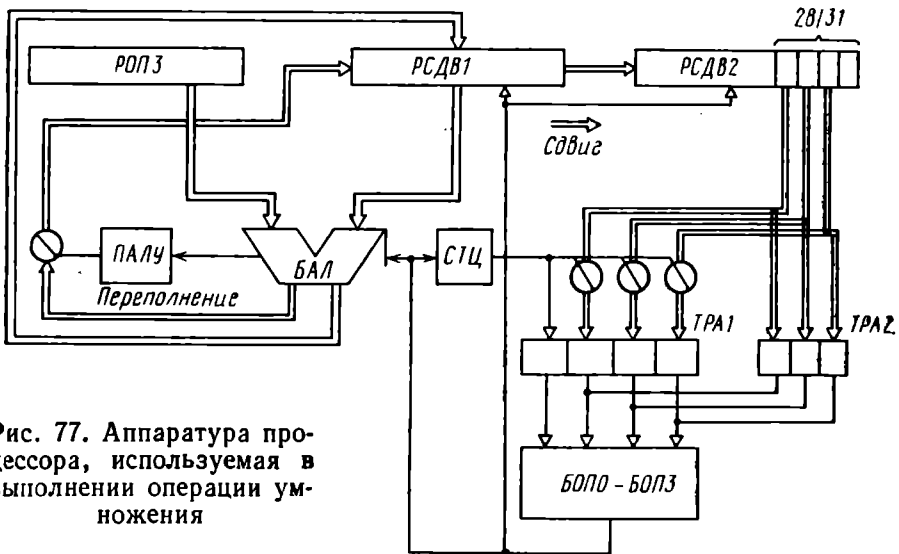


Рис. 77. Аппаратура процессора, используемая в выполнении операции умножения

Перед началом цикла умножения множимое засылается в операционный регистр РОПЗ, а множитель — в РСДВ2. В цикле умножения в РСДВ1 находится частичное произведение (первое частичное произведение — нули). Когда происходит сдвиг вправо информации в РСДВ, то биты частичного произведения, перемещаемые из РСДВ1 в РСДВ2, являются младшими битами окончательного результата. Таким образом, окончательный результат накапливается в РСДВ2, а биты множителя, которые при сдвиге вправо выходят за пределы разрядной сетки РСДВ2, теряются.

Схема переносов АЛУ (ПАЛУ) обеспечивает сохранение переносов при сложении (вычитании) и распространение значения этих переносов в левые разряды РСДВ1 при сдвигах вправо информации в РСДВ. Если результат сложения (вычитания) положительный, то в крайние левые разряды РСДВ1 вдвигаются нули, если отрицательный — единицы.

В зависимости от значения очередной пары разрядов и старшего разряда предыдущей пары должен быть сформирован адрес микрокоманды соответствующей ветви обработки. Значения этих разрядов для анализа помещаются в один из табличных регистров анализа ТРА1 или ТРА2 (см. табл. 18). Их содержимое анализируется по микроприказу АН1 или АН2 соответственно. Чтобы сократить время выполнения операции, процессы текущего цикла умножения и запись в ТРА значений разрядов множителя для очередного цикла совмещены. Именно с этой целью применяются два регистра ТРА. При подготовке цикла умножения происходит запись в ТРА1 и формирование адреса БОП, соответствующего началу цикла. В первом цикле по сформированному адресу выполняются действия цикла умножения и одновременно происходит запись в ТРА2 значений очередных разрядов множителя для следующего цикла и формирование соответствующего ему адреса БОП. Во втором цикле по сформированному адресу выполняются действия цикла умножения и происходит новая запись в ТРА1 и т. д. Значение счетчика СТЦ анализируется при анализе ТРА1, а его содержимое уменьшается на единицу при анализе ТРА2. Таким образом, по одному значению СТЦ выполняются два цикла умножения. Константа, помещаемая в СТЦ перед началом умножения, равна 07. При нулевом значении СТЦ блокируется очередная запись в ТРА1 и осуществляется выход из цикла. Поскольку в текущем цикле в ТРА помещаются значения разрядов множителя, анализ которых необходим для выхода на следующий цикл, то эти разряды множителя перед выполнением текущего цикла оказываются расположенными в РСДВ2 [28/30]. В РСДВ2 [28/29] находится очередная пара разрядов множителя, на которую будет производиться умножение в следующем цикле, а в РСДВ2 [30/31] — пара разрядов множителя, на которую производится умножение в текущем цикле.

Пример. Два следующих примера иллюстрируют процесс умножения. Для простоты и наглядности в этих примерах формат операндов и регистров (РОП, РСДВ1, РСДВ2) сокращен до четырех разрядов. Смысл операции и действий по табл. 23 при этом остается прежним, в ТРА по-прежнему записываются значения трех предпоследних разрядов множителя из РСДВ2.

Перед началом цикла умножения первое частичное произведение (нули) и множитель помещаются в РСДВ1 и РСДВ2. Для подготовки цикла умножения множитель сдвигается влево на два разряда, справа в РСДВ2 вдвигаются нули. Таким образом, первая пара разрядов множителя, на которые будет производиться умножение в первом цикле, и старший разряд предыдущей пары (в первом цикле он равен нулю) оказываются в положении, подготовленном для записи в ТРА1. После анализа содержимого ТРА1 и формирования адреса БОП для первого цикла множитель сдвигается вправо на два разряда и

занимает в РСДВ прежнее положение. Результат цикла умножения (очередное частичное произведение) с выхода БАЛ через информационную магистраль записывается в РСДВ1. Отрицательный результат получается в дополнительном коде.

Пример. $5 \times 6 = 30$.

В машине операнды будут иметь следующий вид:

$$0000\ 0005 \times 0000\ 0006 = 0000\ 0000\ 0000\ 001E$$

| | | РСДВ1 | РСДВ2 | | |
|---|---|---------|--------------------|---|----------------------|
| 1-е частичное произведение | | 0 0 0 0 | 0 1 1 0 | Множитель | |
| | | 0 0 0 1 | <u>1 1 0 0 0</u> 0 | Сдвиг на два разряда влево (подготовка цикла умножения) | |
| | | 0 0 0 0 | <u>0 0 1 1 1</u> 0 | → Запись в ТРА1 (для 1-го цикла) | |
| | | | | Сдвиг на два разряда вправо | |
| | | | | → Запись в ТРА2 (для 2-го цикла) | |
| Множимое (дополнительный код) | + | 0 0 0 0 | 0 0 1 1 | Сдвиг на один разряд вправо Вычитание | 1-й цикл анализ ТРА1 |
| | | 1 0 1 1 | | | |
| Результат < 0 2-е частичное произведение | | 1 0 1 1 | | Сдвиг на один разряд вправо | |
| | | 1 1 0 1 | 1 0 0 1 | | |
| Результат > 0 Произведение (окончательный результат) | + | 1 1 1 0 | 1 1 0 0 | Сдвиг на один разряд вправо Прибавление | 2-й цикл анализ ТРА2 |
| | | 0 1 0 1 | | | |
| | | 0 0 1 1 | | Сдвиг на один разряд вправо | |
| | | 0 0 0 1 | 1 1 1 0 | | |

Пример. $5 \times (-4) = -20$.

В машине операнды будут иметь следующий вид:

$$0000\ 0005 \times 1111\ 111C = 1111\ 1111\ 1111\ 11EC$$

| | | РСДВ1 | РСДВ2 | | |
|---|---|---------|------------------|---|----------------------|
| 1-е частичное произведение | | 0 0 0 0 | 1 1 0 0 | Множитель (дополнительный код) | |
| | | 0 0 1 1 | <u>0 0 0 0</u> 0 | Сдвиг на два разряда влево (подготовка цикла) | |
| | | 0 0 0 0 | <u>1 1 1 0</u> 0 | → Запись в ТРА1 | |
| | | | | Сдвиг на два разряда вправо | |
| | | | | → Запись в ТРА2 | |
| 2-е частичное произведение | | 0 0 0 0 | 0 0 1 1 | Сдвиг на два разряда вправо | 1-й цикл анализ ТРА1 |
| | | 0 0 0 0 | | | |
| Результат > 0 | + | 1 0 1 1 | | Вычитание | 2-й цикл анализ ТРА2 |
| | | 1 0 1 1 | | | |
| Результат < 0 Произведение (окончательный результат) в дополнительном коде | | 1 0 1 1 | | Сдвиг на два разряда вправо | |
| | | 1 1 1 0 | 1 1 0 0 | | |

Результат в прямом коде: 0001 0100.

При перемножении 32-разрядных чисел с помощью 32-разрядного БАЛ истинный промежуточный результат может иметь 33 разряда. В этом случае его старший (знаковый) разряд поступает в триггер ПАЛУ и учитывается при сдвигах. Эта ситуация представляет собой переполнение сумматора. Единица переполнения управляет формированием нулей или единиц, вдвигаемых слева в старшую часть РСДВ.

6.3. ДЕЛЕНИЕ

Наличие остатка при выполнении команд деления операндов с фиксированной запятой значительно ограничивает возможности применения логических методов ускорения деления. В машине деление выполняется методом без восстановления остатка. При этом каждый разряд частного представляет собой инверсию знака остатка в данном цикле. Если остаток положительный, то в следующем цикле производится сдвиг частичного остатка влево на

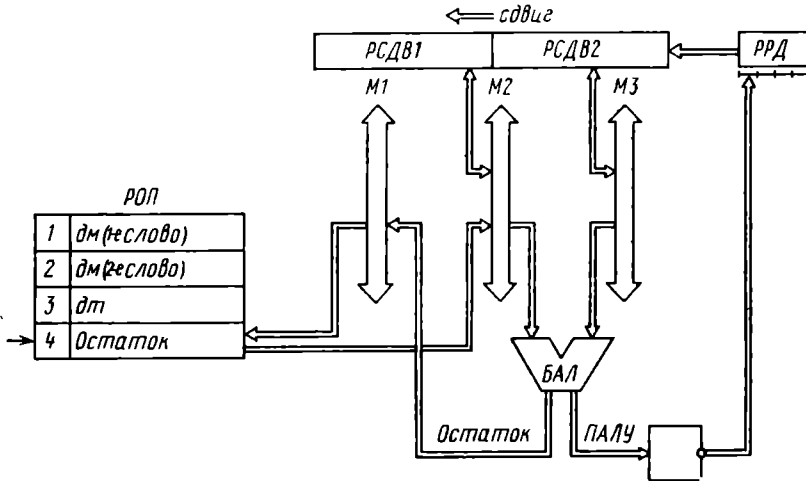


Рис. 78. Аппаратура процессора, используемая в выполнении операции деления

один разряд и вычитание из него делителя, если остаток отрицательный, то после сдвига производится прибавление делителя. Деление выполняется над числами, представленными в прямом коде.

В начале операции делимое (дм) помещается в РОП1 и РОП2 (соответственно старшая и младшая половины 64-разрядного слова делимого, рис. 78). В первом цикле деления оба слова делимого записываются в РСДВ и сдвигаются на один разряд влево. После этого из делимого вычитается делитель. Знак остатка анализируется аппаратно и, если остаток положительный, в регистр РРД заносится единица ($РРД := 1$), если отрицательный — нуль ($РРД := 0$). Таким образом, в РРД формируется значение разряда результата. Остаток вычитания с магистрали М1 записывается в РОП4. В следующем цикле остаток из РОП4 помещается

в РСДВ1, и информация, находящаяся в РСДВ1 и РСДВ2, сдвигается на один разряд влево. Справа в РСДВ1 из РСДВ2 вдвигается очередной разряд делимого, а в РСДВ2 из РРД очередной разряд частного, где таким образом накапливается результат деления. Затем снова происходит вычитание делителя и анализируется знак остатка. Значение очередного разряда частного, которое формируется в РРД, равно инверсному значению знака остатка. Остаток формируется в РСДВ1. Количество циклов деления равно 32. В конце операции результат (остаток и частное) записывается по адресу первого операнда (делимого).

Приведенный ниже пример иллюстрирует процесс деления. Для простоты форматы регистров (РОП, РСДВ1, РСДВ2) и делителя сокращены до четырех, а делимого — до восьми разрядов.

Пример. $37 : 4 = (1) 9$.

В машине операнды будут иметь следующий вид:

0000 0000 0000 0025 : 0000 0004 = 0000 0001 0000 0009

| Знак | РСДВ1 | РСДВ2 | РРД |
|------|-------|-------|-----|
| 0 | 0010 | 0101 | 0 |
| 0 | 0100 | 1010 | 0 |
| 1 | 1100 | | |
| 0 | 0000 | 1010 | 1 |
| 0 | 0001 | 0101 | |
| 1 | 1100 | | |
| 1 | 1101 | 0101 | 0 |
| 1 | 1010 | 1010 | |
| 0 | 0100 | | |
| 1 | 1110 | 1010 | 0 |
| 1 | 1101 | 0100 | |
| 0 | 0100 | | |
| 0 | 0001 | 0100 | 1 |
| 0 | 0010 | 1001 | |

↓
 → 0001 1001
 остаток частное
 из из
 РОП4 РСДВ2

Делимое
 Сдвиг на один разряд влево
 Вычитание
 Остаток > 0, РРД := 1
 Сдвиг на один разряд влево
 Вычитание
 Остаток < 0, РРД := 0
 Сдвиг на один разряд влево
 Прибавление
 Остаток < 0, РРД := 0
 Сдвиг на один разряд влево
 Прибавление
 Остаток > 0, РРД := 1, конец цикла
 Сдвиг на один разряд влево, запись результата

Результат

Делимое размещается в двух смежных РОН: с четным номером, указанным полем *R1* команды, и с нечетным номером, следующим за ним. Остаток должен быть помещен по адресу первого операнда соответственно РОН с четным номером, а частное в РОН — с нечетным номером. В любом случае результат должен занимать не более двух РОН. Перед началом цикла деления микропрограммно проверяется, не будет ли результат занимать больше, чем два регистра.

Такую возможность проиллюстрируем на примере деления десятичных чисел, если предположить, что одна цифра занимает один условный регистр:

$37 : 5 = (2)7$ — результат занимает два регистра.

$37 : 2 = (1) 18$ — результат занимает три регистра.

Последний случай выявляется пробным вычитанием делителя из старшего слова делимого, и в случае положительного результата событие классифицируется как некорректность деления с фиксированной запятой. Происходит переход по адресу микрокоманды 0.0.C. (см. п. 4.3) для программного прерывания.

КОМАНДЫ ОБРАБОТКИ ДАННЫХ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

7.1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

По командам с плавающей запятой выполняются операции загрузки, записи в ОП, операции над знаками операндов и арифметические операции (прил. 4).

В общем случае любое число A может быть представлено в виде

$$A = 16^p \cdot M,$$

где p — порядок числа, равный степени, в которую возводится основание системы счисления; M — мантисса числа.

Форматы данных с плавающей запятой приведены в прил. 6. Порядок представлен в двоичном коде с избытком 64 и называется характеристикой, которая может принимать значения от 0 до 127, что соответствует изменению порядка от -64 до $+63$. Таким образом, характеристика числа, в отличие от порядка, всегда является положительной величиной, что облегчает действия над порядками в арифметических операциях.

В командах с плавающей запятой регистры общего назначения используются только для формата RX , где поле $X2$ определяет номер РОН, являющегося регистром индекса, а поле $B2$ — номер РОН, содержащего базовый адрес. Для хранения операндов регистры общего назначения не используются. Вместо них используются четыре 64-разрядных регистра РПЗ. Поскольку магистрали и БАЛ оперируют 32-разрядными словами, в микропрограммах операций предусмотрены считывание и обработка отдельно старших и младших частей РПЗ. Адреса РПЗ, так же как и РОН, определяются содержимым счетчиков $ST1$ РОН и $ST2$ РОН. Тип регистров, РОН или РПЗ, определяет состояние триггера плавающей запятой (ТПЗ), при единичном состоянии которого используются РПЗ, при нулевом — РОН. Этот триггер устанавливается в 1 во всех операциях с плавающей запятой по окончании выборки команды по микроприказу $РАБОП := РДШК$ или $РАБОП := РХК$ (см. п. 3.6). В конце операции ТПЗ сбрасывается по микроприказу $АН38$.

При выполнении операций над знаками чисел, операций сложения, вычитания и сравнения устанавливается признак результата, который может принимать значения 0, 1 или 2 (прил. 10). При выполнении остальных операций с плавающей запятой признак результата не изменяется. Признак 3 в операциях с плавающей запятой не устанавливается. Механизм формирования признаков результата описан в п. 2.4.

Для получения максимальной точности арифметические операции могут выполняться с нормализацией результатов. Нормализация заключается в сдвигах мантииссы влево до тех пор, пока старшая шестнадцатиричная цифра не будет отлична от нуля. При этом характеристика должна быть уменьшена на величину, равную количеству сдвигов мантииссы на четыре разряда влево.

Величину сдвига мантииссы определяет параметр нормализации (ПНЛ), который численно равен количеству нулевых тетрад слева от старшей значащей шестнадцатиричной цифры. Этот параметр формируется специальной схемой (п. 2.5), которая подключена к магистрали М2 и определяет равенство нулю содержимого каждой тетрады магистрали. Для его определения мантиисса операнда считывается на магистраль М2, а схема формирует ПНЛ для разрядов [8/31] и [0/31] операнда. По микроприказу $M1 := \text{ПНЛ [8/31]}$ для коротких операндов или для старшего слова длинных операндов ПНЛ считывается на $M1[5/7]$. По микроприказу $M1 := \text{ПНЛ [0/31]}$ на $M1[5/7]$ считывается ПНЛ для младших слов мантиисс длинных операндов. Затем ПНЛ по микроприказу $\text{РПС} := M1[5/7]$ с $M1[5/7]$ заносится в РПС блока сдвиговых регистров. Мантиисса, подлежащая сдвигу, с магистрали М2 передается в сдвигатель, после чего происходит сдвиг по параметру, код которого находится в РПС.

В арифметических операциях с плавающей запятой используется ряд специфических анализов, которые выполняются специальными схемами, входящими в состав БФА БОП. К числу таких анализов относятся анализы 28, 29, 31, 36, 37 (см. табл. 17). При проведении каждого из них за один машинный такт выполняется комплекс проверок, результаты которых выдаются на МАН в этом же такте. Это позволило в весьма значительной степени, по сравнению с традиционным подходом, сократить время выполнения операций.

Хотя конечные результаты имеют мантииссу из 6 шестнадцатиричных цифр для операций с обычной точностью и из 14 цифр для операций с повышенной точностью, промежуточные результаты в арифметических операциях могут иметь еще одну учитываемую младшую цифру. Эта дополнительная цифра служит для увеличения точности результата.

7.2. ЗАГРУЗКА, ПОПОЛАМ, СЛОЖЕНИЕ, ВЫЧИТАНИЕ, СРАВНЕНИЕ

По командам типа ЗАГРУЗКА (рис. 79) второй операнд считывается на магистраль М3 либо из РИП (формат RX), либо из РПЗ (формат RR) и передается в БАЛ, а результат с его выхода через магистраль М1 помещается на место первого операнда. В этой же микрокоманде дается запрос на выборку следующей команды и осуществляется анализ выборки. При обработке длинных операндов вначале происходит загрузка старшего слова, за-

тем содержимое счетчиков СТ1 РОН и СТ2 РОН увеличивается на 1 для задания адресов РПЗ младших слов операндов (для формата RR) и производится загрузка младшего слова. В команде формата RX увеличивается на 1 содержимое СТ1 РОН, задавая тем самым номер РПЗ для помещения младшего слова, а также по сигналу

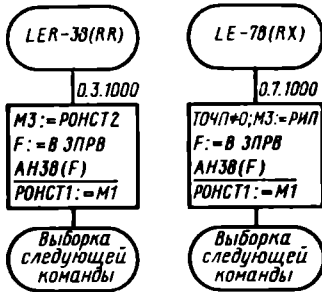
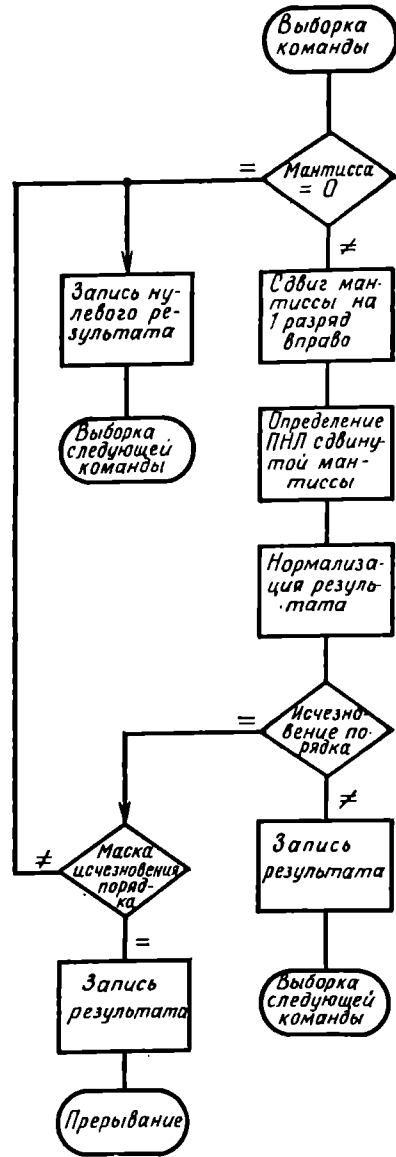


Рис. 79. Микропрограммы команд загрузки LER (38) и LE (78)

Рис. 80. Схема микропрограммы команды ПОПОЛАМ для коротких операндов



РАП := +4 увеличивается на 4 адрес второго операнда, по которому из ОП считывается и помещается в РПЗ младшее слово.

Команды типа ЗАГРУЗКА И ПРОВЕРКА и команды, осуществляющие операции со знаками, выполняются так же, но дополнительно устанавливается признак результата.

При выполнении команды ПОПОЛАМ для короткого операнда (рис. 80) мантисса сдвигается вправо на 1 двоичный разряд, что равносильно делению числа на 2. Выполнение этой команды для длинного операнда отличается тем, что во избежание потери дополнительной пятнадцатой цифры промежуточного результата мантисса операнда сдвигается на одну шестнадцатиричную цифру влево, т. е. на 4 разряда. Но так как сама операция заключается в сдвиге мантиссы вправо на 1 разряд, то в результате мантисса сдвигается влево только на 3 разряда. В этом случае при выполнении нормализации ветвление микропрограммы осуществляется в соответствии с табл. 24 (см. ниже), но характеристика после

корректировки вычитанием кода ПНЛ уменьшается еще на единицу. Этим дополнительным вычитанием учитывается предварительный сдвиг мантиссы на одну шестнадцатиричную цифру влево.

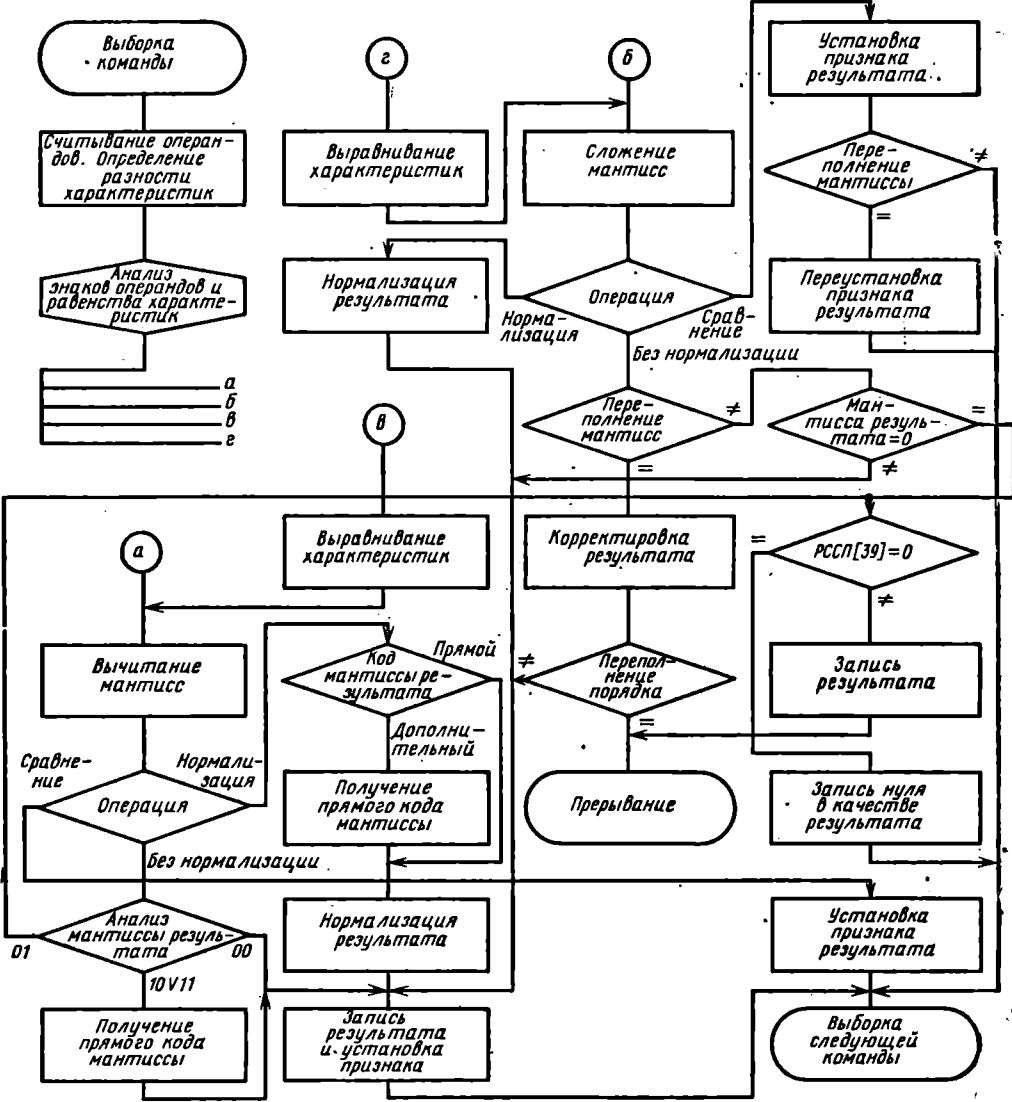


Рис. 81. Схема микропрограмм команд сложения, вычитания и сравнения для коротких операндов

Команды типа СЛОЖЕНИЕ, ВЫЧИТАНИЕ, СРАВНЕНИЕ для коротких форматов выполняются по единой микропрограмме (рис. 81) в четыре этапа: анализ знаков и характеристик операндов; выравнивание характеристик; сложение (вычитание) мантисс; нормализация результата, если это необходимо.

С целью сокращения времени выполнения команды для определения знаков и выявления особенностей характеристик операндов

дов применяется специальный анализ знаков и характеристик операндов — анализ 28 (см. табл. 17). При его проведении по микроприказу АН28 (Е) операнды считываются на магистрали М2 и М3 и определяются равенство характеристик операндов (МАН [0]), неравенство знаков операндов (МАН [1]) и значение знака второго операнда (МАН [2]). По результату анализа происходит ветвление микропрограммы на восемь направлений четырех типов:

- а) характеристики равны, знаки разные;
- б) характеристики равны, знаки одинаковые;
- в) характеристики не равны, знаки разные;
- г) характеристики не равны, знаки одинаковые.

Для выравнивания характеристик операндов характеристика одного вычитается из характеристики другого. Их разность, которая представляет собой разность порядков чисел и служит параметром сдвига при выравнивании, помещается в РПС. Мантисса числа, имеющего меньший порядок, сдвигается вправо в соответствии с величиной параметра. Анализ «Операция» определяет по коду операции с помощью анализа 17 ее характер: с нормализацией, без нормализации, сравнение.

Для выявления особенностей мантиссы результата применяется специальный анализ 33 (З) (см. табл. 17), который определяет код, в котором получился результат (МАН [2]), и значение мантиссы результата (МАН [3]). Для проведения анализа результат считывается на магистраль М2. Микропрограмма разветвляется на три направления в зависимости от кода на МАН [2/3]:

- 00 — результат в прямом коде, мантисса ненулевая;
- 10 или 11 — результат в дополнительном коде;
- 01 — результат в прямом коде, мантисса нулевая.

Для ускорения нормализации в операциях сложения и вычитания используется специальный анализ 29 (см. табл. 17), определяющий направление ветвления микропрограммы в соответствии с табл. 24.

Т а б л и ц а 24

Анализ нормализации в операциях сложения и вычитания
(анализ 29)

| Нормализация | Проверяемые условия |
|--|---|
| Нет нормализации На одну цифру На 2—6 цифр На 7 цифр На 8 цифр и более Нулевая мантисса Переполнение | $M2[8/11] \neq 0$ $M2[8/11] = 0 \wedge M2[12/15] \neq 0$ $M2[8/31] = 0 \wedge M3[0/3] = 0$ $M2[8/31] = 0 \wedge M3[0/7] = 0$ $M2[8/31] = 0 \wedge M3 = 0$ $M2[4/7] \neq 0$ |

Для выполнения этого анализа мантисса операнда помещается на магистрали М2 (старшая часть мантиссы) и М3 (младшая часть мантиссы). В зависимости от проверяемых условий происходит модификация адреса следующей микрокоманды для выполнения одного из действий нормализации:

Нет нормализации. Результат получился нормализованным. Происходят запись результата и, если необходимо, установка признака результата.

Нормализация на одну цифру. Мантисса сдвигается на четыре разряда влево, а характеристика уменьшается на единицу. Сдвиг задается микроприказами СДВЛЕВ, СДВ4 (сдвиг влево на 4 разряда). Нормализация на 2—6 цифр. Сдвиг происходит в пределах одного слова. Вычисляется ПНЛ и помещается в РПС. Сдвигатель осуществляет сдвиг в соответствии с параметром сдвига. Из характеристики вычитается величина ПНЛ.

Нормализация на 7 цифр. Здесь $M2 [8/31] = 0$, $M3 [0/3] = 0$ и сдвиг должен быть осуществлен на 28 разрядов, т. е. в пределах двух слов. Для ускорения нормализации младшая часть мантиссы микропрограммно помещается на место старшей части в разряды $[4/31]$, что эквивалентно сдвигу влево на 32 разряда, а затем дополнительно сдвигается по микроприказам СДВПР, СДВ4 вправо на 4 разряда. Таким образом, результирующий сдвиг мантиссы равен сдвигу на 28 разрядов влево. Из характеристики вычитается число 7.

Нормализация на 8 цифр и более. Сдвиг в пределах двух слов. Для ускорения нормализации младшая часть мантиссы микропрограммно помещается на место старшей, что эквивалентно сдвигу влево на 32 разряда, и из характеристики вычитается число 8. Затем определяется ПНЛ оставшейся ненормализованной младшей части мантиссы, который помещается в РПС, и производится дополнительный сдвиг мантиссы в соответствии с этим параметром. Из характеристики вычитается величина дополнительного ПНЛ.

Нулевая мантисса. В поле результата заносятся нули. Тем самым формируется нулевой результат.

Переполнение. Первая значащая цифра мантиссы занимает разряды $[4/7]$. Мантисса сдвигается вправо на 4 разряда, а характеристика увеличивается на единицу.

Выполнение команд типа СЛОЖЕНИЕ, ВЫЧИТАНИЕ, СРАВНЕНИЕ, в которых участвуют операнды длинного формата, отличается от выполнения этих же команд с числами короткого формата тем, что перед выравниванием порядков мантиссы обоих операндов сдвигаются влево на одну цифру. Это делается для того, чтобы избежать потери дополнительной цифры мантиссы. В этом случае при проведении нормализации учитывается предварительный сдвиг. Если же порядки операндов равны и в выравнивании их нет необходимости, нормализация результата осуществляется обычным путем в соответствии с табл. 24.

7.3. УМНОЖЕНИЕ

По командам типа УМНОЖЕНИЕ порядки сомножителей складываются, а мантиссы перемножаются. Знак произведения определяется по правилам алгебры.

Особенностью выполнения операции является то, что предварительной нормализации подвергается только множимое. Для множителя определяется ПНЛ, величина которого учитывается при

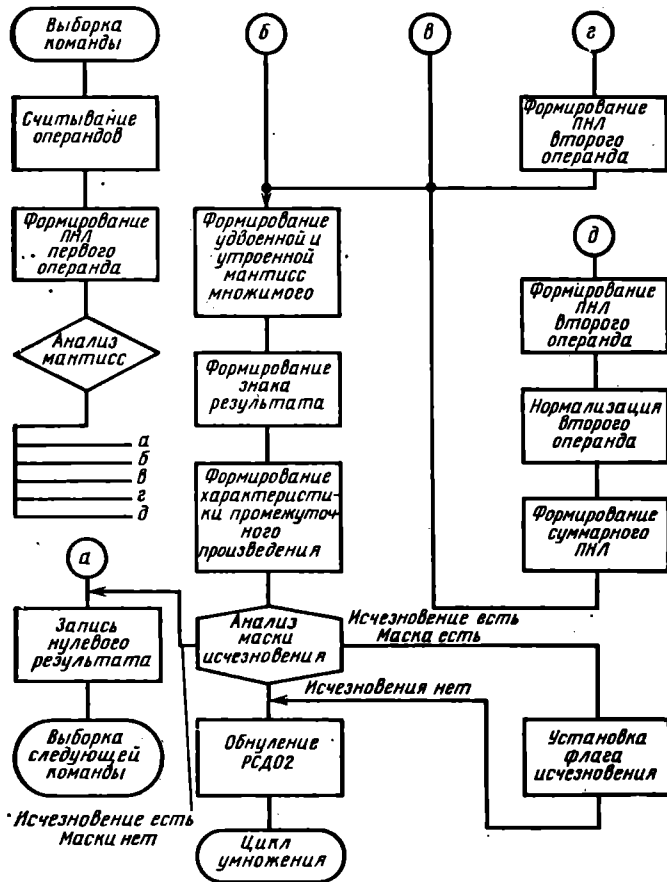


Рис. 82. Схема микропрограммы операции умножения

коррекции характеристики результата, а сдвиг мантиссы множителя не производится. Умножение на ненормализованное число позволяет ускорить выполнение операции. Операции над знаками, характеристиками и мантиссами выполняются отдельно, а результат формируется непосредственно перед его записью в память.

После считывания операндов формируется ПНЛ первого из них (рис. 82), после чего производится анализ мантисс. Это делается с помощью специального анализа нормализации в операциях умножения — деления (анализ 31). Он выполняется за один такт работы процессора, что по сравнению с традиционными способами значительно сокращает время выполнения операции. По резуль-

тату анализа микропрограмма разветвляется на следующие направления:

а) мантисса хотя бы одного операнда равна нулю. Происходит запись нулей в качестве результата, и операция на этом заканчивается;

б) оба операнда нормализованы. Формируются и размещаются в РОП удвоенное и утроенное значения мантиссы множимого для использования их в цикле умножения. Формируются знак и характеристика промежуточного произведения. Последняя формируется следующим образом: характеристики операндов складываются и из суммы вычитается число 64. Таким образом, характеристика промежуточного произведения, так же как и характеристики исходных операндов, представлена в двоичном коде с избытком 64. Затем происходит анализ исчезновения порядка и маски исчезновения РССП [38] (прил. 8). Исчезновение порядка имеет место, если значение характеристики произведения меньше нуля. Если маска исчезновения равна единице ($РССП [38] = 1$), то в конце операции происходит программное прерывание. В этом случае записывается результат с характеристикой, на 128 больше действительной. Если $РССП [38] = 0$, то результат должен быть истинным нулем. В случае если имеют место исчезновение и нулевая маска, то происходит запись нулевого результата и операция на этом заканчивается. При наличии же исчезновения, но при отсутствии маски устанавливается признак (флаг) исчезновения и происходит переход к выполнению операции, поскольку в этом случае мантисса результата должна быть определена;

в) первый операнд не нормализован. Поскольку предварительной нормализации подвергается только множимое (см. выше), то в этом качестве берется нормализованный операнд, т. е. второй операнд, а множителем — первый. Действия те же, что и в направлении разветвления б;

г) второй операнд не нормализован. Он принимается в качестве множителя, а первый — в качестве множимого. Формируется ПНЛ ненормализованного (т. е. второго) операнда и далее проводятся те же действия, что и в направлении разветвления б:

д) оба операнда не нормализованы. Один из них — множимое — должен быть нормализован. В качестве множимого берется второй операнд, множителя — первый. Производится нормализация второго операнда и далее происходит переход на направление разветвления б.

Перемножение мантисс производится в цикле умножения и заключается в поочередном умножении мантиссы множимого на два разряда мантиссы множителя, начиная с младших. Перед началом циклов умножения единичное, удвоенное и утроенное значения множимого хранятся в операционных регистрах, в РСДВ1 и РСДВ2 помещается множитель (рис. 83), а в РСДО1 и РСДО2 — промежуточное произведение (первое промежуточное произведение — нули). В циклах умножения при последователь-

ных сдвигах вправо множитель выходит за пределы РСДВ, а промежуточное произведение в РСДО замещается результатом. Поскольку при умножении РСДО и РСДВ работают совместно и синхронно, осуществляя сдвиг в пределах 128 разрядов (п. 2.5), то благодаря переходу информации из РСДО в РСДВ в конце циклов в РСДВ [0/3] будет находиться 15-я цифра результата. В РСДО [0] хранится знак произведения благодаря особенностям арифметического сдвига в РСДО.

Серия циклов умножения (рис. 84) начинается с анализа 37, который осуществляется за один такт. Для его выполнения множитель из РСДВ считывается на магистрали М2 и М3 и их содержимое анализируется на нуль ($M2 [8/31] \wedge M3 \neq 0$). Кроме того,

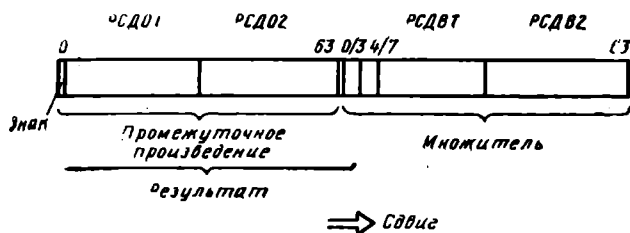


Рис. 83. Расположение операндов в регистрах сдвигателя

проверяется на нуль значение двух последних разрядов множителя, которые находятся в разрядах М3 [30/31] ($M3 [30/31] = 00$) и на которые происходит очередное умножение, а также состояние триггера счета ТСЧ (табл. 25).

Т а б л и ц а 25

Анализ цикла умножения (анализ 37)

| Направление | Проверяемые условия |
|-------------|---|
| а | $M3[30/31]=00 \wedge (M2[8/31] \wedge M3 \neq 0) \wedge ТСЧ \neq 0$ |
| б | $M3[30/31]=01 \wedge (M2[8/31] \wedge M3 \neq 0) \wedge ТСЧ \neq 0$ |
| в | $M3[30/31]=10 \wedge (M2[8/31] \wedge M3 \neq 0) \wedge ТСЧ \neq 0$ |
| г | $M3[30/31]=11 \wedge (M2[8/31] \wedge M3 \neq 0) \wedge ТСЧ \neq 0$ |
| д | $M2[8/31] \wedge M3 = 0 \wedge ТСЧ = 0$ |

Триггер ТСЧ предназначен для подсчета количества сдвигов вправо на 2 разряда. На этапе выборки команды этот триггер устанавливается в нулевое состояние сигналом АН38. По каждому микроприказу СДВ2 триггер ТСЧ поочередно изменяет свое состояние. Окончание перемножения мантисс (выход из циклов умножения, направление «д») определяется тем, что множитель выдвигается за пределы сдвигателя, где он находится во время умножения, и содержимое магистралей М2 и М3 становится равным нулю ($M2 [8/31] \wedge M3 = 0$). Кроме того, для выхода из циклов умножения нужно, чтобы ТСЧ находился в нулевом состоянии, иначе выход из циклов по условию нулевого состояния

магистралей M2 и M3 может произойти раньше действительного окончания умножения (случай, когда два старших разряда мантиссы множителя равны нулю, т. е. при наличии цифр 0011, 0010, 0001) и мантисса результата окажется недосдвинутой на 2 разряда.

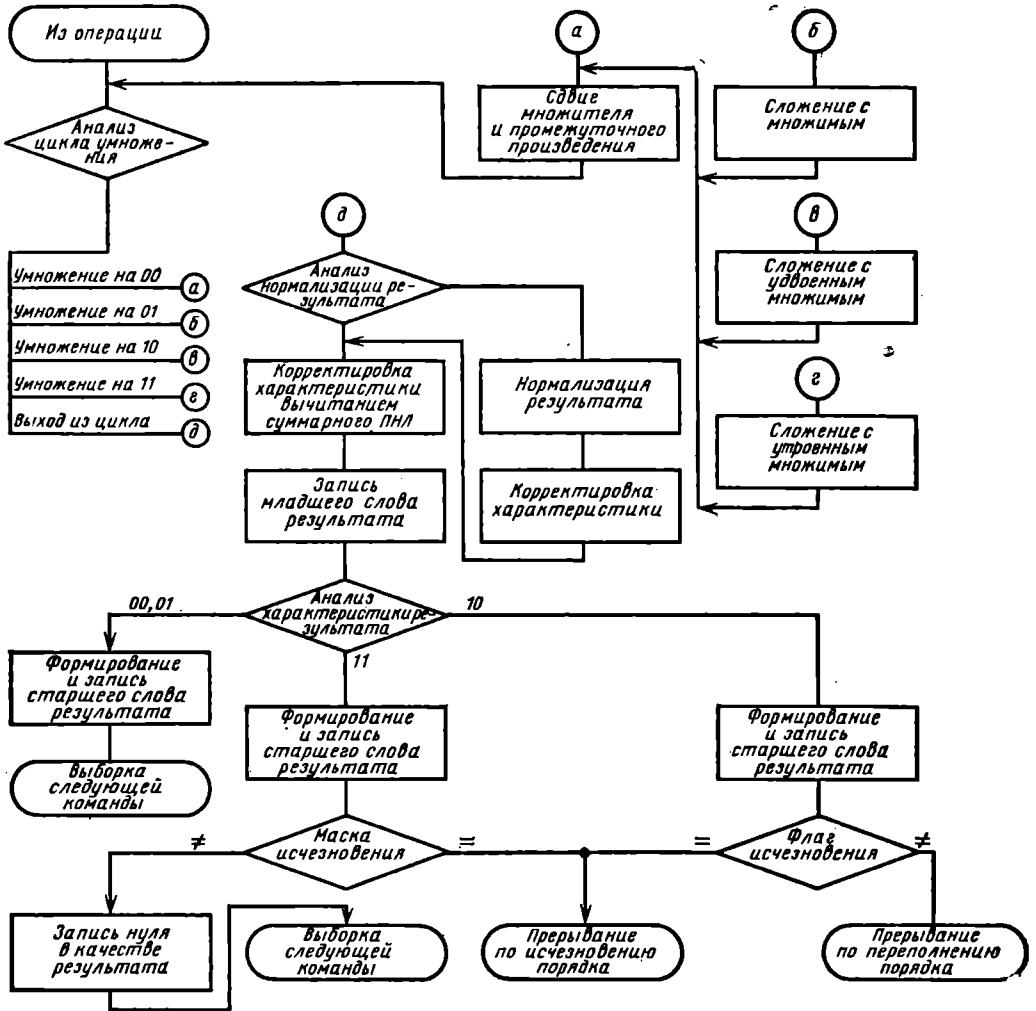


Рис. 84. Схема микропрограммы цикла умножения и окончания операции

При умножении на 01 (направление б) к промежуточному произведению прибавляется множимое, при умножении на 10 (направление в) — удвоенное множимое, при умножении на 11 (направление г) — утроенное. Сложение выполняется с помощью БАЛ, а его результат (промежуточное произведение) с магистралаи M1 записывается в РСДО1 (пп. 2.4, 2.5). При каждом умножении на 2 очередных разряда множителя промежуточное произведение и множитель сдвигаются на 2 разряда вправо. При умножении на 00 (направление а) к промежуточному произведению не прибавляется ничего, осуществляется только сдвиг.

После выхода из цикла умножения, если это необходимо, результат нормализуется.

Алгоритм умножения допускает использование в качестве множителя ненормализованного числа. При этом циклы умножения выполняются до тех пор, пока не встретится первая нулевая цифра. Таким образом, выход из цикла умножения происходит до того, как мантисса множителя будет исчерпана. Это эквивалентно тому, что умножение производится на нормализованный операнд. Поэтому если результат получается ненормализованным, то не более чем на одну цифру. В этом случае происходит нормализация результата с соответствующей корректировкой характеристики. В любом случае характеристика результата перед формированием и записью окончательного результата должна быть откорректирована вычитанием суммарного ПНЛ, который в общем случае равен сумме ПНЛ множимого и множителя.

Затем происходит запись младшего слова результата и производится анализ его характеристики для выхода на прерывание по исчезновению порядка или по переполнению. Для этого проверяются значения разрядов [0/1] регистра, где хранится характеристика результата.

Если значения разрядов [0/1] равны 00 или 01, то из этого следует, что характеристика не выходит за пределы [1/7] разрядов, а нулевой (знаковый) разряд свободен для записи в него значения знака результата. Следовательно, переполнения или исчезновения характеристики нет. Поэтому происходят формирование и запись старшего слова результата и окончание операции.

Если значения разрядов [0/1] равны 11, то имеет место исчезновение порядка. Поскольку операции над порядками выполняются отдельно, то при вычитании из суммы характеристик сомножителей числа 64 результат может оказаться отрицательным — произойдет исчезновение характеристики. Этот результат получается в дополнительном коде, о чем свидетельствуют единицы в двух старших разрядах результирующей характеристики.

Пример. Рассмотрим случай, когда сумма характеристик сомножителей равна: a — нулю и b — 63. Вычислим характеристику результата вычитанием числа 64 (40_{16} , дополнительный код CO):

| | | |
|----------------------------------|-----------------|-----------------|
| сумма характеристик сомножителей | а) 0000 0000 | б) 0011 1111 |
| дополнительный код числа 64 | + | + |
| характеристика результата | 1100 0000 | 1100 0000 |
| | ----- | ----- |
| | 1100 0000, | 1111 1111. |

Имеет место исчезновение порядка.

В этих случаях происходят формирование и запись старшего слова результата и в зависимости от значения маски либо записываются нули в качестве результата, либо происходит прерывание по исчезновению порядка.

Если значения разрядов [0/1] равны 10, то возможно как исчезновение, так и переполнение характеристики. Если сумма характеристик сомножителей находится в пределах от 255 до 192 (от FF_{16} до CO_{16}), то это означает, что произошло переполнение порядка результата.

Пример. Рассмотрим случаи, когда сумма характеристик сомножителей равна: $a - 255$ и $b - 192$: вычислим характеристику результата вычитанием числа 64:

| | | |
|----------------------------------|------------------|------------------|
| | а) | б) |
| сумма характеристик сомножителей | 1111 1111 | 1100 0000 |
| дополнительный код числа 64 | + | + |
| характеристика результата | <u>1100 0000</u> | <u>1100 0000</u> |
| | 1011 1111. | 1000 0000. |

Имеет место переполнение порядка.

Когда сумма характеристик сомножителей равна нулю, а множимое ненормализовано на 5 шестнадцатиричных цифр, ПНЛ множимого будет 5 и на эту величину в случае нормализованного результата должна быть откорректирована его характеристика.

Пример.

| | |
|--|-------------------|
| сумма характеристик сомножителей | 0000 0000 |
| дополнительный код числа 64 | +1100 0000 |
| характеристика промежуточного результата | <u>1100 0000.</u> |

Характеристика отрицательна, и перед входом в циклы умножения происходит установка флага исчезновения. При выходе из циклов умножения производится корректировка характеристик промежуточного результата вычитанием ПНЛ:

| | |
|--|-------------------|
| характеристика промежуточного результата | 1100 0000 |
| ПНЛ (дополнительный код) | +1111 1011 |
| характеристика результата | <u>1011 1011.</u> |

Таким образом, значение разрядов [0/1] регистра, где хранится характеристика результата, равно 10 и имеет место исчезновение порядка. В этом случае происходят формирование и запись старшего слова результата и анализ флага исчезновения. Комбинация 10 совместно с флагом исчезновения вызывает прерывание по исчезновению порядка, без флага исчезновения — прерывание по переполнению порядка. Результат записывается с характеристикой, на 128 меньше действительной.

Пример. Произвести умножение двух чисел. Пусть

7C00 0012 — первый операнд;

2200 6000 — второй операнд.

Оба операнда не нормализованы. Множимое — Оп2, множитель — Оп1. Множимое должно быть нормализовано. ПНЛ1 = 4, ПНЛ2 = 2. Отделяем характеристики от мантисс.

Циклы умножения

| Но- мер цик- ла | ТСЧ | Частичное произведение или результат | | Множитель | | Очередные анализи- руемые разряды множителя | Действия | |
|--------------------------|--------|---|---|------------------------|------------------------|---|---|--|
| | | РСДО1 | РСДО2 | РСДВ1 | РСДВ2 | | | |
| 1 | 0 1 | 0000 0000 00C0 0000 | 0000 0000 0000 0000 | 0000 0000 0000 0000 | 0000 0012 0000 0012 | 0010 0010 | Анализ 37, +2М Сдвиг на два разряда вправо | |
| 2 | 0 | 0030 0000 0030 0000 | 0000 0000 0000 0000 | 0000 0000 0000 0000 | 0000 0004 0000 0004 | 0100 0100 | Анализ 37, +0М Сдвиг на 2 разряда вправо | |
| 3 | 1 | 000C 0000 006C 0000 | 0000 0000 0000 0000 | 0000 0000 0000 0000 | 0000 0001 0000 0001 | 0001 0001 | Анализ 37, +1М Сдвиг на 2 разряда вправо | |
| 4 | 0 | 001B 0000 001B 0000 | 0000 0000 0000 0000 | 0000 0000 0000 0000 | 0000 0000 0000 0000 | 0000 0000 | Анализ 37, +0М Сдвиг на 2 разряда вправо | |
| | | 0006 C000 | 0000 0000 | 0000 0000 | 0000 0000 | 0000 | Анализ 37, выход из цикла | |
| | | 0006 C000 | → результат не нормализован на одну цифру | | | | | |
| | | | Мантисса промежуточного результата | | | | | |

Нормализация множимого

| | РСДВ1 | РСДВ2 | |
|---|-------------|------------|-----------------------------|
| мантисса множимого | 0000 6000 | 0000 0000 | (сдвиг на 8 разрядов влево) |
| нормализованная мантисса множимого (1М) | 0060 0000 | 0000 0000. | |
| Удвоенная мантисса множимого (2М) | 0060 0000 | | |
| | + 0060 0000 | | |
| | ----- | | |
| | 00С0 0000, | | |
| утроенная мантисса множимого (3М) | 0060 0000 | | |
| | + 00С0 0000 | | |
| | ----- | | |
| | 0120 0000. | | |

Формирование характеристики промежуточного результата:

| | |
|------------------------|---------------|
| сложение характеристик | 7С + 22 = 9Е, |
| вычитание числа 64 | 9Е + С0 = 5Е. |

Анализ [0/1] разрядов:

5Е = 0101 1110, [0/1] = 01 — исчезновения нет.

Переход к циклам умножения (табл. 26).

Нормализация результата:

| | РСДО1 |
|------------------------------------|---------------------------------------|
| мантисса промежуточного результата | 0006 С000 (сдвиг на 4 разряда влево), |
| мантисса результата | 006С 0000. |

Корректировка характеристики при нормализации (вычитание единицы):

$$5Е + FF = 5D.$$

После корректировки характеристики вычитанием суммарного ПНЛ (в данном случае ПНЛ множимого — Оп1) получаем характеристику результата: $5D + FE = 5B$.

Анализ характеристики результата: $5B = 0101 1011$, разряды [0/1] = 01. Характеристика получилась в допустимых пределах. Окончательный результат: $5B06 0000$.

Выполнение команд типа УМНОЖЕНИЕ для чисел длинного формата отличается от команд для чисел короткого формата тем, что в первом случае обработка чисел осуществляется отдельно для младших и старших слов.

7.4. ДЕЛЕНИЕ

По командам типа ДЕЛЕНИЕ (рис. 85) порядки операндов вычитаются, а мантиссы делятся. Для получения необходимой точности результата оба операнда должны быть предварительно нормализованы. При нормализации из характеристик операндов вычитаются их ПНЛ. Характеристика результата операции определяется следующим образом:

$$\begin{aligned} X_{рез} &= X_{н1} - X_{н2} = (X1 - ПНЛ1) - (X2 - ПНЛ2) + 64 = \\ &= (X1 - X2 + 64) + (ПНЛ2 - ПНЛ1) = X_{пр} + \sum ПНЛ, \end{aligned}$$

где $X_{н1}$ и $X_{н2}$ — нормализованные характеристики операндов; $X1$ и $X2$ — ненормализованные характеристики операндов; ПНЛ1 и ПНЛ2 — параметры нормализации операндов; $X_{пр}$ — промежуточная характеристика результата; $\sum ПНЛ$ — суммарный ПНЛ.

После выборки операндов из регистров РПЗ или ОП отдельно производится определение характеристики частного и выполнение действия над мантиссами операндов. Окончательный результат формируется перед его записью в РПЗ или ОП.

Для определения нормализации мантисс операндов используется анализ нормализации (анализ 31), проводимый в операциях

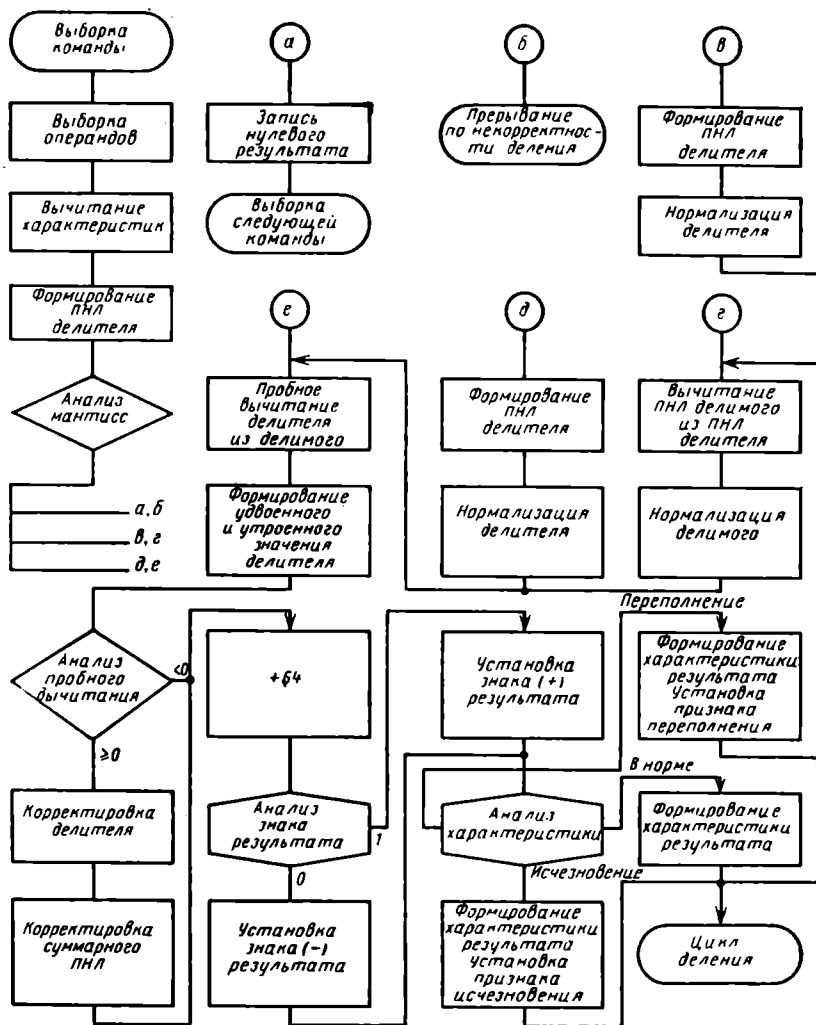


Рис. 85. Схема микропрограммы операции деления

умножения — деления. По его результату формируются и выдаются на МАН коды, которые дают возможность микропрограмме разветвляться на следующие направления:

а) мантисса первого операнда (делимого) равна нулю. В качестве результата записывается нуль и операция завершается;

б) мантисса второго операнда (делителя) равна нулю. Происходит прерывание по некорректности деления;

в) оба операнда не нормализованы. Выполняется нормализация операндов и осуществляется пробное вычитание мантиссы делителя из мантиссы делимого для определения большей из них. Поскольку мантиссы чисел с плавающей запятой всегда меньше единицы, для получения правильного результата деления необходимо, чтобы нормализованная мантисса делимого была меньше нормализованной мантиссы делителя. Если при анализе результата пробного вычитания выясняется, что мантисса делимого больше мантиссы делителя, осуществляется корректировка делителя. С этой целью мантисса делителя, а также его удвоенная и утроенная мантиссы сдвигаются влево на 4 разряда. К суммарному ПНЛ прибавляется единица и устанавливается признак корректировки делителя.

После анализа и установки знака результата, анализируются разряды [0/1] его промежуточной характеристики. Комбинации 00 и 01 в этих разрядах говорят о том, что промежуточная характеристика находится в допустимых пределах, 10 — о том, что имеет место переполнение порядка промежуточного результата, 11 — исчезновение порядка (см. операцию умножения). В случае переполнения или исчезновения порядка устанавливаются соответствующие признаки. Ситуации исчезновения или переполнения порядка промежуточного результата не являются окончательными. В дальнейшем, при формировании характеристики результата сложением промежуточной характеристики с суммарным ПНЛ, характеристика результата может оказаться в области допустимых значений, поскольку суммарный ПНЛ может быть как положительным, так и отрицательным. Промежуточная характеристика складывается с суммарным ПНЛ, и полученная таким образом характеристика результата запоминается отдельно;

г) первый операнд (делимое) не нормализован. Происходит нормализация делимого;

д) второй операнд (делитель) не нормализован. Происходит нормализация делителя;

е) оба операнда нормализованы.

Деление мантисс операндов происходит в цикле деления. Поскольку в этой операции остаток не сохраняется, деление мантисс осуществляется нетрадиционным способом [а. с. 734682 (СССР)]. В его основу положен ускоренный метод деления с восстановлением остатка и формированием двух двоичных разрядов частного. Этот метод можно проиллюстрировать следующим простейшим примером.

Пример: $0,0010\ 0100 : 0,0100 = 0,1001$.

Заранее формируются значения делителя различной кратности:

| единичное (1Д) | удвоенное (2Д) | утроенное (3Д) | прямой код |
|----------------|----------------|----------------|--------------------|
| 0,0100 | 0,1000 | 0,1100 | |
| 1,1100 | 1,1000 | 1,0100 | дополнительный код |

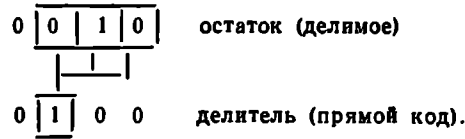
Каждый раз из остатка вычитается сначала 3Д. Если остаток положительный, то очередные цифры частного будут 11. Если остаток отрицательный, то произ-

водится восстановление остатка и из него теперь уже вычитается 2Д. При положительном остатке две очередные цифры частного будут 10, а при отрицательном — производится восстановление остатка и вычитание 1Д и т. д.

| | |
|-----------------|---|
| 0.00100100 | делимое |
| <u>0.100100</u> | сдвиг делимого на два разряда влево |
| +1.0100 | вычитание 3Д |
| <u>1.110100</u> | результат < 0 |
| 0.100100 | восстановление остатка |
| +1.1000 | вычитание 2Д |
| 10 0.000100 | результат > 0, частное 10 (вычиталось 2Д) |
| 0.0100 | сдвиг на два разряда влево |
| +1.0100 | вычитание 3Д |
| <u>1.1000</u> | результат < 0 |
| 0.0100 | восстановление остатка |
| +1.1000 | вычитание 2Д |
| <u>1.1100</u> | остаток < 0 |
| 0.0100 | восстановление остатка |
| +1.1100 | вычитание 1Д |
| 01 0.0000 | результат > 0, частное 01 (вычиталось 1Д) |
| 1001 | |
| ↑ | |

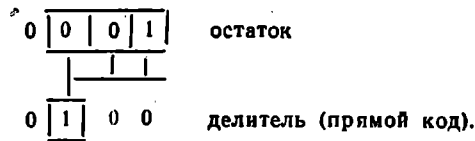
результат; накапливается в отдельном регистре.

Значения двух очередных разрядов частного могут быть предсказаны, что значительно позволяет сократить время выполнения операции. Для этого определяется старший значащий разряд цифры делителя и анализируются соответствующий ему разряд остатка и два его соседних младших разряда. В нашем примере первым остатком, в котором разряды подвергаются исследованию, является делимое



Комбинация разрядов остатка 010. Частное может быть 10 или 01; необходимо сделать пробу на результат 10, для чего следует вычесть 2Д. Операция сокращается на одно вычитание (вычитание 3Д).

Следующим остатком, который подвергается исследованию, является первый положительный остаток:



Комбинация разрядов остатка 001. Частное может быть 01 или 00; необходимо сделать пробу на результат 01, для чего следует вычесть 1Д. Операция сокращается на два вычитания (2Д и 3Д).

С учетом сказанного выше процесс деления в рассмотренном примере будет выглядеть следующим образом:

| | |
|-----------------|---|
| 0.00100100 | делимое; анализ делителя и остатка |
| <u>0.100100</u> | сдвиг на 2 разряда влево |
| +1.1000 | вычитание 2Д |
| 10 0.000100 | результат > 0, частное 10 |
| 0.0100 | сдвиг на 2 разряда влево, анализ делителя и остатка |
| +1.1100 | вычитание 1Д |
| 01 0.0000 | результат > 0, частное 01 |
| 1001 | |
| ↑ | |

результат.

В начале выполнения циклов деления (рис. 86) производится анализ старшей цифры делителя и соответствующей ей цифры остатка. Первым остатком при выполнении деления в машине является делимое, сдвинутое на два разряда влево.

В результате анализа, проводимого в каждом цикле деления, выбирается одно из семи возможных направлений продолжения

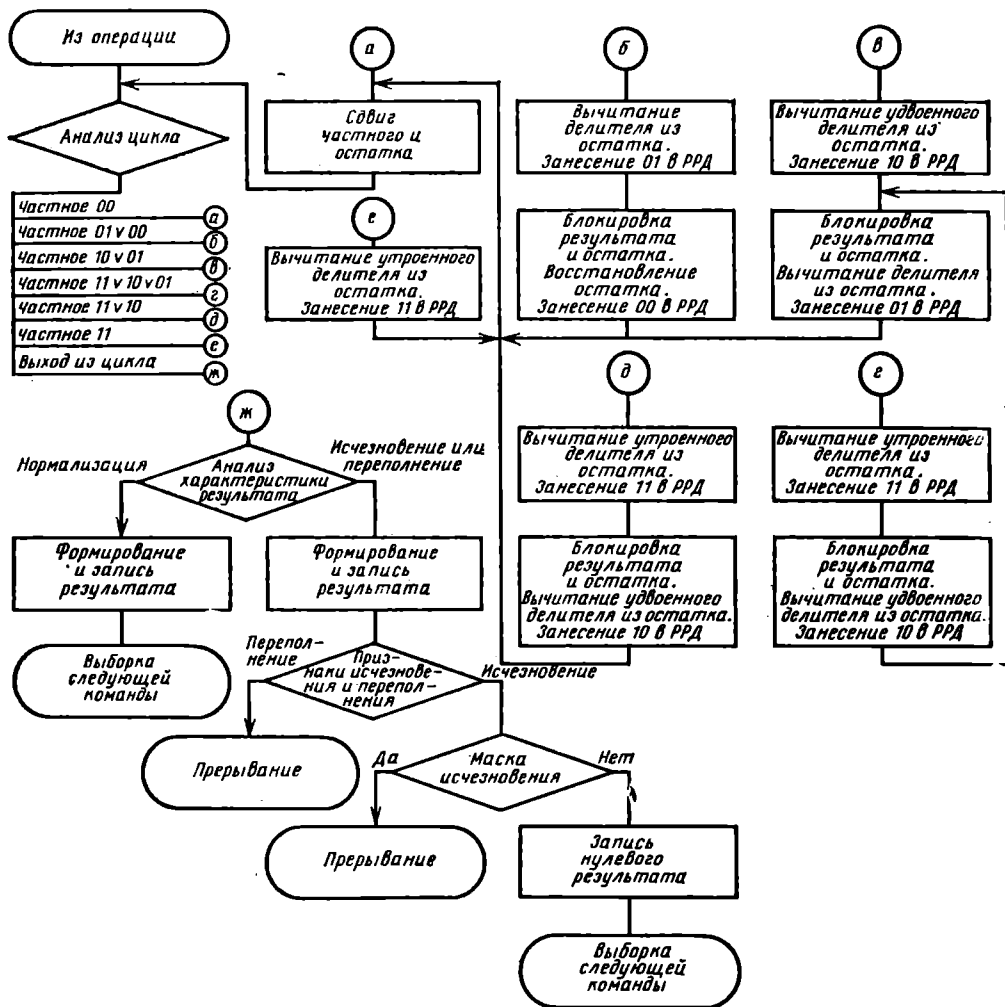


Рис. 86. Схема микропрограммы цикла деления и окончания операции

микропрограммы в зависимости от комбинаций разрядов остатка:

а) 000 — частное 00; б) 001 — частное 01 или 00; в) 010 — частное 10 или 01; г) 011 — частное 11 или 10 или 01; д) 100 или 101 — частное 11 или 10; е) 110 или 111 — частное 11; ж) выход из цикла.

Вместе с анализом значащего разряда и соответствующей ему комбинации разрядов остатка и признаков корректировки делителя в каждом цикле проверяется количество пройденных циклов.

После нужного их количества деление прекращается и происходит выход из цикла на завершение операции. Сформированные перед началом выполнения циклов деления удвоенное и утроенное значения делителя запоминаются в соответствующих РОП.

Поскольку исходные операнды нормализуются, старший значащий разряд делителя находится в старшей тетраде мантииссы, т. е. в разрядах [8/11]. В случае корректировки делителя при попытке разделить число с большей мантииссой на число с меньшей

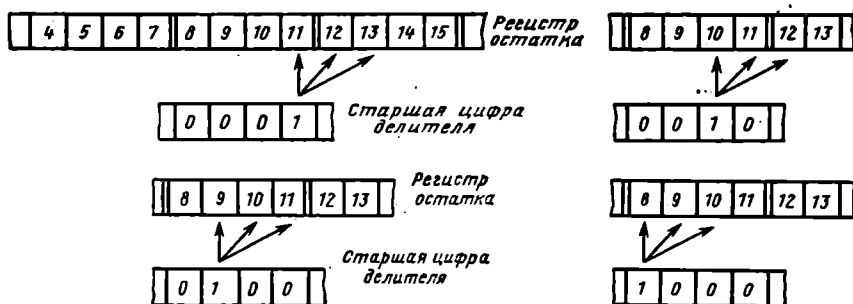


Рис. 87. Зависимость анализируемых разрядов остатка от мантииссы делителя

мантииссой, старший значащий разряд делителя находится в разряде [4/7] регистра, в который был помещен делитель после корректировки (табл. 27, рис. 87).

Т а б л и ц а 27

Зависимость анализируемых разрядов остатка от мантииссы делителя

| Старшая цифра делителя (регистр делителя [8/11]) | Анализируемые разряды регистра остатка | |
|---|--|-----------------------------|
| | делитель корректировался | делитель не корректировался |
| 0001 | [7/9] | [11/13] |
| 0010 ∨ 0011 | [6/8] | [10/12] |
| 0100 ∨ 0101 ∨ 0110 ∨ 0111 | [5/7] | [9/11] |
| 1000 ∨ 1001 ∨ 1010 ∨ 1011 ∨ 1100 ∨ 1101 ∨ 1110 ∨ 1111 | [4/6] | [8/10] |

Для сокращения времени выполнения операции весь описанный выше комплекс выполняется с помощью специальных схем, входящих в состав БФА БОП, по микроприказу АН36 (см. табл. 17).

Перед выполнением циклов деления в счетчик СТЦ помещается код В. В каждом последующем цикле из содержимого счетчика вычитается единица. Последний цикл выполняется при нулевом значении счетчика, после чего следует выход из циклов.

Для выполнения деления мантисса делимого записывается в РСДО1. В РСДО2, РСДВ1 и РСДВ2 записываются нули. В процессе деления вычитание делителя происходит с помощью БАЛ. Результат вычитания (остаток) с М1 записывается в РСДО (пп. 2.4, 2.5), а в РРД записывается очередная пара разрядов частного. При сдвигах влево в РСДО и РСДВ остаток выдвигается за пределы РСДО1, а из РРД в РСДВ2 пара за парой вдвигаются очередные разряды частного. В конце циклов результат (мантисса частного) накапливается в РСДВ2.

При комбинации разрядов остатка 000 (направление *a*, см. рис. 86) частное и остаток сдвигаются на два разряда влево и из содержимого счетчика циклов вычитается единица. Осуществляется анализ цикла.

При комбинации 001 (направление *b*) из остатка вычитается делитель. При этом значение остатка запоминается отдельно для возможного его восстановления. На место прежнего остатка засылается новый. В РРД заносится константа 01. В следующем такте действует условная блокировка (блокировка результата и остатка): если результат вычитания в текущем такте был положительный, значит прежний остаток был больше делителя, и очередные два разряда частного определены верно. Действия следующего такта (восстановление остатка, занесение 00 в РРД) блокируются. Блокировка происходит по микроприказу БЛЗПД: = := 1 (п. 2.5), который имеется в блокируемой микрокоманде при условии, что при вычитании имеется перенос АЛУ (ПАЛУ = 1), т. е. тогда, когда результат вычитания положительный. Блокировка заключается в том, что запрещаются запись в РСДО1 сохраненного значения прежнего остатка и новая запись в РРД. Затем частное с остатком, а с ними и константа в РРД сдвигаются на два разряда влево. В РСДВ2 из РРД вдвигается очередная пара разрядов частного — 01. Из содержимого счетчика циклов вычитается единица. Если же результат вычитания делителя из остатка получился отрицательный, значит остаток был меньше делителя и очередные два разряда частного (01) были определены неверно. В этом случае условная блокировка не действует и в следующем за вычитанием такте в РСДО1 восстанавливается прежний остаток, а в РРД вместо константы 01 заносится константа 00. Затем частное с остатком, а с ними и константа в РРД сдвигаются на два разряда влево. Из содержимого счетчика циклов вычитается единица. В РСДВ2 из РРД вдвигается очередная пара разрядов частного (00). Осуществляется анализ цикла.

При комбинации 010 из остатка вычитается удвоенный делитель. При этом значение остатка запоминается отдельно для возможного его восстановления. Результат вычитания является новым остатком, который засылается в РСДО1 на место прежнего. В РРД заносится константа 10. Если результат вычитания положительный (есть перенос АЛУ), значит остаток был больше удвоенного делителя и очередные два разряда частного (10) опре-

делены верно. В этом случае блокируются действия следующего такта. Если же результат вычитания отрицательный (нет переноса), значит остаток был меньше удвоенного делителя, и очередные два разряда частного определены неправильно. В этом случае действия следующего такта не блокируются, в нем из прежнего, восстановленного остатка вычитается делитель. Полученный результат является новым остатком, который засылается на место прежнего. В РРД заносится константа 01 вместо 10. Затем частное с остатком, а с ними и константа в РРД сдвигаются на два разряда влево. В РСДВ2 из РРД вдвигается очередная пара разрядов частного (01). Из содержимого счетчика циклов вычитается единица. Осуществляется анализ цикла.

При комбинации 011 все происходит аналогичным образом: сначала осуществляется проба на результат, равный 3, вычитанием утроенного делителя из остатка. Если проба прошла (есть перенос), блокируются все остальные пробы. Если эта проба не прошла, осуществляется проба на результат, равный 2, вычитанием удвоенного делителя из восстановленного остатка, а затем, если это необходимо, проба на результат, равный 1. Далее частное с остатком, а с ними и константа в РРД сдвигаются на два разряда влево. Из содержимого счетчика циклов вычитается единица. Осуществляется анализ циклов.

При комбинациях 100 и 101 осуществляются пробы на результат, равный 3, и на результат, равный 2. Затем, как обычно, сдвиг и вычитание единицы из содержимого счетчика циклов.

При комбинациях 110 и 111 очередные два разряда частного могут быть только 11. Поэтому из остатка вычитается утроенный делитель, результат вычитания засылается на место прежнего остатка, в РРД засылается константа 11, следует сдвиг частного и остатка, вычитание единицы из содержимого счетчика циклов и анализ цикла деления.

При нулевом содержимом счетчика циклов независимо от остатка циклический процесс деления заканчивается.

После окончания циклов деления осуществляется анализ характеристики результата для обнаружения исчезновения или переполнения характеристики. Анализ выполняется аналогично тому, как это делается в операции умножения.

Пример. Произвести деление:

0F6C 0000 — первый операнд (делимое),
0700 C000 — второй операнд (делитель).

Вычитание характеристик: $X1 - X2 = 0F - 07 = 08$.

ПНЛ делимого: ПНЛ1 = 0.

Анализ мантисс: второй операнд не нормализован. Это соответствует направлению δ алгоритма деления. ПНЛ2 = 2. Нормализация делителя:

0000 C000 мантисса делителя,
00C0 0000 сдвиг мантиссы на восемь разрядов влево.

Формирование удвоенной мантиссы делителя (2Д):

0 0C0 0000 нормализованная мантисса делителя 1Д,
0 180 0000 сдвиг влево на 1 разряд,
FE80 0000 дополнительный код.

Формирование утроенной мантиссы делителя (ЗД):

| | |
|-----------|---------------------|
| 00C0 0000 | 1Д |
| + | + |
| 0180 0000 | 2Д |
| 0240 0000 | 3Д, |
| FDC0 0000 | дополнительный код. |

Пробное вычитание мантиссы:

| | |
|-----------|---|
| 00C0 0000 | нормализованная мантисса делителя, |
| FF80 0000 | дополнительный код мантиссы делителя; |
| 006C 0000 | нормализованная мантисса делимого, |
| + | + |
| FF80 0000 | дополнительный код нормализованной мантиссы делителя, |
| FFEC 0000 | переноса нет (ПАЛУ=0), результат отрицательный. |

Т а б л и ц а 28

Цикл деления

| | РСДО1 | РСДВ2 | РРД | СТЦ | Действия |
|-------------------------------------|---------------------------------------|--|-----|-----|--|
| Дели- мое (первый остаток) | 006C 0000 | 0000 0000 | 0 | | АН36: 0110 дели- мое 1100 дели- тель |
| | + 01B0 0000 FDC0 0000 | 0000 0000 | 3 | 0B | Сдвиг на два раз- ряда влево Вычитание 3Д |
| | FF70 0000 | переноса АЛУ нет, вычитание неуспешное. Условная блокировка | | | |
| Остаток | 01B0 0000 | 0000 0000 | | | Восстановление остатка |
| | + FE80 0000 | | | | Вычитание 2Д |
| | 0030 0000 | перенос АЛУ есть, вычитание успешное. РРД := 10, запись остатка | | | |
| Остаток | 0030 0000 | 0000 0000 | 2 | | АН36: 0010 оста- ток 1100 дели- тель |
| | + 00C0 0000 FF40 0000 0000 0000 | 0000 0002 | 0 | 0A | Сдвиг на два раз- ряда влево Вычитание 1Д |
| | 0000 0000 | перенос АЛУ есть, вычитание успешное. РРД := 01, запись остатка | | | |
| Остаток | 0000 0000 | 0000 0002 | 1 | | АН36: 0000 оста- ток 1100 дели- тель |
| | 0000 0000 | 0000 0009 | 0 | 09 | Сдвиг влево на два разряда |
| Остаток | 0000 0000 | 0000 0024 | 0 | 08 | АН36. Сдвиг влево на два разряда |
| Остаток Результат | 0000 0000 | 0090 0000 | 0 | 00 | Сдвиг влево на два разряда. Выход из цикла |
| | | ↓ 0090 0000 | | | |

Мантисса делимого меньше мантиссы делителя, корректировки не требуется. Промежуточная характеристика результата:

$$X_{\text{пр}} = (X1 - X2) + 64 = 08 + 40 = 48.$$

Анализ исчезновения или переполнения характеристики (анализ [0/1] рядов): $X_{\text{пр}} = 0100\ 1000$, [0/1] = 01 — характеристика в норме.

Формирование характеристики результата

$$X_{\text{рез}} = X_{\text{пр}} + \sum \text{ПНЛ} = X_{\text{пр}} + \text{ПНЛ}2 = 48 + 2 = 4A.$$

Переход к выполнению циклов деления (табл. 28).

Анализ характеристики результата: $X_{\text{рез}} = 4A = 0100\ 1010$, [0/1] = 01 — характеристика в норме.

Формирование результата:
результат равен 4A90 0000.

Выполнение команд типа ДЕЛЕНИЕ для чисел длинного формата отличается от команд с числами короткого формата тем, что обработка чисел осуществляется отдельно для младших и старших слов.

ГЛАВА 8

КОМАНДЫ ЛОГИЧЕСКОЙ ОБРАБОТКИ

Эти команды (прил. 4) могут быть всех форматов (прил. 5).

Данные, которыми они оперируют, рассматриваются как нечисловая информация, не имеющая внутренней структуры (за исключением команд ПЕРЕСЫЛКА ЦИФР — MVN (D1), ПЕРЕСЫЛКА ЗОН — MVZ (D3) и команд редактирования), и могут иметь либо фиксированную, либо переменную длину (прил. 6). Результаты большинства операций, выполняемых по этим командам, используются для установки в ССП признака результата (прил. 9).

8.1. ОБРАБОТКА ОПЕРАНДОВ ФИКСИРОВАННОЙ ДЛИНЫ

Логические операции (сравнение кодов, И, ИЛИ, исключяющее ИЛИ) выполняются с помощью БАЛ путем задания соответствующих функций ($A - B$, $A \wedge B$, $A \vee B$, $A \oplus B$). Признак результата вырабатывается в БАЛ параллельно с обработкой операндов и записывается в РССП по микроприказам РССП[34/35]: = := ПРСРЛ (признак сравнения логический) для операций сравнения и РССП[34/35]: = ПРЛ (признак логический) для остальных операций (см. табл. 8). В последнем случае, поскольку признак результата может быть только либо нуль, либо единица, РССП[34] микропрограммно обнуляется путем маскирования на магистрали.

Из команд обработки операндов фиксированной длины более подробно рассмотрим микропрограмму команды ПРОВЕРИТЬ ПО МАСКЕ — ТМ (91), которая интересна использованием ре-

зультатов так называемых отсроченных анализов, а также способом установки признака результата. Эта команда имеет формат SI. По ней байт данных, расположенный по адресу первого операнда, маскируется с помощью маски, в качестве которой используется байт из поля 12 непосредственного операнда. Значения отмаскированных бит операнда используются для установки признака результата: 00 — если все отмаскированные биты равны нулю или имеет место нулевая маска; 01 — когда среди отмаскированных бит есть нули и единицы; 11 — если все отмаскированные биты равны единице.

Для выполнения этой операции маска и операнд должны быть расположены в одинаковых позициях двух операционных регистров. Поэтому маску помещают в РОП1 [24/31], а операнд — в РОП2 [24/31].

Байт маски заносится в счетчик СТ РОИ на этапе выборки команды (п. 4.4). В начале микропрограммы команды маска из СТ РОИ через магистраль М3 записывается в РИП [8/15] ($M3[8/15] := \text{СТРОИ}$, $\text{РИП} := := M3$, АБОП = 0.9.1, рис. 88), а затем с помощью перекрестной передачи (п. 3.4) из РИП помещается в РОП1 [24/31] ($M2 := := \text{ПРИП}$, $\text{РОП1} := M2$, АБОП = 2.0.8). По микрокоманде 2.0.8 происходит маскирование передаваемой через М2 информации, благодаря чему в старших разрядах РОП1 оказываются нули.

Байт операнда может находиться в слове ОП в любой позиции. Поэтому по первой микрокоманде (0.9.1) четыре младших разряда кода адреса операнда из РОП15, куда он был помещен на этапе выборки команды, передаются в буферный регистр сравнения РСБФ ($M2 := \text{РОП15}$, $\text{РСБФ} := M2[28/31]$). Два младших разряда кода адреса определяют позицию байта операнда в слове ОП. Далее (АБОП = 2.0.9) ожидается конец обслуживания запроса к ОП за операндом ($\text{ТОЧП} \neq 0$, пп. 2.1, 3.4), который был выдан на этапе выборки команды. После того как операнд поступил в РИП, с помощью логической передачи ЛРИП (п. 3.5) он помещается в позицию последнего байта М1 [24/31] и далее записывается в РОП2 ($M1 := \text{ЛРИП}$, $\text{РОП2} := M1$). Код позиции восьми-

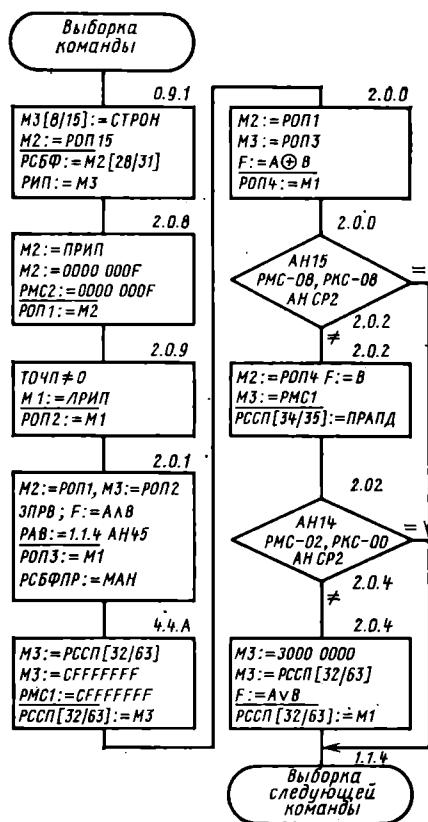


Рис. 88. Микропрограмма команды ПРОВЕРИТЬ ПО МАСКЕ—ТМ (91)

разрядной группы РИП, из которой информация пересылается на М1, определяется значением двух младших разрядов РСБФ, т. е. кодом позиции операнда в слове ОП.

По микрокоманде 2.0.1 с помощью АЛУ осуществляются маскирование операнда и запись в РОПЗ выделенных маской бит для дальнейшего их использования ($M2 := РОП1$, $M3 := РОП2$, $F := A \wedge B$, $РОП3 := M1$). Одновременно по этой же микрокоманде производится анализ результата маскирования по микроприказу АН45 (см. табл. 17). Из всех проверок, производимых по этому микроприказу, в дальнейшем используется только та, результаты которой выдаются на МАН [0]: если в результате операции $F := A \wedge B$ содержимое сумматора ненулевое, то МАН[0] принимает нулевое значение. Следовательно, среди выделенных маской бит операнда имеются единицы. Если содержимое сумматора нулевое, то МАН[0] принимает единичное значение. Этот анализ удобно выполнить одновременно с выполнением операции в АЛУ, однако его результат будет необходим для использования несколько позже. Поэтому с МАН он записывается в 8-разрядный регистр РСБФ. В данном случае он записывается в правую часть (РСБФПР := МАН).

Пример. Рассмотрим случаи, когда среди выделенных маской бит операнда имеются: а) только единицы, б) нули и единицы и в) только нули:

| | а) | б) | в) |
|----------------------|------------|------------|------------|
| операнд | 1100 1011, | 1100 1011, | 1100 1011, |
| маска | 1000 0011, | 1001 0111, | 0001 0100, |
| результат (М1, РОПЗ) | 1000 0011, | 1000 0011, | 0000 0000, |
| МАН[0]=РСБФ[4] | 0; | 0; | 1. |

По этой же микрокоманде (2.0.1) в РАВ записывается адрес возврата (РАВ := 1.1.4) и дается запрос на выборку следующей команды.

По следующей микрокоманде 4.4.А с помощью полноразрядной маски CFFF FFFF в РССП[34/35] устанавливается нулевой признак результата: разряды РССП[34/35], совпадающие с нулевым значением маски, обнуляются, а значения разрядов РССП[32/33] и РССП[36/63], совпадающих с единичным значением маски, остаются без изменения.

По микрокоманде 2.0.0 отмаскированный операнд (хранится в РОПЗ) складывается по mod2 с маской (хранится в РОП1).

Пример.

| | а) | б) | в) |
|-------------------------|------------|------------|------------|
| отмаскированный операнд | 1000 0011, | 1000 0011, | 0000 0000, |
| маска | 1000 0011, | 1001 0111, | 0001 0100, |
| результат (РОП4) | 0000 0000; | 0001 0100; | 0001 0100. |

В дальнейшем анализ на нуль результата, который помещается в РОП4, позволит определить, все ли выделенные маской биты операнда равны единице.

По этой же микрокоманде выполняется ветвление по результату анализа 45, сделанного ранее. Поскольку результат ана-

лиза 45 находится в РСБФ, то для ветвления анализируется содержимое РСБФ — выполняется анализ 15 вида АН СР2 (см. табл. 19). Выбор кода и маски сравнения определяется необходимостью выделить и проанализировать значение РСБФ [4]. Анализ происходит по следующей схеме:

| | а) $CM=0$ | б) $CM \neq 0$ |
|------------------------------------|------------|----------------|
| код сравнения (поле <i>PKC</i>) | 0000 1000, | 0000 1000, |
| маска сравнения (поле <i>PMC</i>) | 0000 1000, | 0000 1000, |
| код условия (РСБФ [0/7]) | xxxx 1xxx, | xxxx 0xxx, |
| МАН [0/3] | 1111; | 1010. |

Здесь *x* означает, что содержимое разряда не играет роли, так как закрытые маской разряды считаются сравнившимися.

При нулевом значении результата ($CM = 0$) левые и правые тетрады РСБФ считаются сравнившимися с константой в *PKC*. При сравнении операция завершается уходом по адресу возврата 1.1.4 на выборку следующей команды с нулевым значением признака результата. Если условие равенства сумматора нулю не выполняется, то это означает, что среди выделенных маской бит операнда имеются единицы. В этом случае происходит переход по базовому адресу следующей микрокоманды. По этой микрокоманде (АБОП = 2.0.2) устанавливается признак результата 01 ($M3 := PMC1, F := B, PCSP[34/35] := ПРАПД$, см. табл. 8) и анализируется, все ли выделенные маской биты операнда равны единице. Если все, то отмаскированный код операнда должен совпадать с кодом самой маски; в этом случае содержимое РОП4 (см. АБОП = 2.0.0) равно нулю. Для анализа содержимого РОП4 оно считывается на *M2*. Затем проводится анализ 14 вида АН СР2 (см. п. 3.5). Из всех проверок этого анализа используется только проверка неравенства нулю содержимого *M2* ($M2 \neq 0$). Анализ выполняется по следующей схеме:

| | а) $M2=0$ | б) $M2 \neq 0$ |
|--|------------|----------------|
| код сравнения (поле <i>PKC</i>) | 0000 0000, | 0000 0000, |
| маска сравнения (поле <i>PMC</i>) | 0000 0010, | 0000 0010, |
| код условия анализа 14 ($M2 \neq 0$) | xxxx xx0x, | xxxx xx1x; |
| МАН[0/3] | 1111; | 1010. |

Здесь *x* означает, что содержимое разряда не играет роли, так как закрытые маской разряды считаются сравнившимися.

При $M2 = 0$ (выделенные маской биты операнда являются нулями и единицами) происходит сравнение левой и правой тетрад с кодом в *PKC* и осуществляется переход по адресу возврата на завершение команды с признаком результата 01. При $M2 \neq 0$, когда все выделенные маской биты операнда являются единицами, происходит несравнение тетрад и осуществляется переход по базовому адресу следующей микрокоманды. По ней происходит установка признака результата 11, после чего операция завершается.

8.2. ОБРАБОТКА ОПЕРАНДОВ ПЕРЕМЕННОЙ ДЛИНЫ

По командам этой группы выполняются операции пересылки, сравнения, И, ИЛИ, исключаящее ИЛИ формата SS с кодами операций D1 — D7 (прил. 4). Их особенностью является то, что операнды могут иметь длину от одного до 256 байт, начинаться с любого слова в ОП, и могут перекрываться в ней любым образом. Системные требования к этим командам определяют обработку операндов последовательно байт за байтом.

Если обращаться к ОП за каждым байтом, то при большой длине операнда время выполнения операции будет велико. Его можно значительно сократить, обрабатывая операнды по словам, однако взаимное расположение операндов в ОП накладывает

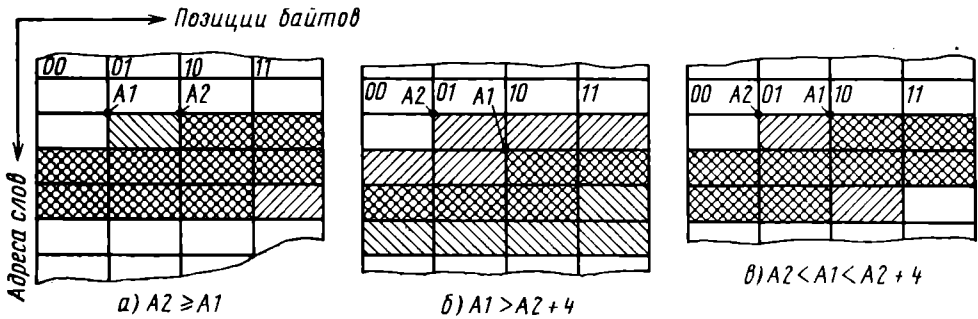


Рис. 89. Примеры перекрытия операндов в ОП

определенные ограничения. Если операнды в ОП не перекрываются, то обработка по словам возможна всегда. Если операнды перекрываются, то возможность пословной обработки определяется видом перекрытия. Если $A2 > A1$ (рис. 89, а), где $A1$ и $A2$ — адреса первого и второго операндов, то обработка операндов по байтам и по словам дает одинаковые результаты и, следовательно, возможна обработка по словам. Если $A1 > A2$, в случае $A1 > A2 + 4$ (рис. 89, б) также возможна обработка по словам, а в случае $A2 < A1 < A2 + 4$ (рис. 89, в) обработка по байтам и по словам дает разные результаты, поэтому возможна обработка только по байтам.

Команды с кодами операций D1 — D7 выполняются по общей микропрограмме, отходя от нее на индивидуальные ветви только для формирования результата.

Микропрограмма начинается с выборки из ОП первого слова второго операнда (Op2) и вычисления количества последующих обращений в ОП (КООП) за его остальными словами (рис. 90). Последнее зависит не только от длины, но и от расположения операнда в ОП. Численно КООП на единицу меньше количества слов, занимаемых операндом в ОП, и равно: $\text{КООП} = (\text{КЛБ} + L) : 4$, где КЛБ — код левого байта операнда; L — длина операнда.

Например, для двух случаев (а и б) расположения операндов в ОП, представленных на рис. 91, при одинаковой длине операндов $L = 6$ КООП будет различным:

КЛБ

а) 10

б) 01

$$\begin{array}{r} + \\ 0110 \\ \hline 1000, \\ = \\ 10, \end{array}$$

$$\begin{array}{r} + \\ 0110 \\ \hline 0111, \\ = \\ 01. \end{array}$$

сумма

КООП

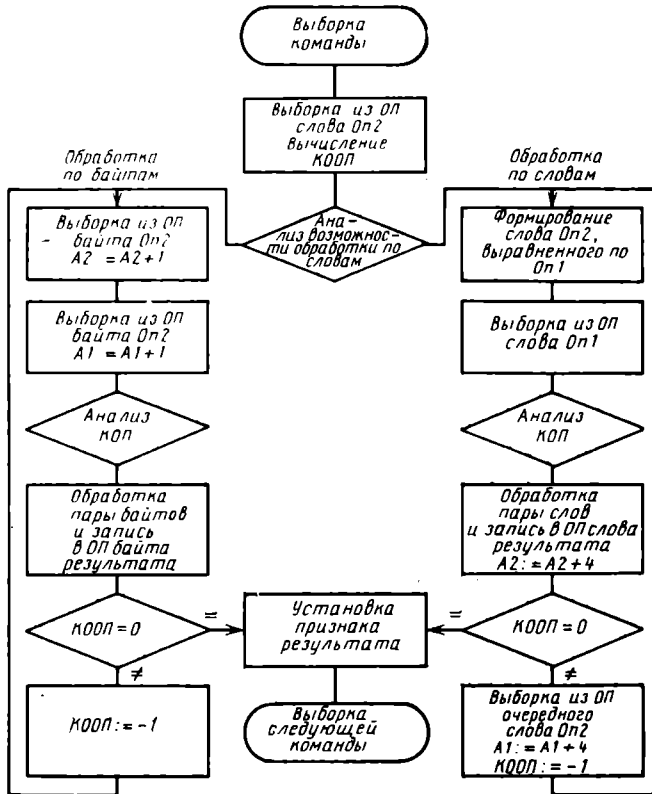
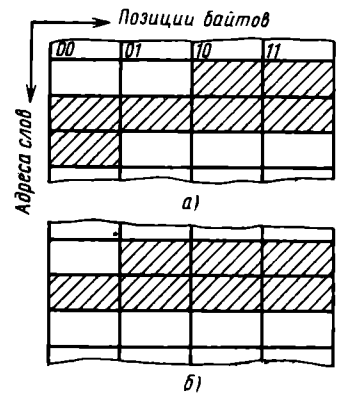


Рис. 90. Схема микропрограммы команд с кодами операций D1—D7

Рис. 91. К определению количества обращений к ОП



КООП фактически определяет количество последующих, после первого, циклов пословной обработки операндов.

Поскольку запись результата производится на место первого операнда, а обработка операндов ведется слева направо, желательно иметь результат, выравненный по границе первого операнда. Для этого при пословной обработке второй операнд выравняется по левой границе первого. Выравнивание осуществляется каждый раз при обработке очередной пары слов. Если $КЛБ\text{ Оп}1 > КЛБ\text{ Оп}2$ (см. рис. 89, в), то при выравнивании Оп2 сдвигается вправо; если $КЛБ\text{ Оп}1 < КЛБ\text{ Оп}2$ (см. рис. 89, а), то влево.

Чтобы при разных направлениях сдвига можно было использовать одну и ту же микропрограмму, применяется аппарат переадресации РОП (п. 2.3). Рассмотрим случай, когда $КЛБ\text{ Оп}1 = 10$, $КЛБ\text{ Оп}2 = 01$ (см. рис. 89, в и 91). Здесь при выравнивании необходимо сдвигать Оп2 вправо.

До начала выравнивания определяются: параметр сдвига, виды байтных масок для освобождения первых слов Оп1 и Оп2 от посторонней информации и константы переадресации.

Параметр сдвига, т. е. количество разрядов, на которые необходимо сдвинуть второй операнд, вырабатывается аппаратурно (п. 2.5) по микроприказу РПС : = ПСДВ. По нему в РПС записывается параметр сдвига (ПСДВ), который формируется при считывании на М2 и М3 адресов Оп1 и Оп2 соответственно из РОП14 и РОП15, где эти адреса хранятся с этапа выборки команды (п. 4.4).

Вид байтной маски (см. табл. 6) определяется аппаратурно при задании ее класса с помощью соответствующей константы и адреса крайнего байта операнда (по адресу операнда, хранящемуся в РОП14 или РОП15). В нашем примере для первого слова Оп2 класс маски 00, адрес байта 01, вид байтной маски 00FF FFFF.

Константы переадресации определяются тем, что при сдвиге вправо операнд после сдвига должен быть записан с магистралей М2 и М3, в РОП1 и РОП2 соответственно. Поэтому согласно п. 2.3 и рис. 33 выбираются следующие константы переадресации: ТРА5 : = 0.5, СТЦ = 1.2 (рис. 92). Таким образом, константа в ТРА5 определяет, что при включении режима переадресации будет происходить запись с М2 и М3 в РОП, номера которых заданы содержимым счетчиков циклов СТЦ.

Затем в РОП3 записываются нули, а в РИП из ОП принимается первое слово Оп2 (рис. 93). Далее содержимое РОП3 и РИП вместе с байтной маской считываются на М2 и М3 соответственно, и освобожденный от посторонней информации операнд с М2 и М3 записывается в сдвигатель РСДВ1 и РСДВ2 по микроприказу записи ЗП СДВ. После этого по микроприказу сдвига СДВ происходит совместный сдвиг в соответствии с вычисленным ранее ПСДВ, а по микроприказу ЧТ СДВ М2, М3 — считывание сдвинутой информации из сдвигателя на М2 и М3. Разряды, выдвинутые вправо за пределы слова, теряются. По микроприказу ВКЛ ПА результат сдвига записывается в РОП1 и РОП2.

Далее выбранное из ОП первое слово Оп1 и выравненное по нему первое слово Оп2, которое хранится в РОП2, вызываются на обработку в АЛУ. Результат, который автоматически оказывается выравненным по границам слова Оп1, помещается в РИП для дальнейшей записи в ОП на место Оп1. В этом же цикле первое слово Оп2, принятое из ОП в РИП, буферизируется в РОП7.

Во втором цикле в РОП3 из РОП7 переписывается первое слово Оп2, а в РИП принимается из ОП его второе слово. Оба слова записываются в РСДВ1 и РСДВ2 соответственно. При сдвиге вправо биты второго слова операнда, вышедшие за пределы РСДВ2, теряются, а биты первого слова, вышедшие вправо за пределы РСДВ1, занимают места освободившихся слева разрядов РСДВ2. Оба цикла одинаковы, поскольку выполняются по одной

и той же микропрограмме. Количество циклов определяется КООП: в каждом цикле значение КООП уменьшается на единицу (см. рис. 90).

Рис. 92. Константы переадресации при сдвиге вправо

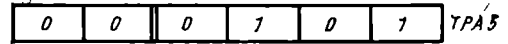
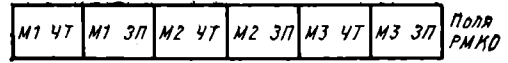
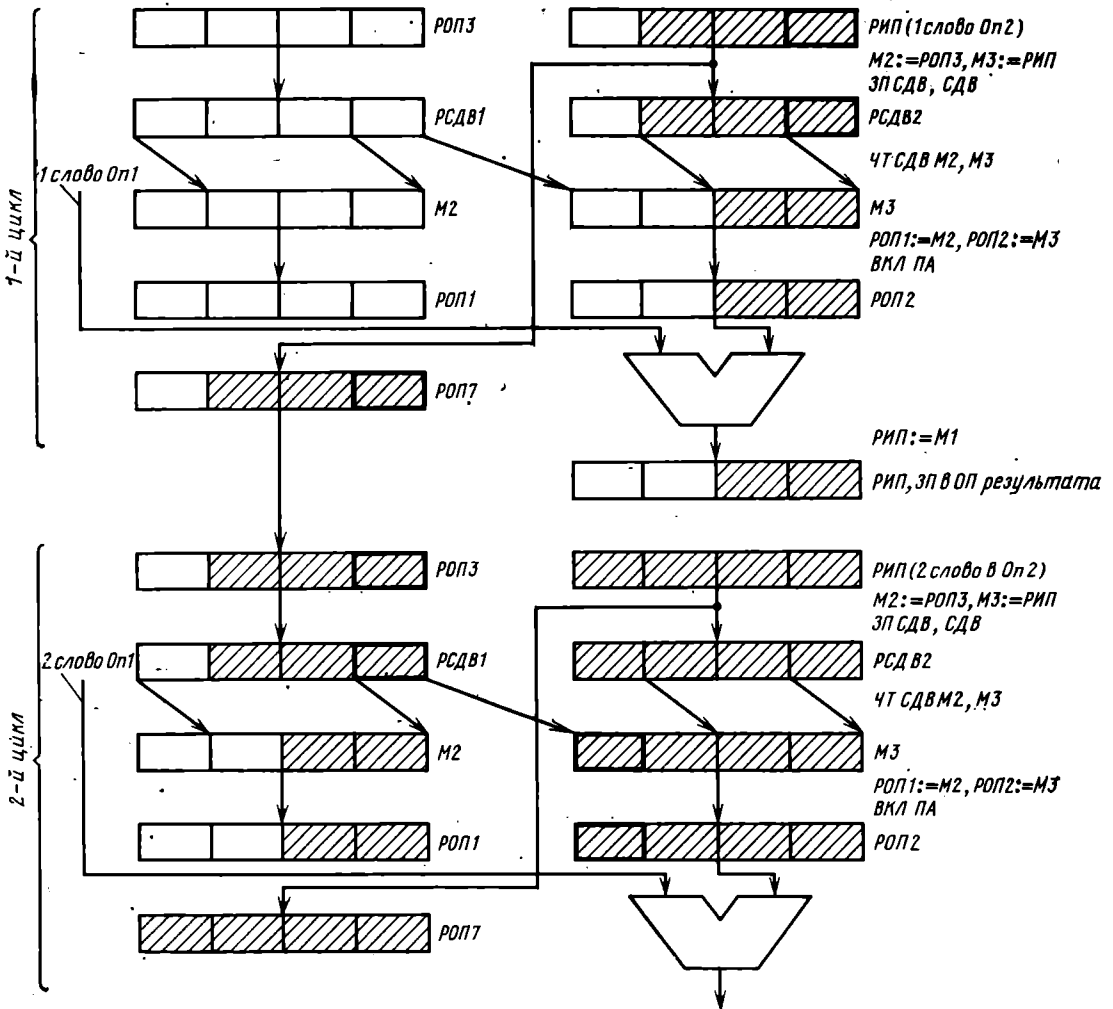


Рис. 93. Выравнивание второго операнда при сдвигах вправо (два цикла)



Рассмотрим случай, когда КЛБ Оп1 = 01, а КЛБ Оп2 = 10 (см. рис. 89, а). Здесь для выравнивания необходимо сдвигать влево Оп2. Если при сдвигах слова операнда вправо в освободившиеся слева разряды вдавигались разряды предыдущего слова, то при сдвигах влево в освободившиеся справа разряды должны

вдвигаться разряды следующего слова операнда. Поэтому к началу первого цикла, в отличие от уже рассмотренного случая, из ОП должны быть выбраны два слова операнда. В цикле обработки как при сдвигах влево, так и при сдвигах вправо используется одна и та же микропрограмма.

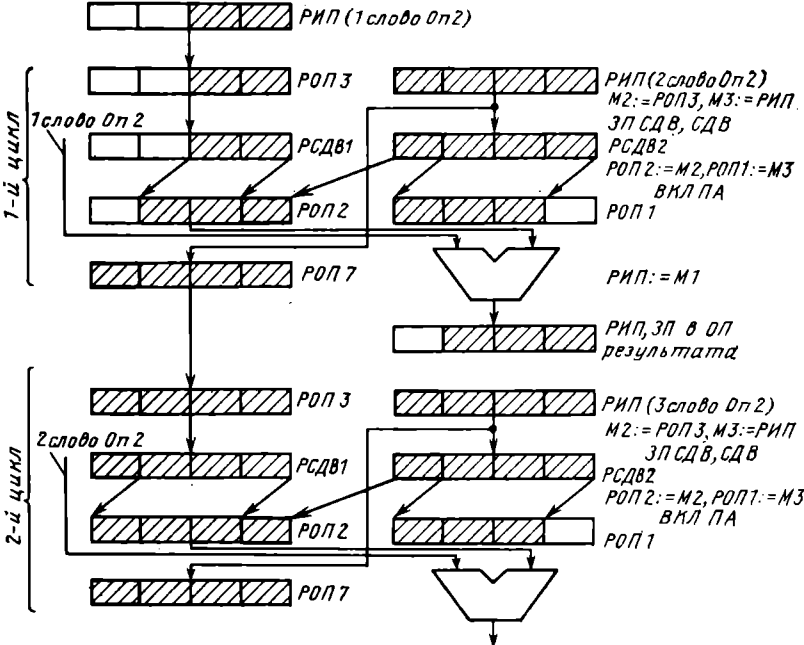


Рис. 94. Выравнивание второго операнда при сдвигах влево (два цикла)

Перед началом первого цикла из ОП выбирается первое слово Оп2 и помещается в РОП3 (рис. 94). Затем в начале цикла из ОП в РИП выбирается второе слово Оп2. Оба слова записываются в РСДВ, где происходит их совместный сдвиг. Если при сдвигах вправо результат (выравненное слово операнда) формируется в РСДВ2, то при сдвигах влево — в РСДВ1. Поскольку выравнивание идет по единой микропрограмме, результат должен быть записан в РОП2. Поэтому при сдвигах влево производится перезапись констант в СТЦ1 и СТЦ2 (рис. 95) так, чтобы содержимое РСДВ1 могло быть записано в РОП2, а РСДВ2 — в РОП1.

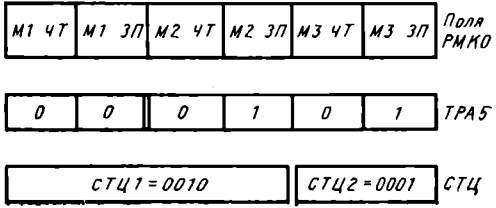


Рис. 95. Константы переадресации при сдвиге влево

При побайтной обработке операндов применяются байтный обмен между РИП и М1 — ЛРИП и побайтная запись в ОП (п. 3.4).

В микропрограммах команд И-НС (D4), ИЛИ-ОС (D6) и ИСКЛЮЧАЮЩЕЕ ИЛИ-ХС (D7) для установки признака результата

используется триггер результата, который устанавливается в нулевое состояние в начале операции. Если очередной результат логической обработки пары операндов отличен от нуля, по микроприказу $TR := \text{МАН}$ триггер результата устанавливается в единичное состояние, при нулевом значении результата по этому же микроприказу состояние триггера не изменяется. После завершения всех циклов операции обнуляется РССП [34/35] на случай нулевого признака результата и производится анализ 6 состояния TR (см. табл. 18). Если $TR = 1$, то в РССП [34/35] записывается признак результата, равный 01 (приложение 10), и команда завершается.

В остальных операциях признак результата устанавливается согласно табл. 8 (п. 2.4).

8.3. ПЕРЕКОДИРОВАНИЕ

Микропрограммы команд ПЕРЕКОДИРОВАТЬ и ПЕРЕКОДИРОВАТЬ И ПРОВЕРИТЬ в большей своей части имеют общие ветви (рис. 96).

Команда ПЕРЕКОДИРОВАТЬ — $TR (DC)$ — имеет формат SS . Эта команда оперирует с таблицей (словарем)-функцией, состоящей из 256 байт. Команда может быть использована для преобразования одного кода, в котором задан первый операнд, в другой, который задан вторым операндом (таблицей). По команде производятся следующие действия:

- последовательно выбираются байты данных (байты аргумента — БА), указанные адресам первого операнда команды;
- складывается значение очередного байта с адресом таблицы, указанным адресом второго операнда команды;
- используя полученный в результате сложения адрес, переходят к элементу таблицы;
- извлекается из элемента таблицы имеющийся там символ (байт функции — БФ) и помещается в поле первого операнда на место байта аргумента.

С целью сокращения времени выполнения команды извлечение аргумента из ОП и запись результата происходят целым словом. Поскольку ограничений на взаимное расположение операндов не накладывается, байт аргумента и байт функции могут быть как в одном, так и в разных словах, извлекаемых из ОП. Обработка в этих случаях идет по разным ветвям микропрограммы.

Рассмотрим пример формирования слова результата для случая, когда байты аргумента и функции расположены в одном слове ОП (рис. 97). Первое слово аргумента считывается из ОП в РИП и помещается в РОП10. Для формирования адреса байта функции байт аргумента из РИП с помощью логической передачи ЛРИП (п. 3.4) пересылается на $M1 [24/31]$. Для осуществления этой передачи код позиции байта аргумента в слове ОП (в нашем примере — 00), являющийся частью адреса первого операнда $Op1$, выбирается из РОП14 и записывается в РСБФ. Адрес первого операнда

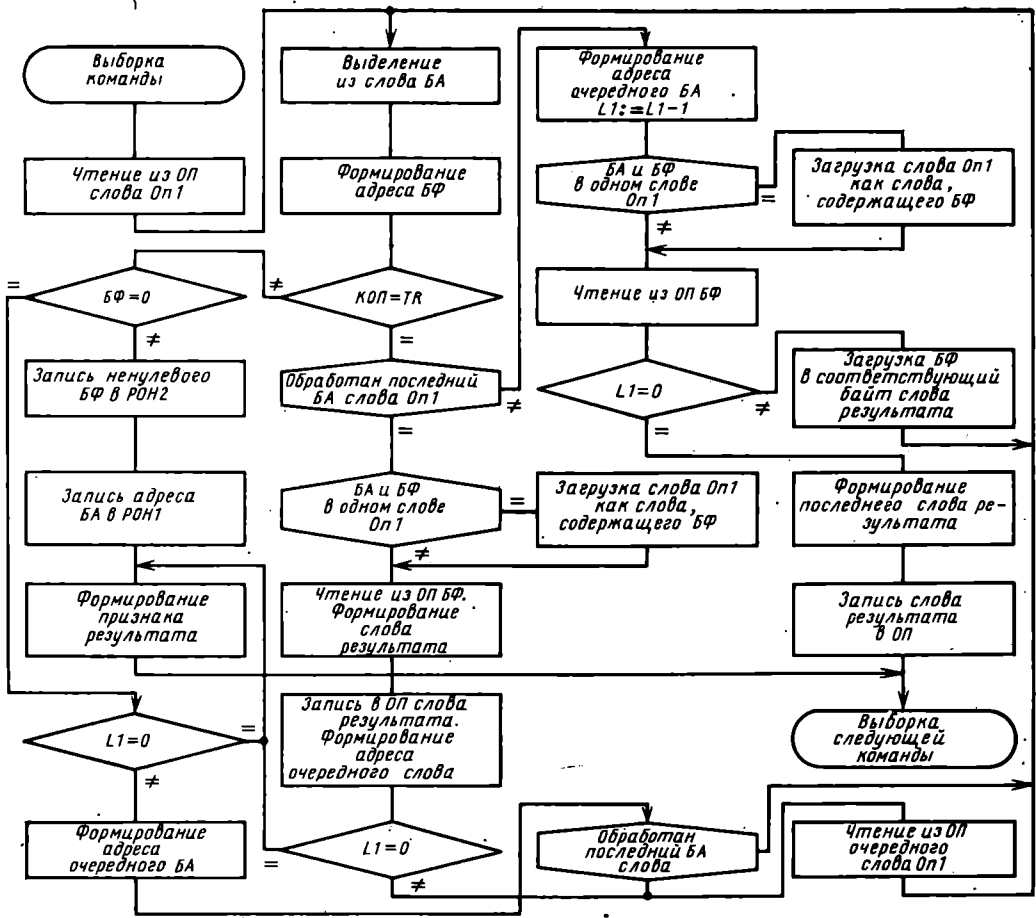


Рис. 96. Схема микропрограмм команд перекодирования

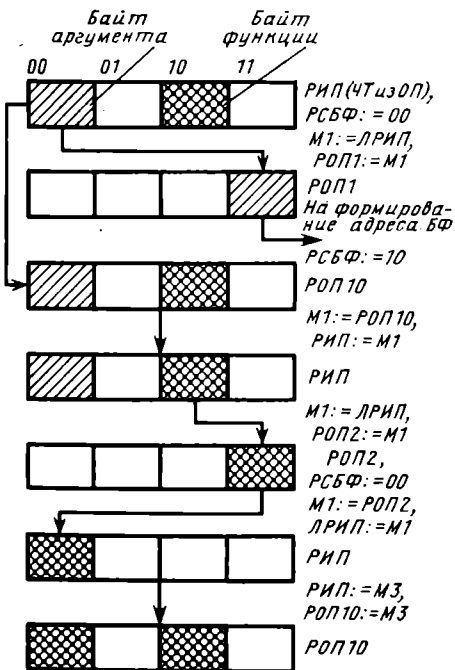


Рис. 97. Схема замещения байтом функции байта аргумента (байты аргумента и функции — в одном слове ОП)

был помещен в РОП14 на этапе выборки команды (п. 4.4). В дальнейшем запись в РСБФ кода позиции текущего байта аргумента производится из РАП.

С М1 [24/31] байт аргумента записывается в РОП1 и в дальнейшем поступает в БАЛ для формирования адреса байта функции. Как только адрес будет сформирован, в РСБФ записывается код байта функции в слове ОП (в нашем примере — 10), а слово ОП, содержащее байт аргумента, выбирается из ОП в РИП (в нашем примере — из РОП10, где оно было запомнено как в буферном регистре). По коду байта функции, находящемуся в РСБФ, байт функции из РИП с помощью ЛРИП считывается на М1 [24/31] и затем запоминается в РОП2[24/31]. В РСБФ снова записывается очередной код позиции байта аргумента в слове ОП для записи байта функции на место байта аргумента. Далее через М1 с помощью ЛРИП байт функции из М1 [24/31] передается в РИП и затем запоминается в РОП10 как в буферном регистре. После этого обрабатывается следующий байт аргумента в выбранном слове первого операнда. Результат компонуется в РОП10 в виде полного слова. Затем слово результата записывается на место обработанного слова первого операнда и из ОП выбирается следующее слово первого операнда.

Команда ПЕРЕКОДИРОВАТЬ И ПРОВЕРИТЬ — TRT (DD) имеет формат SS. Каждый байт функции, выбранный из словаря, используется для того, чтобы определить дальнейший ход операции. Если байт функции равен нулю, то происходит выборка следующего байта аргумента. Если байт функции не равен нулю, то операция заканчивается, при этом адрес соответствующего аргумента помещается в РОП1, а сам байт функции — в РОП2. Эта команда может быть использована, например, для просмотра первого операнда с целью обнаружения определенных символов. В этом случае словарь должен содержать нулевые байты для тех символов аргумента (первого операнда), которые пропускаются, и ненулевые для тех символов аргумента, которые нужно обнаружить. Выборка байт аргумента и байт функции происходит так же, как и в команде ПЕРЕКОДИРОВАТЬ (см. рис. 96), после чего производится проверка на равенство нулю извлеченного из словаря байта функции. В ходе операции признак результата устанавливается равным нулю на тот случай, если все байты функции окажутся нулевыми. Если встретится ненулевой байт функции, то признак 01 (операция не завершилась) или 10 (операция завершилась) переустанавливается микроприказом РССП [34/35] : = = ПРЛ (см. табл. 8) при вычитании количества необработанных байт.

8.4. РЕДАКТИРОВАНИЕ

Команда ОТРЕДАКТИРОВАТЬ — ED (DE) — имеет формат SS. Используется в основном для подготовки десятичных данных для выдачи на печать. Исходные данные — второй операнд —

преобразуются из упакованного формата в формат с зоной и редактируются по образцу, которым является первый операнд. Отредактированный операнд замещает в ОП образец. Длина поля L операнда, заданная в команде, относится к образцу. Команда реагирует на знаки чисел, позволяет вставлять разделители (запятую, десятичную точку, пробел и т. д.) и объединять числовую информацию с текстом. Одной командой можно отредактировать сразу несколько чисел.

В командах редактирования используются специальные триггеры, которые образуют четырехразрядный регистр ТРА7 и некоторые триггеры регистра ТРА6 (см. табл. 18).

Триггер операции редактирования (ТОР) входит в состав ТРА7[0]. Его состояние вместе со значениями цифры, получаемой из поля данных, и символа, получаемого из образца, влияют на то, какой символ помещается в поле результата (первого операнда).

В машине все символы, выбираемые из поля образца, разделены на четыре класса в зависимости от их кодов в таблице ДКОИ (код шестнадцатиричный):

Кл. 0 — код 20. Символ выбора цифры — ВЦФ. При его наличии в поле результата помещается либо распакованная цифра в формате с зоной, либо символ-заполнитель;

Кл. 1 — код 21. Символ начала значимости — НЗН. Выполняет те же функции, что и символ выбора цифры, и, кроме того, устанавливает в единичное состояние ТОР, отмечая тем самым, что все последующие цифры исходных данных воспринимаются как значащие и должны быть помещены в поле результата;

Кл. 2 — код 22. Символ разделения полей результата — РП. Ограничивает отдельные поля при редактировании нескольких полей одной командой и в поле результата замещается символом-заполнителем, при этом ТОР устанавливается в нуль;

Кл. 3 — все остальные коды символа образца. Их обработка осуществляется одинаково: при $ТОР = 1$ символ образца в памяти не меняется; при $ТОР = 0$ символ образца заменяется символом-заполнителем.

Условия установки ТОР и формирование очередного байта результата приведены в табл. 29.

Цифры исходных данных выбирают из поля второго операнда побайтно и проверяют на правильность кодирования. Левая тетрада анализируется первой, а правая хранится до выборки следующего символа образца.

Состояния ТОР имеют локальные значения и влияют только на формирование очередного байта результата. Этот триггер сбрасывается в нулевое состояние в начале операции и в том случае, когда в образце встречаются знак + или символ класса 2, а устанавливается в единичное состояние согласно табл. 29.

Триггер символа класса 1 (ТКЛ1) входит в состав ТРА6[1]. Он устанавливается в единичное состояние тогда, когда текущий

**Установка триггера операции редактирования и формирование
очередного байта результата**

| Символ образца | | | Анализ цифры исходных данных | Состояние ТОР | Значение цифры | Символ, поме- щаемый в поле результата | Смена состоя- ния ТОР |
|----------------|---------------------------------------|--------------------------|------------------------------------|-------------------|---------------------|--|--------------------------------------|
| класс | код | название | | | | | |
| Кл. 0 | 20 | Выбор цифры | Да | T=1 T=0 T=0 | Любое Ц≠0 Ц=0 | Цифра » Заполни- тель | Не меняется T := 1 Не меняется |
| Кл. 1 | 21 | Начало значимости | » | T=1 T=0 T=0 | Любое Ц≠0 Ц=0 | Цифра » Заполни- тель | То же T := 1 T := 1 |
| Кл. 2 | 22 | Раздели- тель полей | Нет | Лю- бое | | То же | T := 0 |
| Кл. 3 | Лю- бой, кроме 20, 21, 22 | Включе- ние текста | » | T=1 T=0 | | Оставить Заполни- тель | Не меняется То же |

символ образца относится к классу 1 (табл. 29). В этом случае устанавливается в единичное состояние ТОР и все последующие цифры будут обработаны как значащие, т. е. этот триггер имеет групповое значение. Триггер ТКЛ1 сбрасывается в нулевое состояние в начале операции и в том случае, если в образце встречается символ класса 2.

Триггер значимости (ТЗНАЧ) входит в состав ТРА7[1]. Он устанавливается в единичное состояние тогда, когда встречается первая ненулевая цифра, записываемая в ОП на место символа образца, и вместе с ТОР используется для установки признака результата. Он сбрасывается в нулевое состояние в начале операции, а также если в образце встречается символ класса 2.

Т а б л и ц а 30

**Установка признака результата
в команде ОТРЕДАКТИРОВАТЬ**

| Состояние триггеров | | Признак результата |
|------------------------|-------|---------------------------------------|
| ТОР | ТЗНАЧ | |
| X | 0 | 00 — все цифры поля данных нулевые |
| 1 | 1 | 01 — результат отрица- тельный |
| 0 | 1 | 10 — результат положи- тельный |

Пр и м е ч а н и е. X — состояние триггера не имеет значения.

Триггер знака (ТЗН1) входит в состав ТРА7[2]. Он устанавливается в единичное состояние тогда, когда в выбранной тетраде находится код знака плюс или минус.

Триггер знака «плюс» (ТЗН «+») входит в состав ТРА7[3]. Он устанавливается в единичное состояние, если код знака соответствует знаку плюс.

Когда в текущей тетраде появляется код знака, эта тетрада как цифра не рассматривается, а происходит выборка очередного байта операнда. При наличии кода знака плюс ТОР сбрасывается в нулевое состояние.

Признак результата относится к последнему отредактированному полю и устанавливается по значению триггеров ТОР и ТЗНАЧ (табл. 30).

Пример. Отредактировать поле десятичных данных (Оп2) по образцу (Оп1)

Оп2 25 67 2С 00 40 8С,

(упакованные десятичные данные)

| | | | | | | | | | | | | | | | |
|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Оп1 (образец) | 40 | 21 | 20 | 20 | 6В | 20 | 20 | 15 | 22 | 21 | 20 | 20 | 6В | 20 | 20, |
| Результат | 40 | F2 | F5 | F6 | 6В | F7 | F2 | 15 | 40 | 40 | 40 | F4 | 6В | F0 | F8, |
| ТОР | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0, |
| ТЗНАЧ | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1. |

Байт второго операнда выбирается из ОП и помещается в РОП (рис. 98). Затем из ОП выбирается байт первого операнда (символ образца) и тоже запоминается в РОП. Одновременно производится установка триггеров ТЗН1 и ТЗН«+» с помощью следующего приема: байт Оп2 из РОП считывается на М2 и производится анализ 40(3) (см. табл. 17). При этом на МАН[2] будет единица, если в разрядах М2 [28/31] находится код знака плюс, а на МАН [3] будет еди-

Т а б л и ц а 31

Последовательность действий при выполнении операции редактирования

| Номер байта Оп2 | Значение байта Оп1 | Класс символа Оп1 | Анализ цифры Оп1 | Символ, помещаемый в поле результата | Результат | Состояние ТОР | Состояние ТЗНАЧ |
|-----------------|--------------------|-------------------|------------------|--------------------------------------|-----------|---------------|-----------------|
| 1 | 40 | 3 | Нет | Заполнитель | 40 | 0 | 0 |
| 2 | 21 | 1 | Да | Цифра с зоной | F2 | 1 | 1 |
| 3 | 20 | 0 | » | То же | F5 | 1 | 1 |
| 4 | 20 | 0 | » | » | F6 | 1 | 1 |
| 5 | 6В | 3 | Нет | Символ образца | 6В | 1 | 1 |
| 6 | 20 | 0 | Да | Цифра с зоной | F7 | 1 | 1 |
| 7 | 20 | 0 | » | То же | F2 | 0 | 1 |
| 8 | 15 | 3 | Нет | Символ образца | 15 | 0 | 1 |
| 9 | 22 | 2 | » | Заполнитель | 40 | 0 | 0 |
| 10 | 21 | 1 | Да | » | 40 | 1 | 0 |
| 11 | 20 | 0 | » | » | 40 | 1 | 0 |
| 12 | 20 | 0 | » | Цифра с зоной | F4 | 1 | 1 |
| 13 | 6В | 3 | Нет | Символ образца | 6В | 1 | 1 |
| 14 | 20 | 0 | Да | Цифра с зоной | F0 | 1 | 1 |
| 15 | 20 | 0 | » | То же | F8 | 0 | 1 |

ница, если в разрядах M2 [28/31] содержится код знака плюс или минус. В этой же микрокоманде во втором полуктакте результат анализа с МАН записывается в ТРА7 по микроприказу ТРА7:=МАН, устанавливая тем самым в соответствующее состояние триггеры ТЗН1 и ТЗН«+». Анализ состояния этих триггеров в дальнейшем производится обычным путем по микроприказу АН7-с необходимым признаком ветвления. Состояние других триггеров анализируется аналогично.

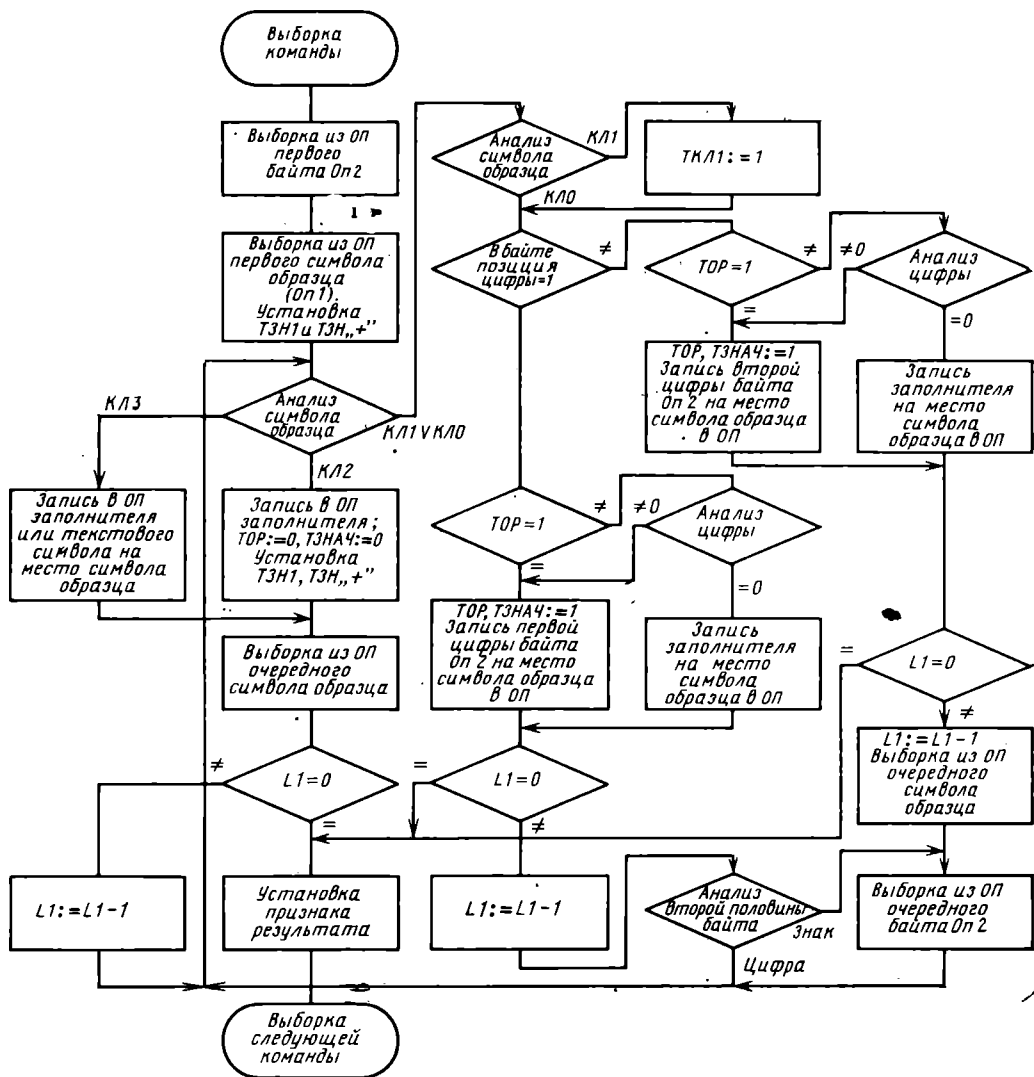


Рис. 98. Схема микропрограмм команд редактирования

Далее обрабатывается символ образца. В зависимости от кода класса выполняются те или иные действия сначала над левой тетрадой байта ОП2, а после выбора очередного символа образца — над правой тетрадой байта ОП2.

Процесс выполнения операции редактирования и формирования результата представлен в табл. 31.

Если учесть, что в ДКОИ символ 40 означает пробел, символ 6В — запятую, символ 15 — новую строку, то отредактированный по образцу операнд при выводе на печать имеет следующий вид:

256,72

4,08

Первая цифра первой строки отстоит на один, а второй строки — на три пробела от начала строки.

Команда **ОТРЕДАКТИРОВАТЬ И ОТМЕТИТЬ** — EDMK (DF) — формата SS отличается от предыдущей тем, что адрес первой значащей цифры результата помещается в PОН1[8/31]. Обе команды выполняются по единому алгоритму.

8.5. СДВИГИ

При выполнении команд сдвигов (прил. 4) операнды фиксированной длины в одно или два слова рассматриваются как произвольные коды без знаков. Поэтому, в отличие от арифметических сдвигов влево, старший разряд как знаковый не рассматривается, он не сохраняется и переполнение не фиксируется. При сдвигах вправо значение старшего разряда как знакового в освободившиеся разряды не распространяется. При сдвигах в любую сторону выдвигаемые разряды теряются, а освобождающиеся разряды заполняются нулями. Признак результата остается без изменения. Адрес второго операнда для обращения к памяти не используется. Код его младших шести разрядов указывает число позиций, на которое должен быть сдвинут первый операнд.

Алгоритм выполнения логических сдвигов аналогичен рассмотренным выше (п. 6.1) алгоритмам арифметических сдвигов, за исключением того, что при сдвигах влево направление сдвига задается микроприказом сдвиг влево логический (СДВ ЛЕВ ЛОГ), отключающим схему фиксации переполнения сдвигателя (п. 2.5).

ГЛАВА 9

КОМАНДЫ ОБРАБОТКИ ДЕСЯТИЧНЫХ ДАННЫХ

Связь человека с машиной облегчается, если исходная информация и результаты вычислений представляются в привычной для человека десятичной системе. Если время ввода исходной информации и вывода результатов невелико по сравнению со временем самих вычислений и требуется высокая точность, то последние в машине обычно выполняются над числами, представленными в двоичной системе. В этих случаях перевод из десятичной системы в двоичную и обратно производится с помощью команд преобразования чисел из одной системы счисления в другую (конвертирования). Если же вычисления невелики по сравнению с объемом вводимой и выводимой информации, как это бывает при экономических расчетах, то время, занимаемое операциями конвертирования, оказывается соизмеримым с продолжительностью вычисле-

ний. В таких ситуациях вычисления лучше производить над числами, представленными в десятичной системе. Для этой цели в системе команд ЕС ЭВМ предусмотрены команды выполнения операций непосредственно в десятичной системе счисления. В этих операциях участвуют числа, представленные в двоично-десятичной форме.

Набор команд десятичной арифметики обеспечивает сложение, вычитание, сравнение, умножение и деление, а также преобразование форматов данных (прил. 4). В результате всех операций типа сложения и сравнения устанавливается признак результата. Десятичные операнды располагаются в основной памяти, начиная с любого байта, и могут иметь длину от одного до шестнадцати байтов (прил. 6). Данные рассматриваются как целые числа. В операции они участвуют выравненными по правым границам. Обработка ведется справа налево. О потере переносов или значащих цифр сигнализируется как о десятичном переполнении.

9.1. СЛОЖЕНИЕ, ВЫЧИТАНИЕ, СРАВНЕНИЕ

При выполнении команд этого типа (рис. 99) можно выделить этапы, отличающие алгоритм, реализованный в ЕС-1033, от алгоритмов выполнения аналогичных команд в других машинах. К таким этапам, существенно сокращающим время выполнения операций, относятся:

выборка второго операнда из ОП с выравниванием его по правой границе первого операнда и запись в регистровую память. Это позволяет в дальнейшем получить результат, расположенный таким образом относительно целочисленных границ, что его можно без предварительного выравнивания записывать в ОП;

выполнение самой операции сложения (вычитания) над целыми словами параллельно с выборкой из ОП очередного слова первого операнда;

формирование признака результата производится одновременно с записью результата в ОП.

Алгоритмы операций перечисленных команд имеют общие ветви микропрограмм.

Выборка и выравнивание второго операнда при сдвигах влево и вправо происходят по единой микропрограмме (рис. 100). Вначале вычисляются адреса крайних правых байтов первого ($A1 + L1$) и второго ($A2 + L2$) операндов и количество обращений в память за младшим словом второго операнда. В ТРА5 записывается константа 0.5, а в счетчик циклов — константа 5.6 (п. 2.3). Последнее означает, что в режиме переадресации ВКЛ ПА будет разрешена запись с М2 и М3 (рис. 101, а), на которые будет считано содержимое РСДВ1 и РСДВ2 после сдвига, в регистры РОП5 и РОП6. Затем анализируется КООП2, который находится в СТ2 РОН. Если $КООП2 = 0$, то запрошенное из ОП слово является единственным. Если $КООП2 \neq 0$, то запрошенное из ОП слово является младшим словом операнда. Слово операнда после вы-

борки его из ОП в РИП (РИП ← ОП, рис. 102, 103) помещается затем в РОП1, а в РИП считывается очередное, старшее слово операнда. Таким образом, перед записью второго операнда в сдвигатель его младшее и старшее слова оказываются расположенными соответственно в РОП1 и РИП. В процессе записи младшего слова

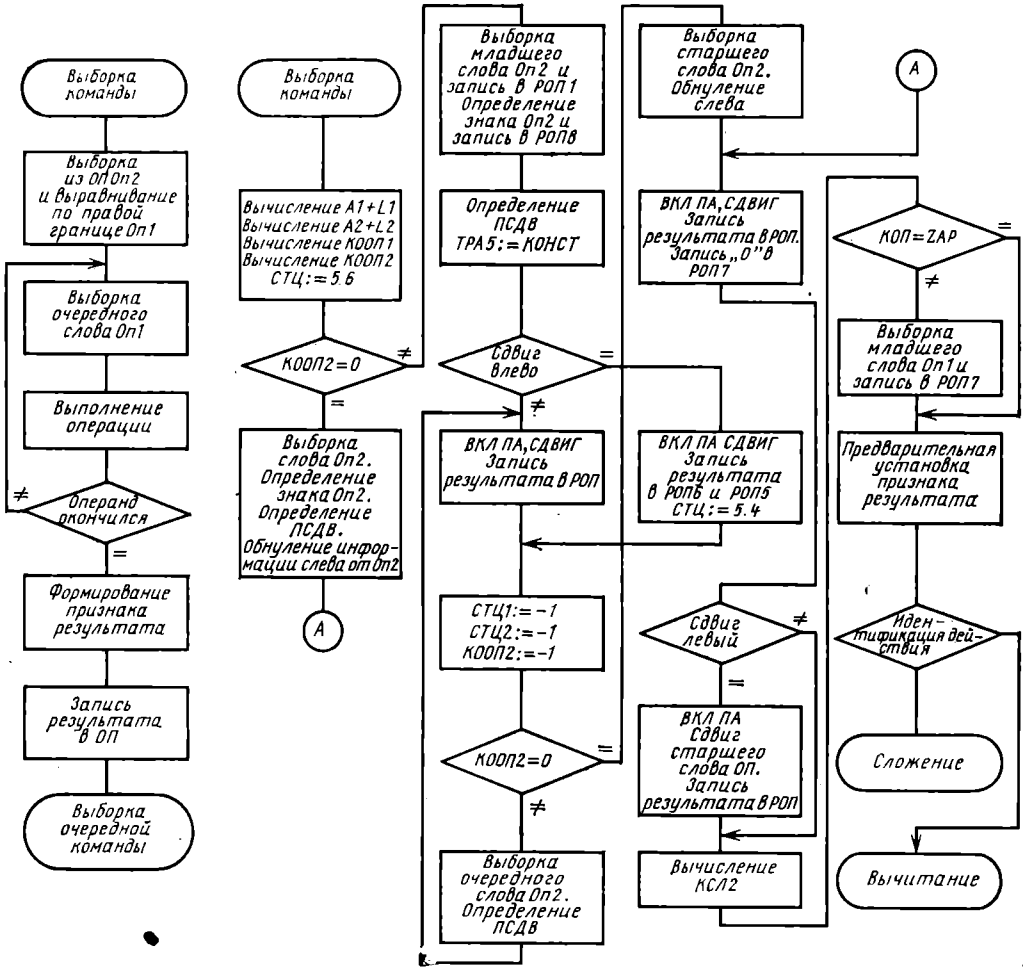


Рис. 99. Схема алгоритма выполнения команд десятичного сложения, вычитания и сравнения

Рис. 100. Схема микропрограммы выборки второго операнда в командах десятичного сложения, вычитания и сравнения

в сдвигатель отделяется и помещается в РОП8 код знака второго операнда. Это осуществляется с помощью передачи ЛРИП (п. 3.4). По кодам крайних правых и байт Оп1 и Оп2 определяется ПСДВ (п. 2.5), который запоминается в регистре параметра сдвига (РПС := ПСДВ). Далее анализируется и запоминается направление сдвига, поскольку начальные и конечные этапы выравнивания и записи второго операнда в поле регистровой памяти при сдвигах вправо и влево (рис. 103) несколько различаются. Младшее и старшее слова операнда пересылаются через магистрали МЗ

и М2 и записываются в сдвигатели РСДВ2 и РСДВ1, после чего происходит сдвиг по ПСДВ (микроприказы: записи в сдвигатель ЗП СДВ и сдвиг СДВ). Одновременно с этим старшее слово с М2 заносится в освободившийся РОП1 (РОП1 := М2). В следующем цикле оно будет использовано как младшее слово. Результат из сдвигателя в режиме переадресации (ВКЛ ПА) записывается в операционные регистры, номера которых определяются содержимым счетчиков циклов СТЦ1 и СТЦ2.

При сдвиге вправо в первом цикле результат будет записан в РОП6 и РОП5, причем в РОП6 будет окончательный результат, а в РОП5 промежуточный (РОП5, РОП6 ← РСДВ, ВКЛ ПА). Содержимое СТЦ1 и СТЦ2 уменьшается на единицу, и начинается следующий цикл выборки. В этом цикле очередное слово операнда поступает из ОП в РИП. В качестве старшего оно считывается на М2, а слово из РОП1 считывается на М3 в качестве младшего.

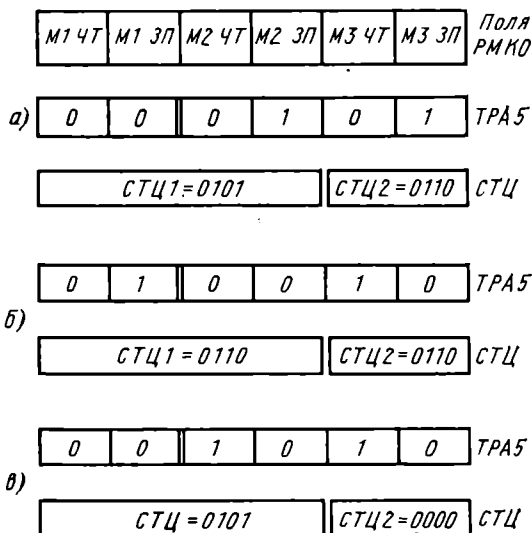


Рис. 101. Константы переадресации в операциях сложения, вычитания и сравнения

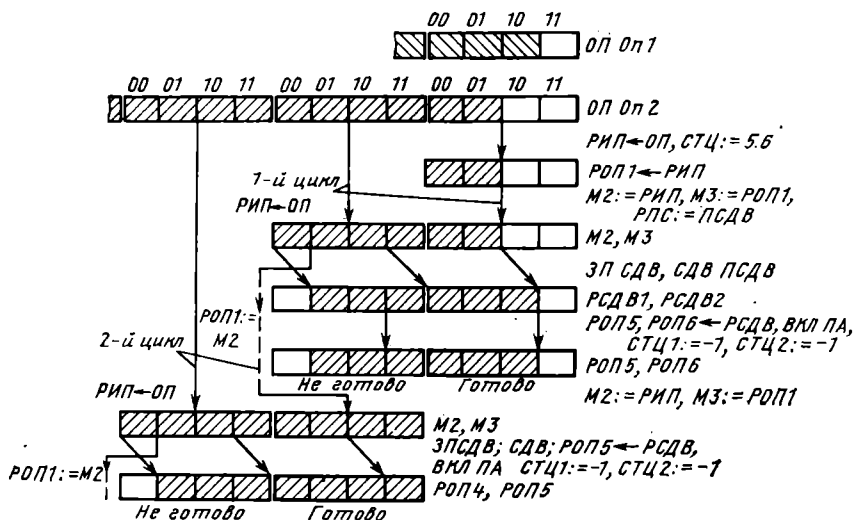


Рис. 102. Выравнивание второго операнда при сдвигах вправо

Оба слова записываются в сдвигатель РСДВ. После сдвига результат записывается в режиме ВКЛ ПА уже в РОП5 и РОП4, причем в первом из них будет окончательный результат, а во втором —

промежуточный. При последнем сдвиге окончательный результат будет расположен в РОП(6 — N)—РОП6, где N — количество слов, занимаемых результатом. При сдвиге влево результат, записанный в первом цикле в регистры РОП5 и РОП6, будет окончательным. Поэтому после его записи содержимое СТЦ должно быть откорректировано занесением новой константы (СТЦ := 5.4). В следующем цикле, после уменьшения на единицу содержимых СТЦ1 и СТЦ2, результат сдвига записывается в РОП3 и РОП4,

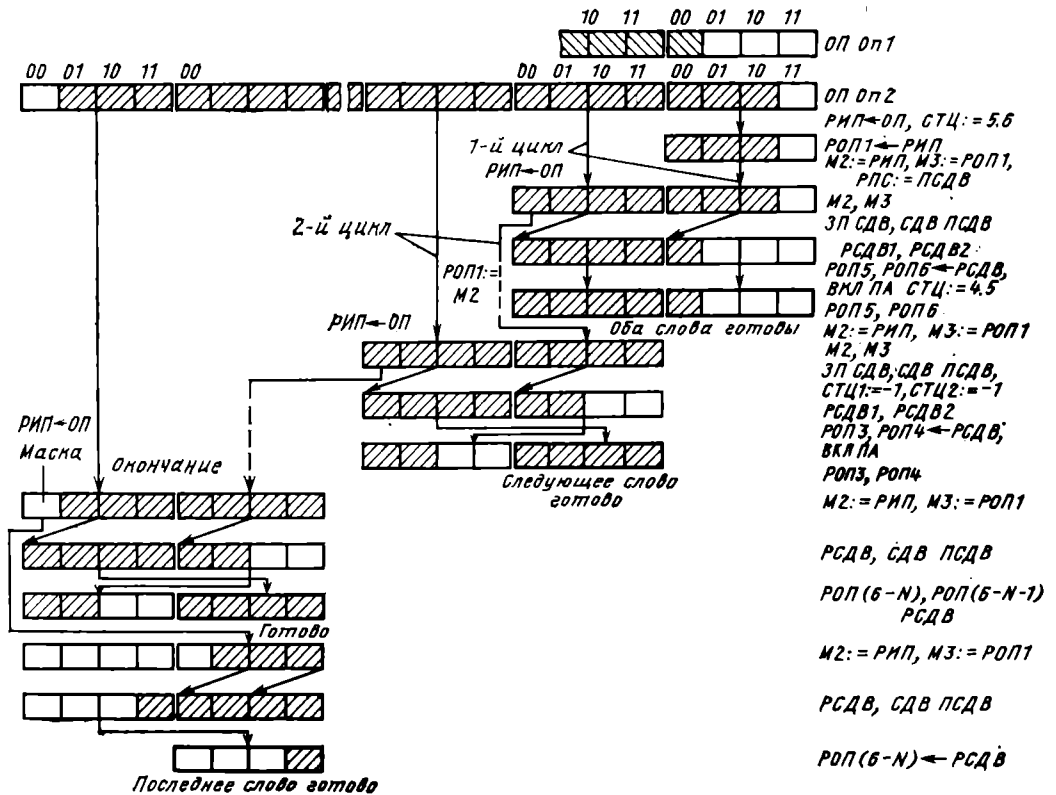


Рис. 103. Выравнивание второго операнда при сдвигах влево

причем в РОП4 — окончательный результат, а в РОП3 — промежуточный. Так как при сдвиге старшего слова его старшие разряды выходят за пределы сдвигателя, то при выходе из цикла старшее слово, находящееся в РОП1, дополнительно сдвигается и помещается в РОП(6 — N).

В конце выборки операнда (см. рис. 100) вычисляется количество слов (КСЛ2), из которых состоит Оп2, расположенный в поле регистровой памяти после его выравнивания по правой границе первого операнда:

$$\text{КСЛ2} = \frac{(\overline{A1 + L1}) + L2}{4},$$

где $\overline{A1 + L1}$ — обратный код суммы $A1 + L1$.

Вычисление и запись в СТ РОН величины КСЛ2 производится аналогично вычислению и записи КООП. Затем производится выборка из ОП младшего слова первого операнда и его запись в РОП7. Предварительно производится анализ КОП и, в случае команды СЛОЖЕНИЕ С ОЧИСТКОЙ — ZAP(F8), выборка первого операнда не производится. При этом в РОП7 сохраняются ранее находившиеся там нули. Далее выполняется предварительная установка признака результата в РССП и производится анализ идентификатора действия по коду операции. В операциях СЛОЖЕНИЕ ДЕСЯТИЧНОЕ — AP (FA) и ВЫЧИТАНИЕ ДЕСЯТИЧНОЕ — SP (FB) идентификатор действия определяет ветвь сложения или вычитания на основе анализа соотношения знаков операндов. Для операции СЛОЖЕНИЕ С ОЧИСТКОЙ всегда определяется ветвь сложения, для операции СРАВНЕНИЕ ДЕСЯТИЧНОЕ — CP (F9) — ветвь вычитания.

Таким образом, после выборки второй операнд оказывается размещенным в регистровой памяти РОП и выравненным по границе первого операнда. В дальнейшем выбираются лишь очередные слова Оп1 из ОП, и сама операция выполняется на фоне обслуживания ОП. Это существенно уменьшает время выполнения команды. Результат записывается в регистровую память на место Оп2, выравненного по границам Оп1, что дает возможность производить запись в ОП по адресу Оп1 целыми словами.

Для сложения двоично-десятичных чисел в АЛУ одно из них необходимо иметь в коде «с избытком 6». Такой код позволяет на обычном двоичном сумматоре идентифицировать межтетрадные двоичные переносы. Если при сложении чисел перенос из тетрады отсутствует, то такая тетрада корректируется вычитанием из нее числа 0110, что обычно заменяется прибавлением числа 1010, являющегося дополнительным кодом числа 0110. При этом вновь возникающие вторичные переносы теряются.

Пример:

$$\begin{array}{r}
 \text{а) } 15 + 3 = 18 \\
 \begin{array}{r}
 0001\ 0101\ 15 \\
 +0110\ 0110\ +\ 6 \\
 \hline
 0111\ 1011\ \overline{15+6} \text{ (код с избытком 6)} \\
 +0000\ 0011\ \underline{3} \\
 \hline
 0111\ 1110\ (15+6)+3 \text{ нет переноса} \\
 +1010\ 1010\ \underline{6} \text{ (корр.)} \\
 \hline
 0001\ 1000\ \underline{18} \text{ — результат}
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 \text{б) } 15 + 8 = 23 \\
 \begin{array}{r}
 0001\ 0101\ 15 \\
 +0110\ 0110\ +\ 6 \\
 \hline
 0111\ 1011\ \overline{15+6} \text{ (код с избытком)} \\
 +0000\ 1000\ \underline{8} \\
 \hline
 1000\ 0011\ (15+6)+8 \text{ есть перенос} \\
 \times \leftarrow -1\ \underline{-} \\
 +1010\ 0000\ \underline{6} \text{ (корр.)} \\
 \hline
 0010\ 0011\ \underline{23} \text{ — результат}
 \end{array}
 \end{array}$$

В случае *a* отсутствует перенос из обеих тетрад, поэтому обе они корректируются. В случае *б* имеется перенос только из младшей тетрады, поэтому коррекции подлежит старшая тетрада.

Вычитание двоично-десятичных чисел производится обычным образом. Если при вычитании в какой-либо тетраде перенос отсутствует, то такая тетрада корректируется, как и в случае сложения.

Пример:

$$\begin{array}{r} \text{а) } 15 - 3 = 12 \\ 0001\ 0101 \quad 15 \\ + 1111\ 1101 \quad - 3 \\ \hline 0001\ 0010 \quad \underline{12} - \text{результат} \\ \uparrow\ _ \quad \uparrow\ _ \end{array}$$

$$\begin{array}{r} \text{б) } 21 - 3 = 18 \\ 0010\ 0001 \quad 21 \\ + 1111\ 1101 \quad - 3 \\ \hline 0001\ 1110 \\ \uparrow\ _ \\ + 0000\ 1010 \quad (\text{корр.}) \\ \hline 0001\ 1000 \quad \underline{18} - \text{результат} \end{array}$$

Если в результате вычитания операндов суммарный перенос отсутствует (нет переноса сумматора), то это означает, что получен отрицательный результат в виде дополнения до 10^n . Такой результат должен быть откорректирован повторным вычитанием из «0».

Пример:

$$\begin{array}{r} 6 - 15 = -9 \\ 0000\ 0110 \quad 6 \\ + \\ 1110\ 1011 \quad -15 \\ \hline 1111\ 0001 \quad \text{результат, переноса нет} \\ \text{Вычитание результата из «0»:} \\ 0000\ 0000 \\ + \\ 0000\ 1111 \quad \text{дополнительный код результата} \\ \hline 0000\ 1111 \quad \text{переноса нет} \\ - \\ 0000\ 0110 \quad \text{коррекция на 6} \\ \hline 0000\ 1001 \quad \text{абсолютное значение результата} \end{array}$$

В цикле сложения (рис. 104) очередное слово Оп2 подготавливается в коде «с избытком 6». Для этого в СЦ помещается константа 6.6, а в ТРА5 — константа 1.2 (см. рис. 101, б). Это означает, что разрешены чтение на М3 из РОП6 и запись с М1 в РОП6 (в первом цикле). На М2 считывается операнд 6666 6666 (константа из БОП3), на М3 в режиме ВКЛ ПА из РОП6 считывается второй операнд (младшее слово). Для АЛУ задается функция сложения и результат сложения обоих операндов с М1 снова записывается в РОП6. Далее из РОП7 считывается на М2 младшее слово Оп1. Оно было занесено туда перед началом цикла сложения. Младшее слово Оп2 в режиме ВКЛ ПА считывается из РОП6 на М3, и оба операнда складываются. Результат с М1 в режиме ВКЛ ПА записывается в РОП6. После этого содержимое СЦ уменьшается на единицу. Благодаря этому в следующем цикле сложения очередное слово Оп2 будет выбрано из РОП5 и в него же помещен результат и т. д. Выход из цикла сложения происходит после того, как будет исчерпан Оп1 (КООП1 = 0, коды 10 и 11).

После выхода из цикла сложения (рис. 105) происходит запись в ОП младшего слова результата, которое хранится в РОП6. Попутно происходит предварительная установка признака результата. Адрес младшего слова Оп1 для записи результата был

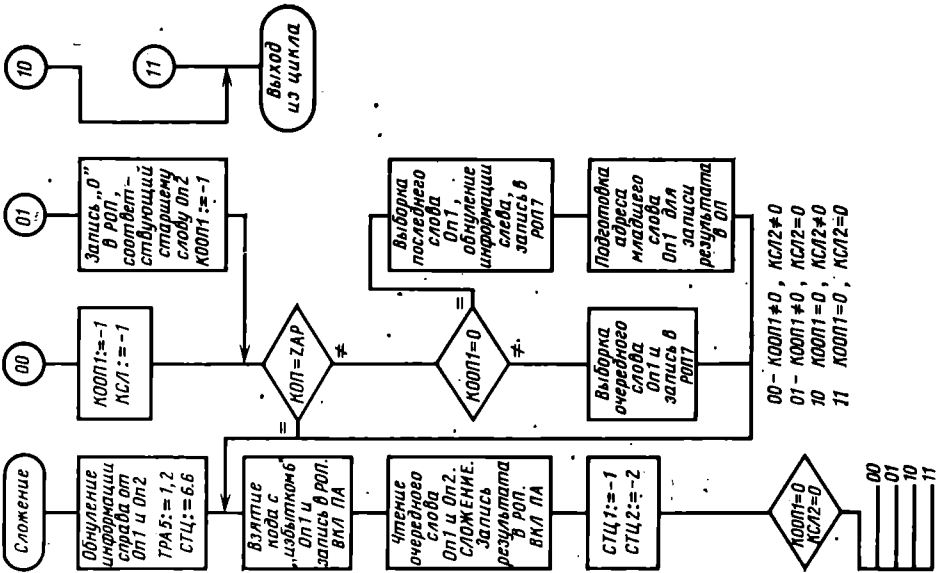


Рис. 104. Схема микропрограммы цикла сложения

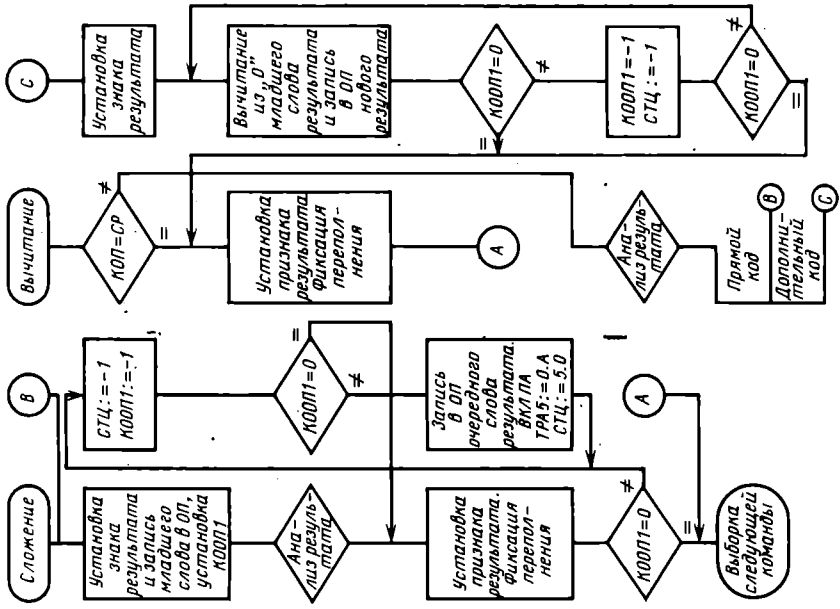


Рис. 105. Схема микропрограммы выхода из цикла сложения и вычитания

подготовлен заранее в цикле сложения сразу после выборки старшего слова Оп1 (см. рис. 104). При этом код позиции правого байта был обнулен, так что фактически был сформирован адрес слова, содержащего четыре или меньше последних байт операнда. Код последнего (правого) байта операнда был записан

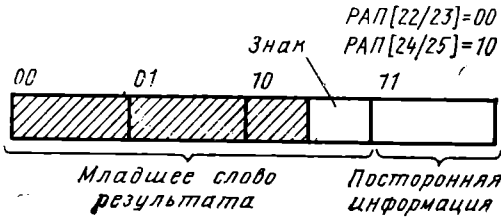


Рис. 106. Подготовка младшего слова результата для записи в ОП

в РАП[24/25] в конце выборки второго операнда. Тем самым были подготовлены все условия для побайтной записи (п. 3.4) младшего слова результата (рис. 106). К младшему слову подформируется знак, и оно записывается в ОП. Далее результат анализи-

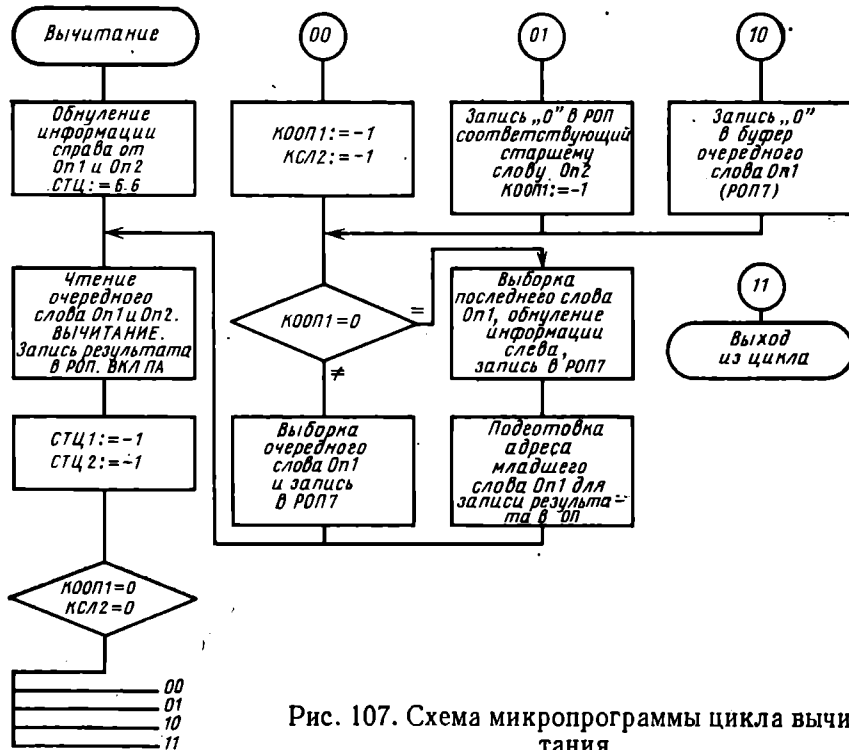


Рис. 107. Схема микропрограммы цикла вычитания

руется. В случае десятичного переполнения фиксируется возникшая ситуация и происходит переустановка признака результата. Для записи следующего слова результата в ТРА5 записывается константа О.А, разрешающая в режиме ВКЛ ПА считывание на М2 из того РОП, номер которого задан счетчиком СТЦ, а в СТЦ записывается константа 5.0 (см. рис. 101, в). В очередном цикле содержимое соответствующего РОП считывается на М2, слово результата помещается в РИП и затем записывается в ОП. Самое

старшее слово результата записывается в ОП аналогично младшему с применением аппарата побайтной записи. При этом в РАП содержится следующая информация: $РАП[22/23] = \text{код левого байта}$, $РАП[24/25] = 11$.

Цикл вычитания построен аналогично циклу сложения (рис. 107). Разница заключается в том, что не требуется формирование кода с «избытком 6». Кроме того, в том случае, когда первый операнд оказывается исчерпанным, а второй нет (код 10), первый операнд дополняется нулями слева (буферный регистр РОП7 очередного слова первого операнда заполняется нулями) и операция продолжается. Выход из цикла вычитания производится тогда, когда оба операнда будут исчерпаны (код 11).

После выхода из цикла вычитания (см. рис. 105) анализируется код операции для выделения операции сравнения — СР. В этом случае записи результата в ОП не происходит, а операция заканчивается формированием признака результата.

После выхода из цикла сложения или вычитания анализируется результат. Если первый операнд больше второго, то результат будет получен в прямом коде. Запись его в ОП происходит, как и в операции сложения. Если же второй операнд больше первого, то результат будет получен в виде дополнения до 10^n . В этом случае результат необходимо вычесть из нуля. Повторное вычитание совмещено с записью результата в ОП. При этом результат помещается в тот же РОП, из которого было взято слово результата в дополнительном коде. Запись в ОП происходит аналогично описанному выше.

9.2. УМНОЖЕНИЕ

По команде умножения произведение множителя (второй операнд) и множимого (первый операнд) замещает множимое. Длина множителя ограничена 15 цифрами и знаком и должна быть меньше длины множимого. Код длины $L2$, больший семи или равный коду длины $L1$, рассматривается как неправильная спецификация. Так как число цифр в произведении равно суммарному числу цифр операндов, то в поле множимого должно быть столько нулевых цифр, сколько их содержится в поле множителя. В противном случае данные считаются неправильными и происходит прерывание программы. Максимальная длина произведения не должна превышать 31 цифры. Знак произведения определяется по правилу знаков алгебры.

Алгоритм выполнения команды умножения (рис. 108) начинается с микропрограммы выборки второго операнда, которая одинакова для команд умножения и деления. Вторым операнд выбирается из памяти, выравнивается по левой целочисленной границе слова и записывается в РОП5 и РОП6 (рис. 109). Знаковая тетрада и посторонняя информация (т. е. все тетрады, расположенные вправо от нее) обнуляются. Если длина операнда пре-

вышает одно слово, то обнуляются младшие разряды РОП6, а сам операнд (без знака) помещается в РОП5 и РОП6 (рис. 109, а). Если длина операнда менее одного слова, он помещается в старшие разряды РОП5, а его младшие разряды и все разряды РОП6 обнуляются (рис. 109, б). Выборка операнда начинается со старшего слова, выравнивание операндов производится в процессе его записи в РОП. Для этого по коду адреса крайнего левого байта формируется параметр сдвига (п. 2.5). При выборке младшего слова используется байтная маска (п. 2.2).

Первый операнд выбирается из ОП, начиная с младшего слова, и выравнивается по правой целочисленной границе слова. При выборке старшего слова используется байтное маскирование. Выравненный операнд размещается в РОП0 — РОП3, причем обнуляются знаковая тетрада и биты, содержащие постороннюю информацию, т. е. биты, расположенные слева от поля операнда.

Проверка правильности данных заключается в определении соответствия полей операндов. Поскольку число цифр в произведении равно суммарному числу цифр сомножителей, поле множимого должно содержать, по меньшей мере, столько нулевых старших

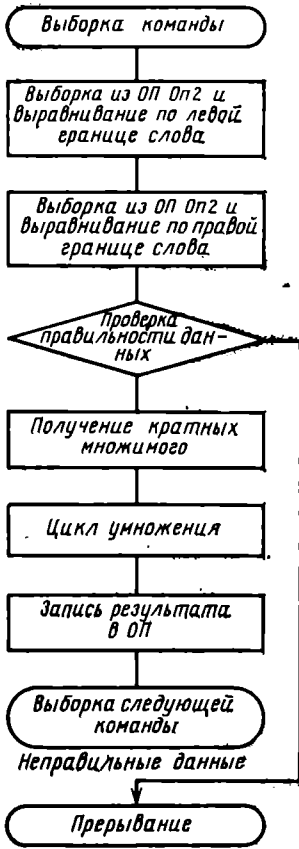


Рис. 108. Схема алгоритма выполнения команды умножения

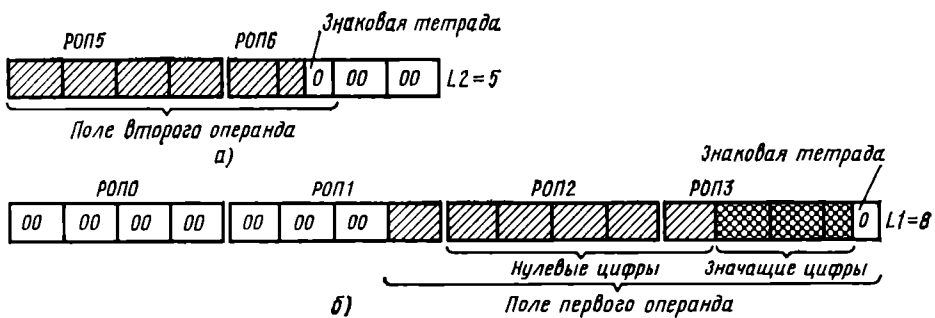


Рис. 109. Расположение операндов в регистровой памяти

цифр, каков размер поля множителя. Это необходимое количество нулевых цифр равно $L2$. Разность $L1 - L2$ определяет максимальное количество значащих цифр первого операнда. Младшие разряды этой разности определяют в слове операнда, содержащем первую значащую цифру, количество находящихся слева от нее байт, которые должны быть нулевыми. Номер РОП,

в котором находится старшая значащая цифра первого операнда, определяется инверсным значением двух старших разрядов разности $L1 - L2$. В примере, показанном на рис. 109, $L1 = 1000$, $L2 = 0101$, $L1 - L2 = 0011$, номер РОП, в котором должна находиться старшая значащая цифра первого операнда — 11 (РОПЗ).

По результату анализа младших разрядов разности $L1 - L2$ микропрограмма разветвляется для подготовки соответствующей маски (табл. 32).

Т а б л и ц а 32

Вид маски для выделения байт, содержащих нули

| Младшие разряды разности $L1-L2$ | Позиции байт, в которых должны содержаться нули | Вид маски |
|----------------------------------|---|-----------|
| 00 | 00, 01, 10, 11 | — |
| 01 | 00, 01, 10 | FFFF FF00 |
| 10 | 00, 01 | FFFF 0000 |
| 11 | 00 | FF00 0000 |

Затем номер операционного регистра, в котором должна находиться первая значащая цифра и с которого следует начать проверку на наличие байт, содержащих нули, заносится в СТЦ, а в ТРА5 записывается константа 2.0. В режиме ВКЛ ПА происходит считывание на магистраль М1 из того операционного регистра, номер которого задан содержимым СТЦ1. Одновременно на М1 накладывается соответствующая маска, отделяя тем самым цифры, которые должны быть значащими, и содержимое магистрали анализируется на нуль. Если содержимое магистрали, выделенное с помощью маски, не равно нулю, то это означает, что в проверяемом поле имеется ненулевая цифра и происходит прерывание по причине обнаружения неправильных данных. Если в выделенных маской разрядах находятся нули, то содержимое счетчика СТЦ уменьшается на единицу и таким же образом продолжается проверка старших слов операнда. При этом маска не используется, так как на наличие ненулевых цифр слово проверяется целиком.

Вычисление кратных множимого происходит перед циклом умножения. Множимое, кратное 1, 2, 4, 6 и 8, представляется в коде «с избытком 6» и помещается в операционные регистры. Каждое кратное множимое размещается в двух смежных РОП (старшее и младшее слова). Для вычисления кратных множимого используется АЛУ. В качестве множимого берется второй операнд.

Количество циклов умножения (параметр цикла — ПЦ) определяется количеством цифр множителя. Сам цикл умножения заключается в прибавлении к частичному произведению (первое частичное произведение — нули) кратных множимого (кратность множимого соответствует значению очередной цифры множителя, на которую производится умножение) и в сдвиге результата на одну цифру вправо.

Перед началом цикла первое частичное произведение (нули) размещается в РСДО1 и РСДО2, а множитель — в РСДВ1 и РСДВ2 (берется из РОП0, РОП1, РОП2 и РОП3).

В начале цикла (рис. 110) производится сдвиг множителя на четыре разряда вправо (на одну двоично-десятичную цифру). При чтении на М2 и М3 сдвинутой информации из РСДВ1 и РСДВ2 последняя цифра множителя оказывается в позиции М3 [28/31],

где она анализируется. В зависимости от ее значения к первому частичному произведению прибавляется множимое соответствующей кратности. Поскольку имеется неполный набор кратных множимого, то при отсутствии множимого соответствующей кратности оно формируется. Например, утроенное множимое получается сложением удвоенного и единичного. Далее полученный результат, т. е. частичное произведение, записывается в РСДО1, сдвигается вправо на четыре разряда, и цикл повторяется. Выход из цикла производится по нулевому значению параметра цикла (ПЦ = 0). В этом случае частичное произведение является окончательным результатом и занимает все четыре слова в РСДО и РСДВ. Произведение оказывается выравненным по левой целочисленной границе слова, в знаковой тетраде находятся нули. Если

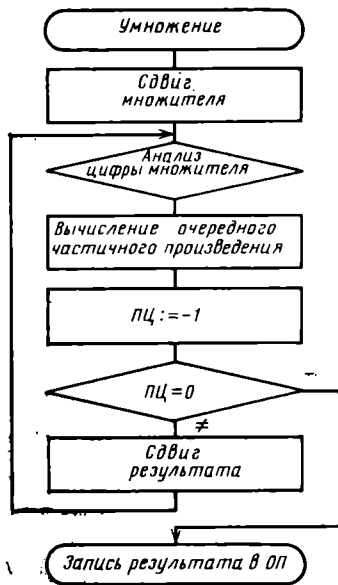


Рис. 110. Схема микропрограммы цикла умножения

длина результата меньше шестнадцати байт, справа от знаковой тетрады будут находиться нули.

Пример. Пусть в машине вычисляется произведение: $25 \times 231 = 5775$.

Произведение двухзначного числа на трехзначное может содержать пять цифр. Поэтому поле множимого должно иметь, по меньшей мере, три старшие нулевые цифры (по числу цифр множителя). Знак размещается в последней тетраде.

Таким образом:

поле первого операнда — 00 0250,
поле второго операнда — 2310.

В цикле умножения, представленном в табл. 33, для большей наглядности рассматриваемого процесса разрядность РСДО и РСДВ вместо 64 условно принята равной 16.

При записи результата в ОП он предварительно из РСДО и РСДВ помещается в РОП0 — РОП3. Поскольку результат должен быть записан по адресу первого операнда, то в процессе записи он выравнивается по левой границе Оп1. Для этого по коду крайнего левого байта слова (код 00) и коду крайнего левого байта результата формируется соответствующий параметр сдвига (п. 2.5). Запись результата начинается со старшего слова с использованием

Цикл умножения

| РСДО | РСДВ | Действия |
|--------------------------|--------------|---|
| 0000 0000 25 | 2310 0231 | Сдвиг множителя Анализ последней цифры Прибавление множимого кратности 1 |
| 2500 0250 75 | 0231 0023 | Сдвиг результата и множителя Анализ последней цифры Прибавление множимого кратности 3 |
| 7750 0775 50 | 0023 0002 | Сдвиг результата и множителя Анализ последней цифры Прибавление множимого кратности 2 |
| 5775 0577 | 0002 5000 | Сдвиг результата и множителя] |
| Произведение (без знака) | | |

механизма переадресации РОП. При записи старших и младших слов результата используется аппарат записи байтов. К младшему слову результата подформировывается знак произведения.

9.3. ДЕЛЕНИЕ

Делимое (первый операнд) делится на делитель и замещается частным и остатком. Частное помещается в левую часть поля первого операнда, остаток — в правую часть и имеет длину, равную длине делителя. Частное и остаток занимают все поле делимого, длина поля частного равна $L1 - L2$. Случаи, когда код длины делителя больше семи или больше или равен коду длины делимого, рассматриваются как неправильная спецификация. Получение частного с числом цифр, большим допустимого, фиксируется как некорректность десятичного деления. Знак частного определяется по правилу знаков алгебры; знак остатка тот же, что и у делимого. Чтобы частное поместилось в отведенное для него поле, делимое должно иметь, по меньшей

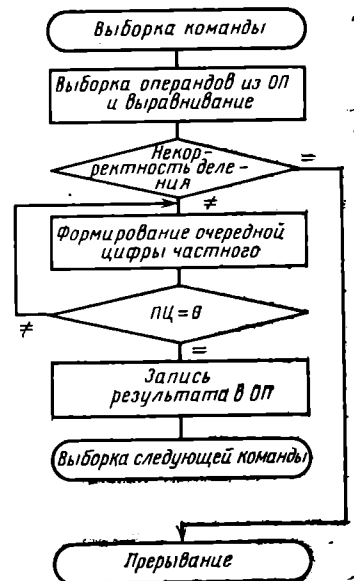


Рис. 111. Схема микропрограммы команды деления

Выборка второго операнда (рис. 111) и выравнивание его по левой границе слова осуществляются по той же микропро-

грамме, что и при умножении. Для этого перед его выборкой микропрограммно устанавливается признак сдвига влево. Выбранный операнд сдвигается и помещается в операционные регистры.

Некорректность деления обнаруживается пробным вычитанием. Для этого делитель сдвигается на одну цифру вправо (операнды были выравнены по левым границам) и вычитается из делимого. Если результат вычитания положительный, то операция деления некорректна (случай а), если отрицательный — корректна (случай б), т. е. частное поместится в отведенное для него поле.

Пример. Выполняется деление двух чисел $132 : 25 = 05 (07)$.

| | | |
|---|--|---|
| <p>а) поле делимого имеет три цифры</p> <pre> 132 250 132 — 25 ----- результат > 0 перация не корректна </pre> | <p>б) поле делимого имеет четыре цифры</p> <pre> 0132 делимое 2500 делитель 0132 — 250 делитель сдвинут на одну цифру вправо ----- результат < 0 операция корректна. </pre> | <p>операнды выровнены по левым границам</p> |
|---|--|---|

о

Процесс деления состоит в последовательном вычитании делителя из делимого с восстановлением остатка, если он получился отрицательным. После каждого вычитания, если остаток неотрицательный, формируемая цифра частного увеличивается на единицу. С целью ускорения деления область значений цифры частного разбивается на две половины: $0 \leq Ц \leq 3$ и $4 \leq Ц \leq 9$. Цикл формирования цифры частного начинается со второй половины значений. Из делимого вычитают учетверенное значение делителя и цифре частного присваивается значение 4. Если остаток неотрицательный, вычитание продолжается. При каждом неотрицательном остатке цифра частного увеличивается на единицу до тех пор, пока получится отрицательный остаток (формируется цифра частного $4 \leq Ц \leq 9$). Если же при вычитании учетверенного значения делителя получится отрицательный остаток, то это означает, что цифра частного меньше четырех. Значение делимого восстанавливается и вычитание единичного значения делителя производится, как описано выше (формируется цифра частного в пределах $0 \leq Ц \leq 3$).

Перед началом цикла деления делимое помещается в четырехсловный сдвигатель РСДО1, РСДО2, РСДВ1, РСДВ2, а единичное и четырехкратное значения делителя — в соответствующие операционные регистры. В каждом цикле деления очередная цифра частного формируется в РРД и при сдвигах влево сдвигается справа в РСДВ2. Таким образом, частное формируется в РСДВ, а остаток — в РСДО.

В начале первого цикла (рис. 112) деления делимое сдвигается на одну цифру (4 разряда) влево и в РРД заносится нуль (первая цифра частного). Одновременно по микроприказу СБР БЛ ЗП (п. 2.5) сбрасывается триггер блокировки записи информации в РСДО и РРД. Тем самым разрешается запись в РСДО, а также модификация содержимого РРД и запись в него. В дальнейшем

используется условная блокировка: по микроприказу блокировки записи в РСДО и РРД в операциях десятичного деления БЛЗПД := 1 блокируются действия следующего такта — запись остатка в РСДО и модификация содержимого РРД. Блокировка действует только в том случае, если остаток от вычитания отрицательный.

Цикл деления начинается с вычитания учетверенного делителя. При неотрицательном остатке в РРД будет занесена константа 4 (новая сформированная цифра частного). Остаток будет записан в РСДО на место делимого, и из него теперь уже будет вычитаться единичное значение делителя. После вычитания (если остаток не отрицательный), модифицируется содержимое РРД ($РРД := +1$), остаток вновь записывается в РСДО, и процесс продолжается до получения отрицательного остатка. Если после вычитания учетверенного делителя остаток будет отрицательным, по сигналу БЛЗПД := 1 будут заблокированы занесение константы 4 в РРД и запись в РСДО отрицательного остатка от этого вычитания; в РСДО сохранится прежнее значение остатка (делимого). Микропрограмма перейдет на ветвь вычитания единичного значения делителя.

Когда после вычитания единичного значения делителя будет получен отрицательный остаток, по сигналу БЛЗПД := 1 будут заблокированы модификация РРД (т. е. это вычитание не вызовет изменения цифры частного) и запись в РСДО остатка от этого вычитания. Формирование очередной цифры частного заканчивается. Последний неотрицательный результат и оставшаяся часть делимого сдвигаются в РСДО и РСДВ на 4 разряда (одну цифру) влево. При этом очередная цифра частного из РРД вдвигается в младшие разряды РСДВ, и далее выполняется очередной цикл деления. На рис. 113 показаны: пример расположения частного и остатка в РСДО и РСДВ для случая, когда длина остатка больше одного слова, и запись результата в регистровую память.

Пример. Выполняется деление двух чисел:

$$132 : 25 = (07) 05.$$

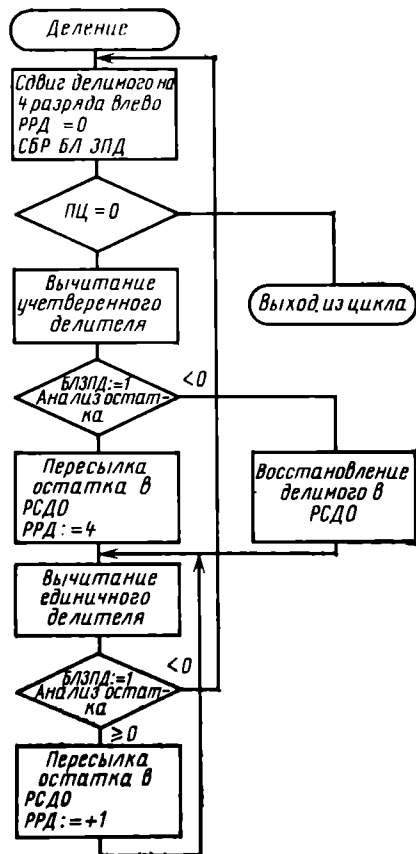


Рис. 112. Схема микропрограммы цикла деления

Проверка некорректности деления была рассмотрена ранее. Цикл деления поясняется табл. 34, где для наглядности и простоты разрядность РСДО и РСДВ сокращены с 64 разрядов до 8.

Т а б л и ц а 34

Цикл деления

| Сдвигатель (РСДО, РСДВ) | РРД | Действия |
|--|-----|--|
| 0132 | 0 | Делимое |
| 1320 | 0 | Сдвиг влево на одну цифру; РРД := 0, СБР БЛ ЗПД |
| 100 | | Вычитание 4-кратного делителя |
| 0320 | 4 | Остаток > 0; РРД := 4. пересылка остатка в РСДО |
| 025 | 5 | Вычитание делителя из остатка |
| 0070 | | Остаток > 0; РРД := +1, пересылка остатка в РСДО |
| 025 | | Вычитание делителя из остатка |
| -0180 | | Остаток < 0; БЛЗПД := 1, запись остатка в РСДО и модификация РРД заблокированы |
| 0070 ← ————— 5 0705 ————— Частное Остаток | | Сдвиг на одну цифру влево |

При записи результата операции в ОП он из сдвигателя (РСДО и РСДВ) поступает сначала в регистровую память РОП0 — РОП4 (старшее слово частного всегда в РОП0). Предварительно частное сдвигается на одну цифру влево для освобождения знаковой тетрады (см. рис. 113).

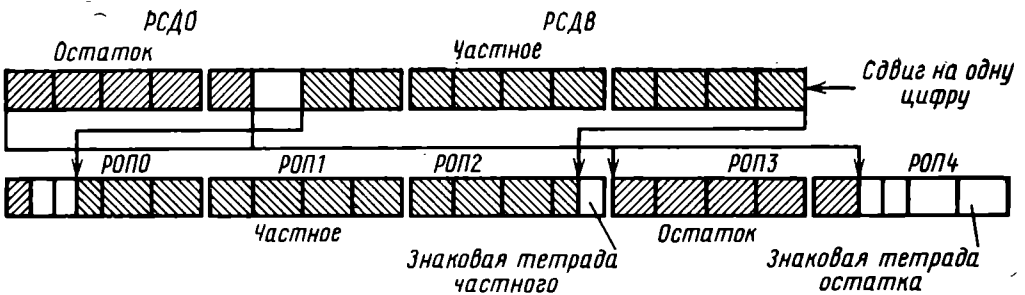


Рис. 113. Расположение результата деления в регистровой памяти

Знак результата помещается в знаковую тетраду частного, знак делимого — в знаковую тетраду остатка. При записи в ОП частное вместе с остатком выравниваются по крайней левой границе первого операнда. Запись производится с использованием механизма переадресации операционных регистров (п. 2.3). Крайние слова записываются с помощью аппарата записи байт (п. 3.4).

9.4. УПАКОВКА, РАСПАКОВКА, ПЕРЕСЫЛКА

Особенность этих команд состоит в том, что по принципам работы ЕС ЭВМ поля операндов могут перекрываться. Поэтому их обработка должна вестись так, как будто каждый байт результата записывается в память после того, как были выбраны необходимые байты операнда. Это требует большого количества обращений в оперативную память, равного удвоенному числу байтов результата.

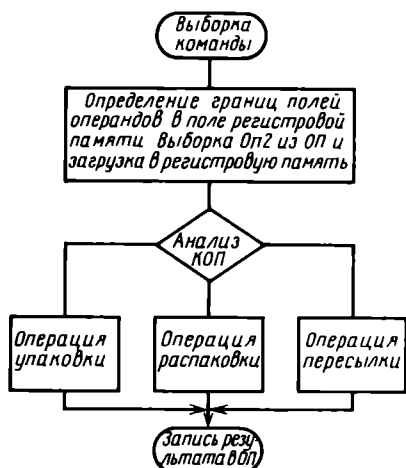


Рис. 114. Схема алгоритма выполнения команд УПАКОВАТЬ, РАСПАКОВАТЬ; ПЕРЕСЫЛКА СО СДВИГОМ

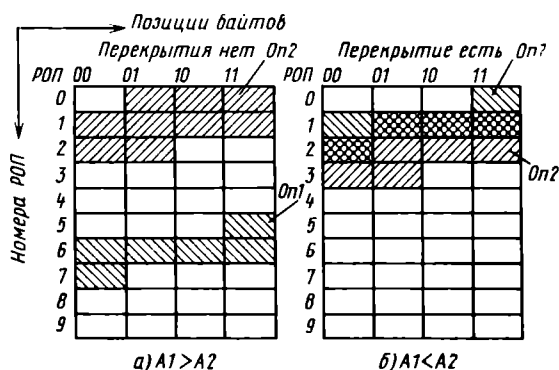


Рис. 115. Пример расположения операндов в регистровой памяти

С целью уменьшения числа обращений в ОП в регистровой памяти процессора моделируется расположение полей операндов в ОП (рис. 114). При этом поле каждого операнда занимает несколько операционных регистров с последовательно возрастающими номерами так, чтобы позиции соответствующих байтов операнда в ОП и регистровой памяти совпали. Если поля операндов в ОП перекрываются, то они перекрываются и в регистровой памяти. Второй операнд выбирается из ОП и помещается в регистровую память в отведенное для него поле. Поле первого операнда в регистровой памяти определяет поле записи результата.

Обработка операндов ведется побайтно справа налево, результат формируется в поле первого операнда в регистровой памяти. Результат из регистровой памяти записывается в ОП пословно, начиная со старшего слова.

Таким образом, в машине, во-первых, обращение к ОП за операндом и запись в нее результата производятся пословно, а не побайтно. И, во-вторых, для выполнения операции обращение за байтом операнда производится не к ОП, а к регистровой памяти, цикл обращения к которой намного меньше цикла ОП. Эти приемы позволили в значительной мере сократить время выполнения рассматриваемых команд.

Для определения расположения полей операндов в регистровой памяти вначале производится сравнение адресов $A1$ и $A2$ и длин $L1$ и $L2$ операндов.

Если $A1 = A2$, то это означает, что операнды перекрываются, начиная со старших байтов. Поля обоих операндов в регистровой памяти начинаются с РОП0, длины полей определяются величинами $L1$ и $L2$.

При $A1 > A2$ поле второго операнда в регистровой памяти начинается с РОП0 и определяется длиной операнда $L2$. Здесь возможны две ситуации: перекрытие полей может быть, а может и не быть.

В первом случае вычисляется номер РОП, определяющий конец поля первого операнда. В этот РОП будет помещено младшее слово результата, затем в РОП с номером на единицу меньше очередное, более старшее слово результата и т. д. Длина поля определяется величиной $L1$.

Во втором случае (рис. 115, а) конец поля первого операнда определяется регистром РОП8, в который будет помещено младшее слово результата.

Если же $A1 < A2$, то поле первого операнда начинается с РОП0 и определяется длиной операнда $L1$. Здесь перекрытие полей операндов также может быть или не быть.

Перекрытие есть (рис. 115, б). В этом случае определяется начало поля второго операнда, длина поля определяется длиной операнда.

Перекрытия нет. Поле второго операнда начинается с РОП5.

В результате вычисляются и размещаются в регистры следующие величины:

РОП12[30/31] — позиция крайнего правого байта первого операнда (поля результата);

РОП13[30/31] — позиция крайнего правого байта второго операнда;

СТЦ1 — номер РОП, в который будет загружаться младшее слово результата;

СТЦ2 — номер РОП, с которого начинается поле второго операнда.

Загрузка второго операнда в регистровую память и запись результата в ОП производятся по слову с использованием механизма переадресации РОП (п. 2.3). Старшее и младшее слова записываются в ОП с помощью аппарата выбора и записи байтов (п. 3.4).

Перед началом загрузки второго операнда в ТРА5 заносится константа 0.3 (рис. 116). Это означает, что в режиме включения переадресации разрешено чтение на М3 из РОП и запись с М3 в РОП, номер N которого определяется содержимым СТ2. Запись и чтение происходят по микроприказам $M3 ; = РОПN$ и $РОПN ; = M3$.

При загрузке второго операнда очередное слово, поступившее из ОП в РИП, считывается на МЗ и записывается с использованием режима переадресации в РОП, номер N которого также определяется содержимым СТЦ2. Одновременно содержимое СТЦ2 увеличивается на единицу, адрес в РАП увеличивается на четыре байта и дается запрос к ОП за очередным словом операнда ($ТОЧП \neq 0$, $МЗ := РИП$, $РАП := +4$, $ВКЛ ПА \parallel РОПN := МЗ$, $СТЦ2 := +1$, $ЗПРП 000$).

В конце загрузки второго операнда содержимое СТЦ2 будет соответствовать номеру Y того РОП, в котором хранится младшее слово Оп2. Содержимое СТЦ1 соответствует номеру X того РОП, в который будет загружаться младшее слово результата.

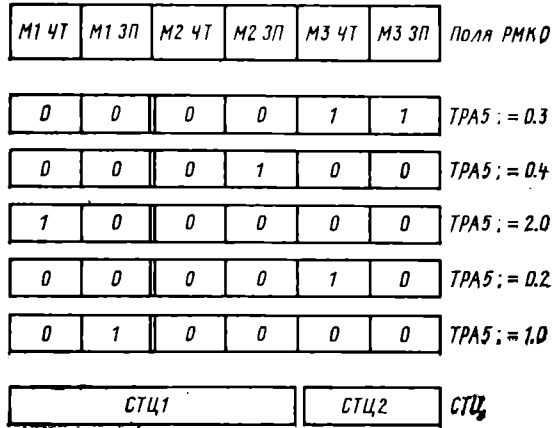
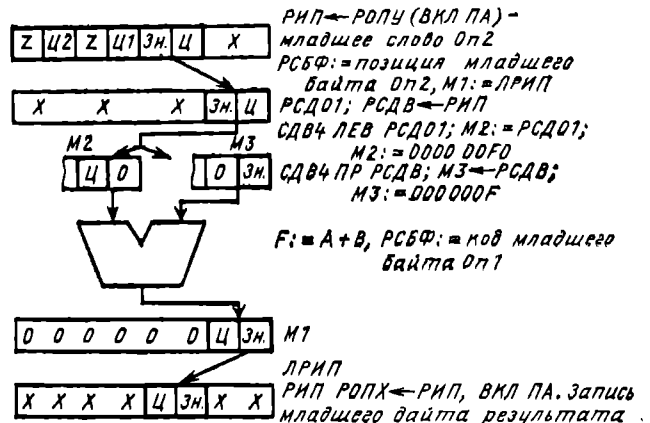


Рис. 116. Константы переадресации в операциях упаковки и распаковки

Формирование и запись в поле результата первого байта результата для операций упаковки и распаковки выполняются одинаково и заключаются в том, что цифра записывается в знаковую тетраду, а знак — в тетраду цифры (рис. 117).

Рис. 117. Формирование и запись младшего байта результата в операции УПАКОВАТЬ:

Z — зона; Ц — цифра; Зн. — знак; X — содержимое безразлично



При включении режима переадресации из РОПУ хранящееся в нем младшее слово Оп2 считывается на МЗ, а затем отсылается в РИП ($МЗ := РОПУ$, $ВКЛ ПА$, $РИП := МЗ$, т. е. $РИП \leftarrow РОПУ$). В РСБФ из РОП13[30/31] заносится код 10, определяющий позицию крайнего правого байта второго операнда. По микроприказу $М1 := ЛРИП$ выбранный таким образом байт выдается на $М1$ [24/31], а затем записывается в РСДО1 и РСДВ

(РСДО1, РСДВ ← РИП). После этого в РСДО1 операнд сдвигается на 4 разряда влево и считывается на М2 вместе с маской (СДВ4 ЛЕВ РСДО1; М2 := РСДО1; М2 := 0000 00F0). В результате этих действий на М2[24/27] выделяется последняя цифра «Ц» младшего слова Оп2. Одновременно в РСДВ операнд сдвигается на 4 разряда вправо и затем считывается вместе с маской на М3 (СДВ4 ПР РСДВ, М3 := РСДВ, М3 := 0000 000F). В результате этих действий на М3[28/31] выделяется знаковая тетрада младшего слова Оп2. После сложения в АЛУ содержимого М2

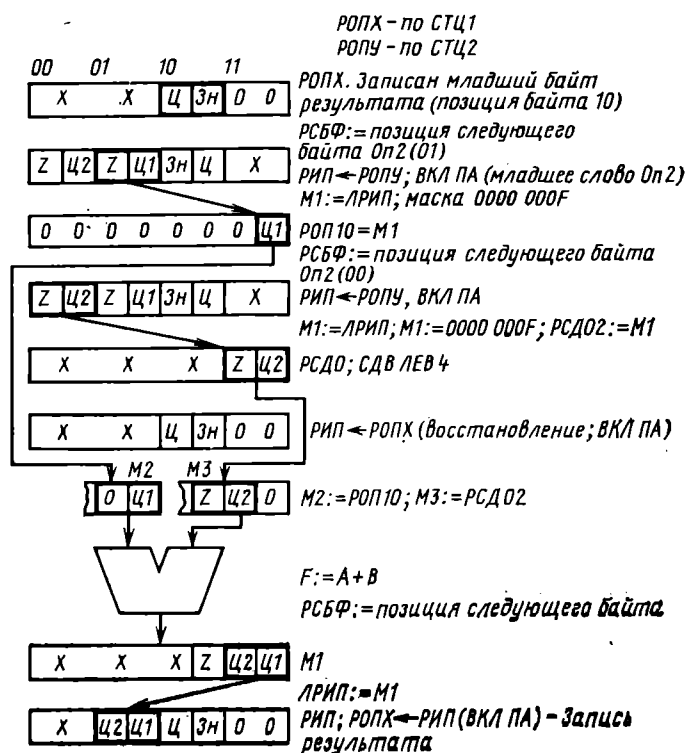


Рис. 118. Формирование очередного байта результата в операции УПАКОВАТЬ:

Z — зона; Ц — цифра; Зн. — знак; X — содержимое безразлично

и М3 на М1[24/31] будет получен младший байт результата. Он должен быть записан в позицию младшего байта в поле первого операнда (поле результата). В предыдущем такте код 10, определяющий позицию крайнего правого байта Оп1, заносится в РСБФ из РОП12[30/31] и используется для записи байта результата с М1[24/31] в выбранную позицию РИП.

При формировании очередного байта результата в операции, выполняемой по команде УПАКОВАТЬ (рис. 118), вначале в ТРА5 заносится новая константа ТРА5 := 0.4 (см. рис. 116). Она дает возможность в режиме ВКЛ ПА поместить сформированный ранее младший байт через М2 в РОП, номер X которого задан содержимым СТЦ1, т. е. в младший РОП поля результата (М2 := РИП, ВКЛ ПА; РОПХ := М2, т. е. РОПХ, РИП, см. последнее действие на рис. 117).

Для считывания на обработку текущих байтов Оп2 в ТРА5 заносится новая константа ТРА5 = 2.0 (см. рис. 116). Она позволяет в режиме ВКЛ ПА поместить в РИП очередной байт результата из того РОП, где хранится слово Оп2; номер у этого РОП определяется содержимым СТЦ2 ($M1 := РОПУ$, ВКЛ ПА, РИП := $M1$, т. е. $РИП \leftarrow РОПУ$). В РСБФ заносится код 01, определяющий номер текущего байта Оп2. Выбранный таким образом байт по микроприказу $M1 := ЛРИП$ считывается на $M1[24/31]$ вместе с маской 0F этого байта. Тем самым у считанного на $M1$ байта обнуляется тетрада зоны и сохраняется тетрада цифры. Этот операнд запоминается в РОП10 ($РОП10 := M1$), а в РСБФ заносится код 00, являющийся номером следующего байта Оп2. Снова слово Оп2 передается в РИП из РОПУ ($M1 :=$

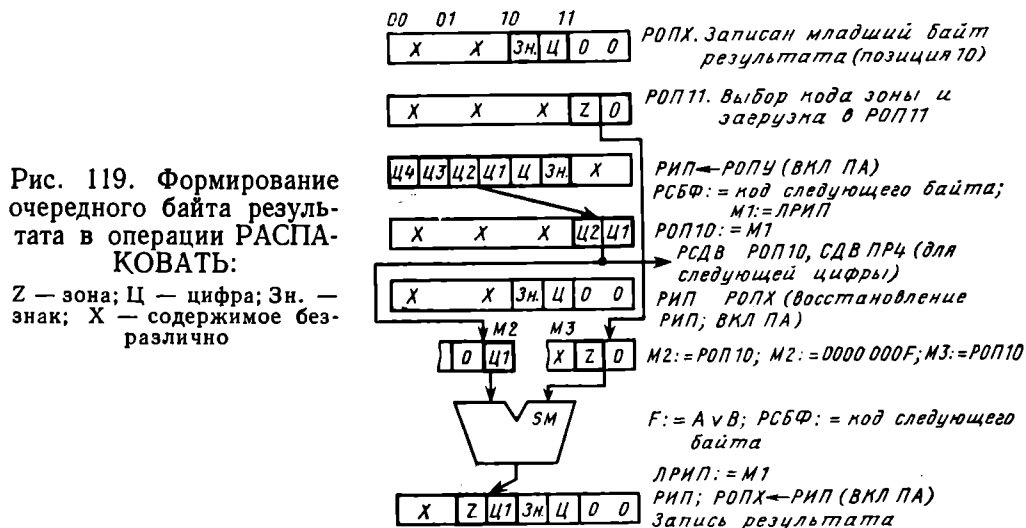


Рис. 119. Формирование очередного байта результата в операции РАСПАКОВАТЬ:

Z — зона; Ц — цифра; Зн. — знак; X — содержимое безразлично

: = РОПУ, ВКЛ ПА, РИП := $M1$) и по микроприказу $M1 :=$:= ЛРИП выделяется очередная цифра. Полученный таким образом операнд с $M1$ записывается в РСДО и сдвигается влево на четыре разряда. Это нужно для того, чтобы цифра данного обрабатываемого байта заняла позицию зоны предыдущего байта, хранящегося в РОП10. Из РОП10 и РСДО операнды считываются соответственно на $M2$ и $M3$ и складываются в АЛУ, а в РСБФ заносится код 01 позиции текущего байта Оп1. Предварительно в ТРА5 помещается константа ТРА5 := 0.2 (см. рис. 116). Она дает возможность в режиме ВКЛ ПА передать из РОПХ через $M3$ в РИП сформированный ранее результат ($РИП \leftarrow РОПХ$, восстановление $РИП: M3 := РОПХ$, ВКЛ ПА, $РИП := M3$). Младший байт результата сложения в АЛУ по микроприказу $ЛРИП :=$:= $M1$ помещается с $M1[24/31]$ в позицию РИП, код которой задан в РСБФ. Таким образом, в РИП оказались записанными два очередных байта результата. В ТРА5 снова заносится константа

ТРА5 := 0.4, которая дает возможность сформированный в РИП результат записать в младший РОП поля результата (M2 := := РИП, ВКЛ ПА, РОПХ := M2, т. е. РОПХ ← РИП).

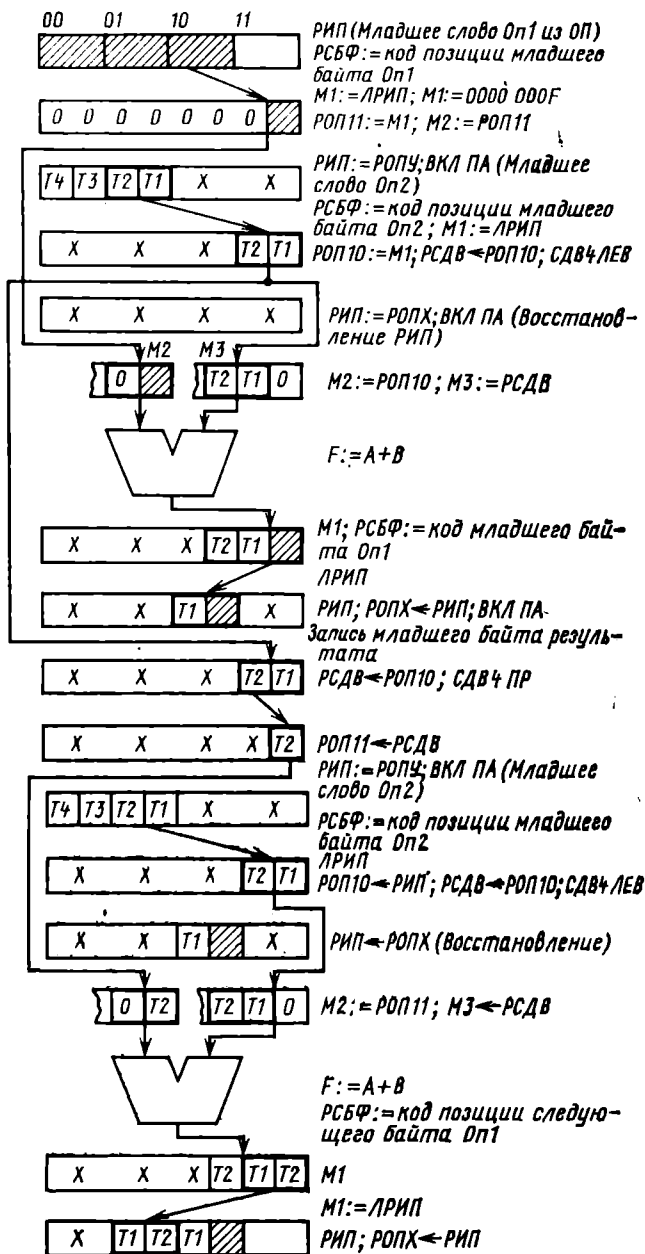


Рис. 120. Формирование первого байта и последующих тетрад результата в операции ПЕРЕСЫЛКА СО СДВИГОМ

При формировании очередного байта результата в операции, выполняемой по команде РАСПАКОВАТЬ (рис. 119), вначале в ТРА5 устанавливается константа ТРА5 := 1.0 (см. рис. 116). Она нужна для управления записью сформированного младшего

байта результата в младший РОПХ поля результата ($M1 := := \text{РИП}$, ВКЛ ПА, РОПХ $:= M1$). Затем формируется и заносится в РОП11 код зоны; значение этого кода зависит от значения РССП [12] (прил. 7). В ТРА5 помещается константа ТРА5 $:= := 0.2$, которая позволяет выбрать нужный РОП по содержимому СТЦ2. Хранящееся в этом РОП слово Оп2 переписывается в РИП ($M3 := \text{РОПУ}$, ВКЛ ПА, РИП $:= M3$, т. е. РИП \leftarrow РОПУ). В РСБФ заносится код 01 текущего байта операнда. Этот байт из РИП считывается на $M1 [24/31]$, откуда он передается в РОП10. Далее операнды из РОП10 и РОП11 считываются соответственно на $M2$ и $M3$. При этом на $M3$ считывается еще и маска и с ее помощью на магистрали обнуляется тетрада, находящаяся справа от зоны. Операнды складываются в АЛУ, а в РСБФ заносится код 01 текущего байта результата. Сформированный байт по микроприказу ЛРИП $:= M1$ помещается в выбранную позицию РИП; содержимое РИП до этого предварительно восстанавливается (ТРА5 $:= 0.8$, затем $M2 := \text{РОПХ}$, ВКЛ ПА, РИП $:= M2$). Таким образом, в РИП находятся два очередных байта результата, которые записываются в РОПХ поля результата (ТРА5 $:= 0.4$, затем $M2 := \text{РИП}$, ВКЛ ПА, РОПХ $:= M2$).

В операции, выполняемой по команде ПЕРЕСЫЛКА СО СДВИГОМ, второй операнд помещается вплотную слева к младшей тетраде первого операнда (рис. 120). Младший байт Оп1 с помощью ЛРИП считывается на $M1 [24/31]$, где с помощью маски выделяется его младшая тетрада, и помещается в РОП11. Младший байт Оп2 с помощью ЛРИП передается в РОП10, откуда он записывается в сдвигатель РСДВ, где сдвигается на четыре разряда влево. Операнды из РОП10 и РСДВ считываются на $M2$ и $M3$, и в АЛУ формируется младший байт результата. С помощью ЛРИП он заносится в позицию младшего байта в РИП, а затем — в РОПХ младшего слова результата. В следующем цикле содержимое РОП10 записывается в РСДВ, сдвигается вправо на четыре разряда и отсылается в РОП11. Младший байт Оп2 поступает в РОП10, откуда он записывается в РСДВ и сдвигается на четыре разряда влево. Далее операнды из РОП10 и РСДВ считываются на $M2$ и $M3$ и поступают в АЛУ, где формируется результат, содержащий три тетрады. С помощью ЛРИП байт с $M1 [24/31]$ помещается в РИП в позицию текущего байта результата, однако истинным сформированным результатом является правая половина байта.

Во всех операциях для выбора очередного байта второго операнда и для записи в регистровую память байта (тетрады) результата производится модификация кодов номеров байтов, которые хранятся в РОП12 и РОП13. При записи в регистровую память и чтении из нее в режиме переадресации РОП ее адрес (номер РОП) определяется по содержимому СТЦ1 и СТЦ2. При переходе границ слов содержимое СТЦ1, СТЦ2, а также коды длины операндов $L1$ и $L2$ модифицируются.

СРЕДСТВА КОНТРОЛЯ И ДИАГНОСТИКИ ЦЕНТРАЛЬНОГО ПРОЦЕССОРА

10.1. ОБЩИЕ ПРИНЦИПЫ ОРГАНИЗАЦИИ КОНТРОЛЯ И ДИАГНОСТИКИ

Средства контроля и диагностики ЦП построены в соответствии с общей концепцией их организации, принятой в ЭВМ ЕС-1033 (п. 1.4). Контрольное оборудование ЦП является составной частью его функциональных узлов и блоков. Поэтому блок контроля может быть выделен только лишь условно (см. рис. 22). Блок диагностики (БД) процессора является самостоятельным функциональным блоком и имеет автономное микропрограммное управление.

Память микропрограмм БД состоит из одного блока БОП 2048-33. Этот блок с прошивками диагностических микропрограмм называется БОП6 (п. 1.7). В нем хранятся микропрограммы всех диагностических процедур, включая тесты локализации неисправностей (ТЛН). Для диагностики используются также микрокоманды ЦП, которые хранятся в БОП0 — БОП3.

При обнаружении ошибки схемами контроля БД прерывает нормальную работу ЦП, осуществляя его диагностический останов. При этом сохраняются состояния регистров ЦП, которые имели место в момент возникновения ошибки. Это необходимо для последующей их записи в ОП. Такой процесс записи состояния в ОП в операционной системе называется логаутом. Содержимым логаута называется содержимое тех ячеек ОП, в которые была произведена запись состояния (прил. 10). Обычно ошибка обнаруживается в том такте, в котором она возникает. Исключение составляют те случаи, когда в текущем такте происходит блокировка ошибки специальным сигналом (п. 10.2). В этом случае ошибка обнаруживается в следующем такте. Диагностический останов ЦП происходит всегда в том такте, в котором была обнаружена ошибка. После записи состояния ЦП в ОП предпринимается попытка восстановления вычислительного процесса путем повторения команды, при выполнении которой обнаружена ошибка. Если команду оказывается возможным повторить, такая попытка делается семь раз. В том случае, когда команда в конце концов выполняется правильно, событие классифицируется как сбой. Если же команду с начала повторить нельзя, то классификация события производится с помощью ТЛН. Во всех случаях, когда причиной ошибки был сбой, после ее классификации БД передает управление программным средствам восстановления (гл. 14). Когда причиной ошибки был отказ в работе технических средств, БД с помощью ТЛН локализует место отказа.

При поступлении в БД сигнала ошибки ОК его состояние запоминается в ОП и управление передается программным средствам восстановления.

В процессоре используются несколько методов контроля.

Контроль по mod2 применяется для проверки правильности хранения информации в блоках ЦП, обмена информацией между блоками ЦП, а также между ЦП и ОП. Контроль дублированием используется для проверки правильности обработки информации в БАЛ. Вычисление контрольных бит по исходной информации и виду ее обработки применяется для контроля работы счетчиков РОН и РОП, узлов РРД, ПНЛ, а также РСДВ.

Выбор четного и нечетного дополнения при контроле по mod2 определяется необходимостью отличать нулевую информацию от отсутствия информации. В трактах, связывающих процессор и ОП, а также БОП с другими блоками ЦП, отсутствие информации характеризуется нулевым состоянием всех разрядов, включая контрольные. В этих цепях используется дополнение до нечетности. Во всех узлах процессора, связанных с магистралями, а также в трактах его связи с каналами отсутствие информации характеризуется единичным состоянием всех разрядов, включая контрольные. Поэтому в этих узлах и цепях связи используется дополнение до четности. В зависимости от структуры данных контрольным битом снабжается либо целое слово, либо байт, либо тетрада. Контрольные биты хранятся в триггерах контрольных разрядов и пересылаются вместе с контролируемой информацией.

Ошибки центрального процессора и оперативной памяти фиксируются в регистре РМО.

Для рассмотрения схемы контроля процессора (рис. 121) представим его в самом общем виде, состоящем из массива функциональных и логических блоков 1 — N, пронизанного тремя

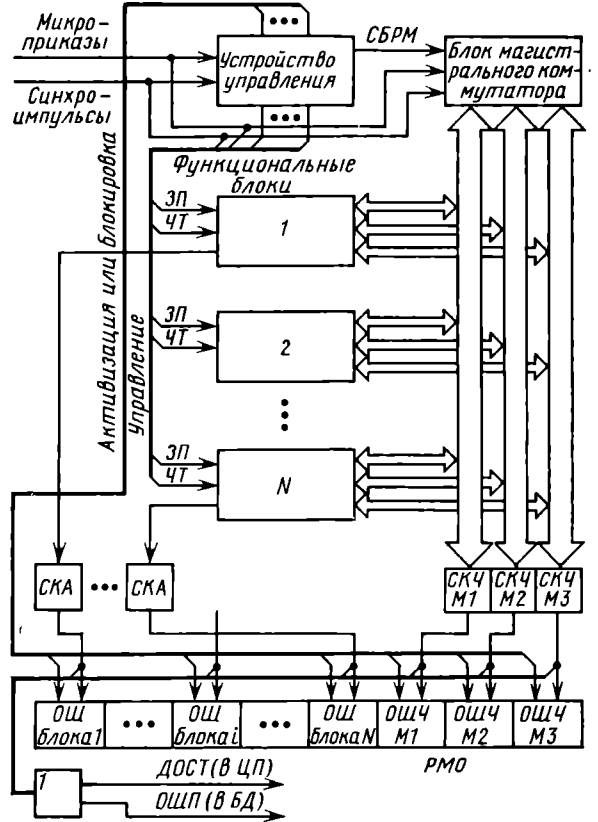


Рис. 121. Обобщенная схема контроля процессора

Точки контроля и их индикация

| Объект контроля | Идентификатор ошибки | Индикация РМО на пульте управления | Приоритет анализа РМО |
|---|---|------------------------------------|-----------------------|
| 1 | 2 | 3 | 4 |
| Пересылка 1—4 байтов информации в ОП | ОШ ИНФ 1Б — ОШ ИНФ 4Б | М1[0/3] | 4 |
| Пересылка 1—3 байтов адреса в ОП | ОШ АДР 1Б — ОШ АДР 3Б | М1[4/6] | 3 |
| СТЦ1 | ОШ.СТЦ1 | М1[7] | 10 |
| СТЦ2 | ОШ.СТЦ2 | М1[8] | 11 |
| СТ1 РОН | ОШ СТ1 РОН | М1[9] | 12 |
| СТ2 РОН | ОШ СТ2 РОН | М1[10] | 13 |
| ПКЗ | ОШ ПКЗ | М1[11] | — |
| РИ ОП | ОШ РИ | М1[12] | — |
| РА ОП | ОШ РА | М1[13] | — |
| 1—4-й байты РИП | ОШ РИП 1Б — ОШ РИП 4Б | М1[14/17] | 2 |
| 1—3-й байты РАП | ОШ РАП 1Б — ОШ РАП 3Б | М1[18/20] | 1 |
| БОП0 — БОП3 | ОШ0 — ОШ3 | М1[21/24] | 5 |
| 6—8 байты РССП | ОШ РССП [40/47] ОШ РССП [48/55] ОШ РССП [56/63] | М1[25/27] | — |
| 1—4-й байты числа на М1 | ОШ ЧМ1 1Б — ОШ ЧМ1 4Б | М1[28/31] | 17 * |
| То же на М2 | ОШ ЧМ2 1Б — ОШ ЧМ2 4Б | М2[0/3] | 18 * |
| » » М3 | ОШ ЧМ3 1Б — ОШ ЧМ3 4Б | М2[4/7] | 19 * |
| 1—4-й байты маски на М1 | ОШ ММ1 1Б — ОШ ММ1 4Б | М2[8/11] | 7 |
| То же на М2 | ОШ ММ2 1Б — ОШ ММ2 4Б | М2[12/15] | 8 |
| » » М3 | ОШ ММ3 1Б — ОШ ММ3 4Б | М2[16/19] | 9 |
| Тетрада результата БАЛ[0/3]—БАЛ[28/31] | ОШ АЛУ[0/3] — ОШ АЛУ[28/31] | М2[20/27] | 16 |
| Пересылка операнда А в БАЛ, тетрады [0/3]—[28/31] | ОШ ВХА[0/3] — ОШ ВХА[28/31] | М2[28/31] М3[0/3] | 14 |
| Пересылка операнда В в БАЛ, тетрады [0/3]—[28/31] | ОШ ВХВ[0/3] — ОШ ВХВ[28/31] | М3[4/11] | 15 |
| Пересылка 0—2 слов РМК | ОШ РМК0 — ОШ РМК2 | М3[12/14] | 6 |

Примечание: В графе 3 М1, М2, М3 — линейки индикации пульта управления; * — ошибка числа на магистрали.

магистральями, аналогично рис. 22, блока магистрального коммутатора и устройства управления, на входы которого поступают микроприказы и синхроимпульсы СИ. Устройство управления вырабатывает сигналы записи ЗП и чтения ЧТ информации, сброса магистралей СБРМ, а также сигналы активизации контроля. В соответствии с принятыми в процессоре организацией связей и концепцией контроля контрольное оборудование ЦП можно условно разделить на два вида: схемы контроля числа (СКЧ), находящегося на магистрали, и автономные схемы контроля (СКА).

Схемы контроля числа, расположенные в блоке магистрального коммутатора, позволяют обнаружить некорректность числа, считанного на магистраль из любого функционального блока. В момент считывания схемы контроля активизируются специальными сигналами и в случае некорректности числа выдают сигнал ошибки (ОШЧ) в РМО. При любом количестве блоков, связанных с информационными магистралями, всего три узла контроля числа, расположенные каждый на своей магистрали, обеспечивают непрерывный контроль всех передач информации.

На информационные магистрали может быть считана информация из любого узла или блока процессора, поэтому при наличии ошибки числа заранее нельзя определить, где она возникла. Идентификация узла или блока и локализация места ошибки производятся в процессе выполнения тестов локализации неисправности.

Автономная схема контроля охватывает один определенный функциональный узел и имеет отдельные разряды в РМО для посылки туда сигналов ошибок. Как правило, это узлы, не имеющие непосредственной связи с информационными магистралями (узел контроля регистров РИ и РА оперативной памяти, узел передачи адреса и информации в ОП, узел контроля БОП и др.), или адресные регистры (СТЦ, СТ РОИ, РАП, РССП [40/63] и др.). Ряд регистров, непосредственно связанных с магистралями (например, РАП и РИП), имеет также и автономные схемы контроля, что позволяет вести контроль как при записи, так и при чтении и способствует повышению оперативности локализации неисправностей. В табл. 35 приведены перечень контролируемых узлов процессора и распределение разрядов РМО.

10.2. РЕГИСТРАЦИЯ МАШИННЫХ ОШИБОК

Сигналы обнаруживаемых ошибок поступают для регистрации в РМО, который представляет собой 79-разрядный регистр, реализованный на RS-триггерах.

Обмен информацией между блоками ЦП сопровождается переходными процессами. Если схемы контроля будут анализировать информацию до их завершения, то могут возникнуть ложные сиг-

налы ошибок, хотя установившееся значение информации может быть правильным. Кроме того, сброшенные магистрали ЦП имеют дополнение до нечетности (п. 2.2). Поэтому, чтобы отличить достоверную информацию о наличии ошибки от ложной, прием сигналов ошибок в РМО разрешается или запрещается специальными управляющими сигналами.

Прием в РМО сигналов ошибок от схем контроля числа на магистрали производится по сигналам активизации контроля числа, поступающего на магистраль. Эти сигналы формируются схемами, представляющими собой сборки всех микроприказов чтения, по которым информация выдается на магистраль и которые стробируются сигналом с единичного выхода триггера рабочего состояния (п. 3.5). При наличии одного из сигналов чтения, допустим на М1, эти схемы вырабатывают четыре сигнала активизации для каждой из 8-разрядных групп М1. Аналогично построены схемы формирования сигналов активизации контроля М2 и М3.

Описанный способ контроля числа, считанного на магистраль, кроме обнаружения ошибок в информации дает возможность обнаруживать также ошибки в цепях управляющих сигналов. При неисправности в цепи управления чтения на магистраль не будет, и магистраль, которая была сброшена в конце предыдущего такта, сохранит свое состояние и в текущем такте. Но так как сброшенная магистраль имеет некорректное дополнение, а схемы контроля активизируются микроприказами чтения, то последние обнаружат некорректность числа на магистрали и сформируют сигнал ошибки.

Сигналы активизации контроля пересылки информации и адреса в ОП вырабатываются по управляющим сигналам из БУ ОП в том случае, если с ОП работает процессор (ТРАБП = 1, п. 3.4) и в данный момент действительно осуществляется пересылка информации.

Прием в РМО сигналов ошибок от тех блоков ЦП, в которых смена информации всегда происходит по С1 или С2 и, следовательно, переходные процессы имеют место всегда в одно и то же время относительно начала машинного такта, происходит по синхросериям С1' или С2' (С2'БЛ).

Ряд узлов и блоков ЦП может изменять свое содержимое как по синхросерии С1, так и по синхросерии С2, поэтому переходные процессы могут возникать в разное время. В этих случаях для стробирования выбирается такая синхросерия, с которой совпадают переходные процессы большинства действий. Прием сигналов ошибок от прочих действий в текущем такте блокируется и производится только в следующем такте. Например, содержимое РАП может изменяться по микроприказам: РАП := +2 (по С1), РАП := +4 (по С1), РАП := М1 (по С2), РАП := М2 (по С2), РАП := РССП[40/63] (по С1) и РАП := РССП[40/63] + 2. Последний микроприказ выполняется в два приема: по С1 происходит передача РАП := РССП[40/63] и по С2 модификация

РАП : = +2. Прием в РМО сигналов ошибок РАП стробируется синхросерией С1. По этой серии происходит прием в текущем такте сигналов ошибок, обнаруживаемых при выполнении двух первых микроприказов, а в следующем такте — двух следующих. Прием сигналов ошибок, возникающих при выполнении двух последних микроприказов, в текущем такте блокируется. Их прием осуществляется также в следующем такте.

Если хотя бы одной из схем контроля будет выработан сигнал ошибки, формируются два диагностических сигнала — диагностический останов (ДОСТ) и ошибка процессора (ОШП). Условия их формирования одинаковы, однако функциональное назначение различно. Первый поступает в БРР и останавливает нормальную работу ЦП путем прекращения выборки микрокоманд из БОП0 — БОП3 процессора. Второй активизирует работу БД. Формирование этих сигналов блокируется, если ошибки формируются при выполнении блоком БД диагностических процедур, например ТЛН, а также при нормальной работе процессора, когда на ПУ нажата клавиша «БЛ. Д» (п. 3.6).

Ошибки в РМО не фиксируются, если присутствует программная маска прерываний от схем контроля: $ССП[13] = 0$, а также если их прием заблокирован аппаратно при выполнении микропрограммы сброса системы (п. 4.2) или из диагностической микропрограммы, например записи в ОП состояния машины в момент возникновения ошибки.

10.3. ДИАГНОСТИЧЕСКИЕ ПРОЦЕДУРЫ

Блок диагностики выполняет следующие процедуры:
управляет записью в ОП состояний ЦП, регистров РА и РИ ОП и общего канала ОК, имеющих место в момент возникновения ошибки;

классифицирует причину ошибки (сбой или отказ);
подготавливает процессор к прерыванию от схем контроля;
локализует место отказа с помощью ТЛН;
выполняет команду ДИАГНОСТИКА в процессоре.

Этот блок может быть активизирован с ПУ или следующими сигналами: ОШП — при возникновении ошибок в ЦП и ОП (рис. 122), АН38 — при ошибках в ОК и АН54 — при микропрограммой передаче управления от ЦП к блоку БД (см. табл. 17). При этом формируется адрес диагностической микрокоманды (АДМК), с которой начинается выполнение соответствующей диагностической процедуры. Дальнейшая работа БД состоит в поочередной выборке из БОП6 диагностических микрокоманд (ДМК) и выполнении предписанных ими действий.

В общем случае сигнал ошибки может быть сформирован схемами контроля как при нормальной работе ЦП, так и при выполнении процедур блоком диагностики, т. е. при работе процессора под управлением блока диагностики. Поэтому при возникновении

ошибки производится аппаратная проверка на наличие условий, разрешающих фиксацию ее в РМО. В результате ошибка либо игнорируется и продолжается нормальная работа ЦП, либо фиксируется в РМО. Если ошибка возникла при выполнении ТЛН, продолжается диагностическая микропрограмма, если при нормальной работе ЦП, формируются сигналы ДОСТ и ОШП и произ-

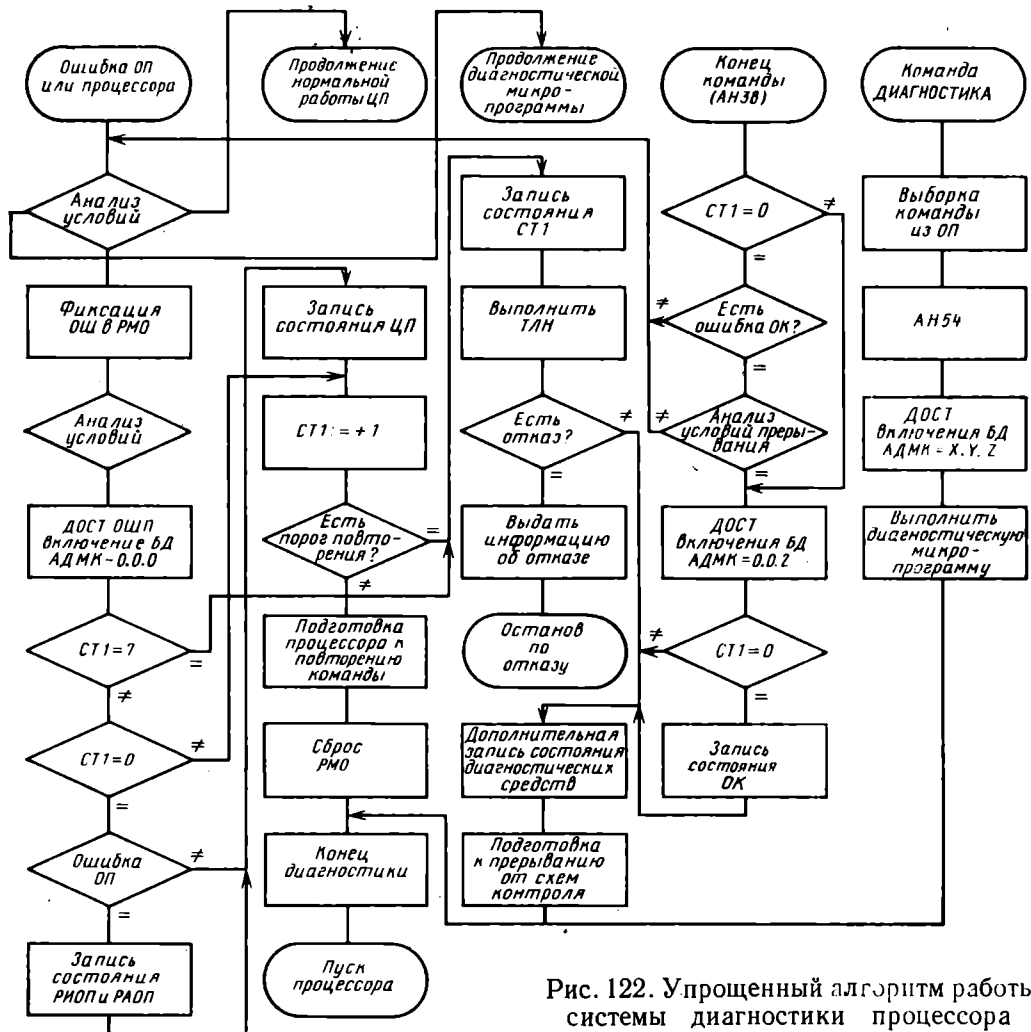


Рис. 122. Упрощенный алгоритм работы системы диагностики процессора

водится останов ЦП. По сигналу ОШП формируется адрес ДМК 0.0.0 и начинает работу микропрограммная диагностика. Дальнейшее рассмотрение алгоритма будет относиться только к случаю, когда ошибка возникла при нормальной работе ЦП, т. е. при выполнении команд процессором.

Если после последнего сброса системы или последнего прерывания от схем контроля зафиксированная в РМО ошибка — первая ($СТ1 = 0$), происходит запись состояния машины в ОП. Запись осуществляется в постоянно распределенные ячейки памяти (прил. 10). Запись состояния РИ и РА ОП выполняется только

при ошибках ОП, запись состояния ЦП — как при ошибках ОП, так и при ошибках ЦП. Диагностический счетчик СТ1 фиксирует число ошибок, которые обрабатываются блоком диагностики. Первая ошибка — ошибка, возникшая при работе ЦП под управлением программы, последующие ошибки — ошибки, возникшие при повторении команды, в процессе выполнения которой произошел сбой. Запись состояния выполняется только после первой ошибки. Запомненная при этом информация является исходной для программных средств восстановления (гл. 14), а также может быть использована для статистики ошибок.

Классификация причины ошибки (сбой или отказ) осуществляется либо семикратным повторением команды, при выполнении которой возник сигнал ошибки, либо путем выполнения тестов локализации неисправностей ТЛН в зависимости от того, пройден или нет порог повторения.

Порог повторения является условным понятием. Если на момент ошибки процессор обрабатывал прерывания или ожидания, то вопрос о повторении команды не имеет смысла. Здесь есть смысл говорить о повторении подпрограммы, допустим, ожидания. Но ошибка может исказить исходное состояние процессора и повторение подпрограммы будет происходить в условиях, отличных от тех, при которых была начата данная подпрограмма. Поэтому при обработке прерываний и ожиданий считается, что порог повторения пройден, и это следует понимать как невозможность повторения подпрограммы сначала.

Если ошибка возникла при выполнении микропрограммы команды процессора, команду можно повторить сначала при определенных условиях. Во-первых, зафиксированная ошибка не должна быть связана с РССП. В противном случае повторение команды будет происходить при ином, нежели в первоначальном случае, значении ССП, что, естественно, может привести к иному результату. Во-вторых, зафиксированная ошибка не должна быть связана с адресными регистрами РОН (СТ РОН), так как при повторении команды могут быть искажены адреса операндов как в регистровой памяти РОН, так и в ОП. В-третьих, если ошибка возникла после того, как произошла запись продвинутого адреса команды из РССП [40/63] в РАКБ, адрес текущей команды оказывается утраченным. В процессе выполнения команды в РАКБ хранится адрес текущей команды, а в РССП [40/63] — этот же адрес, продвинутый на этапе выборки, т. е. адрес следующей команды. Для повторения команды сначала текущий адрес из РАКБ переписывается в РССП [40/63] (п. 3.1). Смена адреса в РАКБ адресом следующей команды из РССП [40/63] происходит в конце выполнения текущей команды перед началом выборки следующей по микроприказу АНЗ8 (п. 4.4). Поэтому, если ошибка связана с РССП, СТ РОН или произошла смена адреса в РАКБ, порог повторения считается пройденным. И наконец, команду можно повторить сначала,

если не пройден собственно ее порог повторения. Собственно порогом повторения команды является та стадия ее выполнения, при которой исходные данные еще не утрачены. Он считается пройденным в том случае, если один из операндов частично или полностью замещается результатом операции.

Анализ прохождения порога повторения выполняется специальной диагностической микропрограммой. Если порог повторения не пройден, т. е. команду повторить можно, содержимое СТ1 увеличивается на единицу и процессор подготавливается к ее повторению, после чего управление передается программе. Повторение команды заключается в ее повторной выборке и выполнении. Если после k -й попытки ($СТ1 = k, k < 7$) команда выполнялась до конца, по сигналу АН38 вновь активизируется блок диагностики для подготовки процессора к прерыванию от схем контроля. Цель этого прерывания — сообщить операционной системе о том, что в процессоре был сбой. После этого система продолжает выполнение программы путем выборки следующей команды.

Если порог повторения пройден, то классификация ошибки производится выполнением тестов локализации неисправности. В том случае, если ТЛН не обнаружат отказ, событие классифицируется как сбой и производится прерывание от схем контроля. Операционная система принимает решение о возможности продолжения программы.

При прерываниях от схем контроля осуществляется дополнительная запись в ОП состояния диагностических средств (адреса ОП 1A0₁₆ и 19C₁₆). Эта информация идентифицирует причину, вызвавшую прерывание (табл. 36), и используется программными средствами восстановления (гл. 14).

Т а б л и ц а 36

Дополнительная запись состояния диагностических средств при различных событиях

| Адрес слова ОП | Разряды слова ОП | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|----------------|------------------|-----|-----|----------|----------|----------|-----|-----|-----|
| 1A0 | 1/3 | 000 | 000 | 001 | 010 | 100 | 101 | 011 | 110 |
| | 5/9 | X | X | X | X | <i>m</i> | X | X | X |
| | 10/12 | 000 | 000 | 000 | 000 | 000 | X | 000 | 000 |
| 19C | 10/12 | X | X | <i>n</i> | <i>k</i> | X | X | 001 | 111 |

Примечание: k — количество попыток повторения команды, $1 \leq k < 7$; m — номер теста, обнаружившего отказ РПЗ; n — количество повторений команды перед выходом на ТЛН ($n = 1$ — порог повторения пройден, $n = 7$ — не пройден); X — информация игнорируется; 1 — ошибка ОК; 2 — сбой ЦП или ОП при чтении (команда повторялась); 3 — сбой ЦП (прошли ТЛН); 4 — сбой диагностического оборудования; 5 — отказ РПЗ; 6 — сбой РБСП (прошли ТЛН); 7 — ошибка ОП при записи (порог повторения команды пройден); 8 — отказ ОП при чтении (порог повторения команды не пройден).

Если после семикратного повторения команды ошибка сохраняется, то событие классифицируется как отказ и для локализации места его возникновения выполняются ТЛН. В любом случае, если ТЛН обнаружат отказ, на ПУ выдается сообщение о нем и происходит останов процессора.

В конце любой команды всегда по микроприказу АН38 производится анализ содержимого СТ1 и наличие запомненного в отдельном триггере сигнала ошибки общего канала. Если при выполнении текущей команды не было ошибок и, следовательно, не было попыток ее повторения, а также не было ошибок ОК, то продолжается нормальная работа ЦП путем выборки следующей команды. По микроприказу АН38 все эти действия выполняются за один такт.

Работа БД может быть активизирована по сигналу АН38 в случае, если СТ1 $\neq 0$ или если при выполнении текущей команды была зафиксирована ошибка ОК. Тогда аппаратурно формируются сигнал ДОСТ и адрес ДМК 0. 0. 2 и начинает работу микропрограммная диагностика. Если СТ1 $\neq 0$ (СТ1 = k), то это означает, что текущая команда выполнена после k -й попытки ее повторения. БД осуществляет дополнительную запись в ОП состояния диагностических средств и, как было сказано выше, подготавливает ЦП к прерыванию от схем контроля. В случае ошибки ОК блок диагностики осуществляет запись в ОП состояния ОК (адреса ОП 1А4₁₆—1В4₁₆, прил. 10), после чего следует прерывание от схем контроля. Операционная система пытается восстановить работоспособность ОК.

Работа БД может быть активизирована микропрограммным путем. Заранее в микропрограмме процессора формируется код адреса начальной микрокоманды диагностической микропрограммы. По микроприказу АН54 этот адрес выдается на адресные шины БОП6 и управление передается блоку диагностики. После этого начинает выполняться выбранная диагностическая микропрограмма. Для обратного перехода в диагностической микропрограмме формируется адрес микрокоманды процессора. По микроприказу КОНЕЦ ДИАГНОСТИКИ происходит передача управления от блока диагностики процессору, после чего будет выполняться микропрограмма процессора, начиная с выбранной микрокоманды.

Работа БД может быть активизирована по сигналу ПУСК в момент нажатия на ПУ кнопки «ПУСК», если предварительно была нажата клавиша «ДИАГ» и на РА ПУ был набран код адреса ДМК (п. 3. 6).

10.4. МИКРОПРОГРАММНОЕ УПРАВЛЕНИЕ В БЛОКЕ ДИАГНОСТИКИ

В общем случае, для выполнения диагностических процедур, например записи в ОП состояния ЦП, поиска места неисправности и др., необходимо решить две основные задачи: выполнить

оборудованием процессора некоторую вполне определенную последовательность действий, задаваемую диагностической микропрограммой, и провести анализ их результата. В первом случае требуется работа основного оборудования процессора под управлением блока диагностики. Во втором — автономная работа диагностического оборудования. Это два крайних случая. Обычно же для решения этих задач работают последовательно то основное оборудование процессора под управлением блока диагностики, то автономно система диагностики.

Для обеспечения обоих указанных режимов работы системы диагностики в традиционных схемах ее построения микрокоманды процессора и диагностические микрокоманды либо выбираются в один и тот же регистр микрокоманд из одного или разных блоков памяти, либо хранятся в отдельных блоках памяти и выбираются каждый в свой регистр микрокоманд. В первом случае можно считать, что оба вида микрокоманд принадлежат одному блоку памяти, даже если блоки физически разделены. При этом поля микрокоманд, предназначенные для микроприказов диагностики, не используются при выполнении микрокоманд процессора. И, наоборот, при выполнении микрокоманд диагностики малоэффективно используются поля, предназначенные для микроприказов процессора. Особенно это заметно при существенно различных форматах микрокоманд процессора и системы диагностики. Следовательно, в целом неэффективно используются объем памяти, предназначенный для хранения управляющей микропрограммной информации, и оборудование дешифрации полей микрокоманд. Во втором случае оборудование процессора должно иметь двойное управление — со стороны процессора и со стороны системы диагностики, что усложняет аппаратуру и затрудняет поиск неисправности, однако позволяет более эффективно использовать объем памяти микропрограмм.

В ЭВМЕС-1033 микрокоманды процессора и блока диагностики имеют разный формат и хранятся в отдельных блоках памяти (п. 1.7). При нормальной работе ЦП происходят выборка микрокоманд из блока памяти микропрограмм процессора (БОП0—БОП3) и их исполнение. Когда управление передается блоку диагностики, то выборка микрокоманд из блока памяти микропрограмм процессора прекращается и начинается выборка и исполнение диагностических микрокоманд из БОП6. В процессе выполнения диагностической микропрограммы возникает необходимость в том, чтобы оборудование ЦП выполнило определенное действие (действия), например, считывание информации из регистра на магистраль, модификацию содержимого счетчика и т. д. С этой целью блок диагностики производит выборку в РМК0—РМК3 какой-либо микрокоманды процессора, в которой содержится код (коды) этого действия (действий). Путем выборочной активизации дешифраторов того поля микрокоманды процессора, в котором это действие закодировано, происходит формиро-

вание необходимого микроприказа и выполнение по нему соответствующего действия. Адрес микрокоманды процессора и код диагностического микроприказа, который осуществляет активизацию дешифратора соответствующего поля, содержатся в диагностической микрокоманде [А. с. 613651 (СССР)]. Такой прием в принципе дает возможность выполнить под управлением блока диагностики любой из 282 микроприказов процессора при относительно небольшой разрядности диагностической микрокоманды. Существенным обстоятельством при этом является тот факт, что отсутствует двойное управление оборудованием процессора

| Биты | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | |
|------|---|-----------------------------|---|---|---|---|---|---|---|---|----|-----|----|----|----|----|----|----|----|----|-------|-----|-----|---------|----|----|----|----|----|----|-------------|----|----|--|
| ДМК1 | У | Адрес следующей ДМК | | | | | | | | | | БСМ | А | | | | | Б | | МД | Адрес | | | | | | | | | | Контрольный | | | |
| ДМК2 | У | Базовый адрес следующей ДМК | | | | | | | | | | БСМ | В | | | | | | | | | | РАИ | N теста | | | | | | | | | | |
| | | Е | | | | | | | | | | | | | | | | | | | | | | РАИ | Ж | | | | | | | | | |
| | | | | | | | | | | | | З | | | | | | | | | | РАИ | И | | | | | | | | | | Д | |
| Г | | | | | | | | | | Г | | | | | | | | | | | | | Д | | | | | | | | | | | |

Рис. 123. Распределение полей диагностических микрокоманд:

У — управляющий бит (задание типа ДМК); БСМ — бит управления сбросом магистралей; А, Б — поля кодов активизации дешифраторов полей команд процессора; В, Д — поля кодов диагностических микроприказов; Г — поля кодов диагностических анализов; РАИ — разрешение анализа (модификации бита 10 кода адреса следующей ДМК при условном ветвлении); Е — биты 0/9 кода адреса следующей ДМК при условном ветвлении; Ж — биты 0/8 кода адреса следующей ДМК при безусловном ветвлении загружаются в РАВД; З — биты 9/10 кода адреса следующей ДМК при безусловном ветвлении (считываются при чтении РАВД); И — биты 15/21 кода адреса ОП при логауте; РЛ — разрешение логаута (записи состояния)

и все микрооперации в процессе работы блока диагностики выполняются по тем же микроприказам и по тем же цепям управления, что и в обычном режиме работы процессора.

В БД применяются два типа диагностических микрокоманд, ДМК1 и ДМК2 (рис. 123), различающихся по своему функциональному назначению. Управление оборудованием процессора путем вызова микрокоманд из БОПО—БОПЗ осуществляет ДМК1. Команды типа ДМК2 предназначены для управления работой диагностического оборудования и для управления записью в ОП состояния машины. Оба типа микрокоманд различаются значением нулевого разряда У, который является управляющим.

Микропрограммное управление работой диагностического оборудования с помощью ДМК2 имеет в своей основе традиционную схему (рис. 124). Базовый адрес следующей микрокоманды (разряды [1/11], см. рис. 123) может быть модифицирован диагностическим аппаратом анализов (на рис. 124 не показан). Модификации по результату анализа, задаваемого полем Г, подвергается только один младший разряд адреса. Дешифратор поля Г активизируется полем разрешения анализа РАИ. Кроме условного имеется еще и безусловное ветвление по адресу возврата.

Код этого адреса предварительно записывается из поля *И* в диагностический регистр адреса возврата (РАВД). При ветвлении по адресу возврата адрес из РАВД дополняется двумя младшими разрядами, которые поступают из зоны *З* (см. рис. 123) базового адреса следующей ДМК. Тем самым по одному адресу

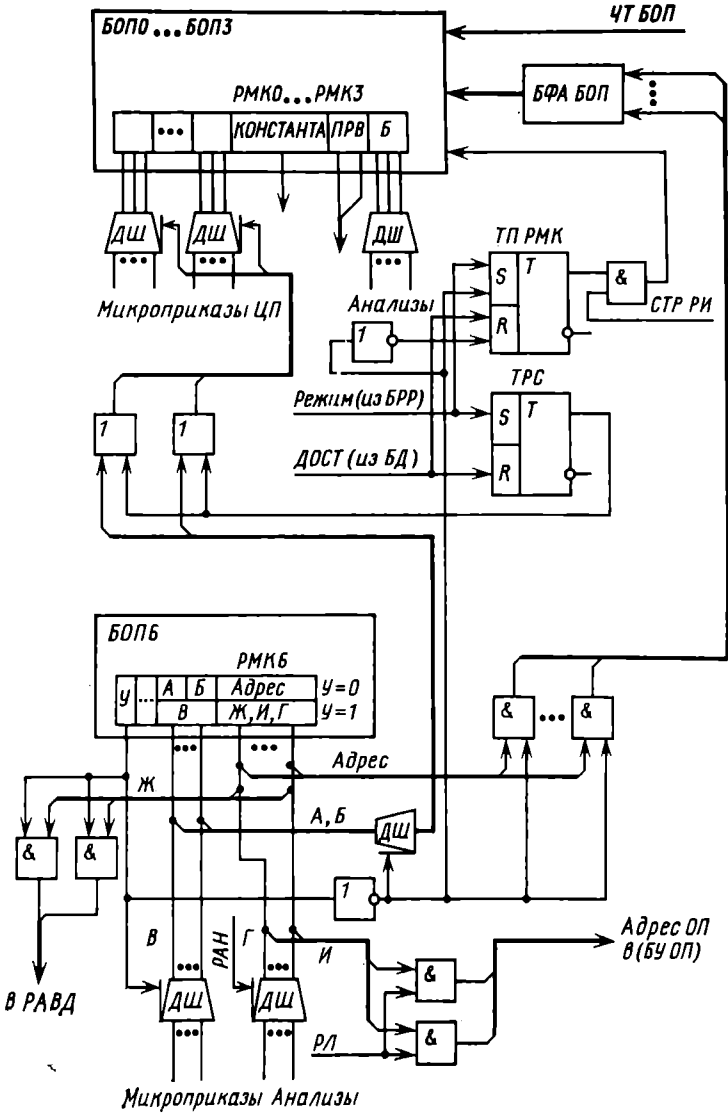


Рис. 124. Взаимодействие микропрограммного управления БД и ЦП

возврата возможно безусловное ветвление по четырем направлениям. В поле *В* закодированы диагностические действия, в том числе и те, которые обеспечивают выполнение записи в ОП состояния машины в момент обнаружения ошибки. В последнем случае адрес ОП, по которому производится запись состояния того или иного регистра или информационной магистрали, из поля *И* поступает в БУ ОП. Передача адреса строится единственным состоянием поля *РЛ* (разрушение логаута). Поскольку

для записи состояния используются ячейки ОП, коды адресов которых кратны 4 (запись идет полным словом), а наибольшим является адрес 1:В.8 (прил. 10), то 2 младших и 15 старших разрядов кода адреса всегда нулевые. Формирование полного кода адреса путем дополнения нулей слева и справа происходит в БУ ОП. Запись состояния РИ ОП и РА ОП производится по фиксированным адресам 1ФЯ и 2ФЯ (п. 1.10). В разрядах [21/31] задается номер теста, обнаружившего неисправность, если произошел останов по отказу.

Взаимодействие между микропрограммным управлением БД и ЦП осуществляется нетрадиционно, на основе использования команд типа ДМК1.

При обычном выполнении микропрограмм процессором установленный в «1» ТП РМК (см. рис. 60) разрешает выборку микрокоманд из БОП0—БОП3, а установленный в «1» ТРС через схемы ИЛИ активизирует дешифраторы операционных полей микрокоманд. При этом вырабатываются микроприказы, управляющие работой ЦП. При обнаружении ошибки схемами контроля БД формирует сигнал ДОСТ (п. 10.1), который сбрасывает ТП РМК и ТРС, тем самым прекращая выборку микрокоманд ЦП и выработку микроприказов дешифраторами. Происходит останов ЦП. Если при выполнении диагностической процедуры возникает необходимость в том, чтобы ЦП выполнил какую-то микрооперацию, то в микропрограмме этой процедуры используется микрокоманда типа ДМК1. При этом из поля АДРЕС ДМК1 код адреса микрокоманды процессора поступает на входы группы схем И. На другие входы этих схем подается сигнал из управляющего поля У. В результате код адреса пересылается в БФА БОП, с выхода которого он поступает на адресные входы БОП0—БОП3. Одновременно сигнал из управляющего поля устанавливает в «1» ТП РМК, разрешая тем самым выборку микрокоманды из БОП0—БОП3. Микрокоманда ЦП по адресу, заданному в ДМК1, выбирается в РМК0—РМК3. Вместе с этим содержимое полей А и Б ДМК1 поступает на входы дешифратора, работа которого стробируется сигналом из управляющего поля. Дешифратор вырабатывает сигналы активизации того поля микрокоманды ЦП, в котором закодирован необходимый для исполнения микроприказ. Эти сигналы через схемы ИЛИ поступают на входы дешифратора соответствующего поля микрокоманды процессора. В результате вырабатывается микроприказ, закодированный в этом поле, который поступает в ЦП. Последний выполняет необходимую микрооперацию. Если в диагностической микропрограмме встречается последовательность команд ДМК1, то ЦП выполняет последовательность микроопераций, предписанных диагностической микропрограммой. При смене команд ДМК1 на ДМК2 сигнал из управляющего поля сбрасывает ТП РМК, и последняя, вызванная диагностикой микрокоманда ЦП сохраняется в РМК0—РМК3 до следующей команды ДМК1.

Дешифратор поля анализов (поля *Б*) микрокоманды ЦП активизируется постоянно. Это позволяет с помощью команды ДМК1 выполнить любой из анализов ЦП (см. табл. 17—19), а затем с помощью команды ДМК2 проанализировать состояние любого разряда МАН процессора.

Команда ДМК1 выполняется за два машинных такта. В первом такте выбирается в РМК6 диагностическая микрокоманда и расшифровываются коды ее полей. Во втором такте выбирается в РМК0—РМК3 микрокоманда ЦП и выполняется активизированный из ДМК1 микроприказ. При графической записи микропрограммы тип диагностической микрокоманды (ДМК1 или ДМК2) указывается над левым верхним углом ее блочного символа, а над правым верхним — адрес. В поле блочного символа микрокоманды ДМК1 указывается микроприказ ЦП, подлежащий выполнению, адрес БОП микрокоманды ЦП, содержащей этот микроприказ, и диагностический микроприказ активизации дешифратора поля, в котором закодирован микроприказ ЦП. В поле блочного символа микрокоманды ДМК2 указываются диагностические микроприказы или проводимые анализы.

В качестве примера рассмотрим фрагмент микропрограммы записи состояния ЦП, а именно запись состояния РССП (рис. 125). Это одно из самых простых действий. Суть его заключается в том, что содержимое РССП необходимо переслать в РИП, после чего записать в ОП.

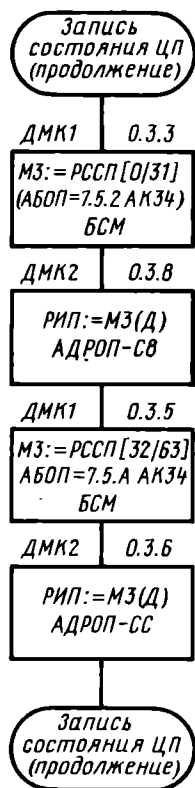


Рис. 125. Фрагмент микропрограммы записи состояния ЦП

Пересылка в РИП осуществляется через МЗ. Для считывания первого слова РССП на МЗ используется микроприказ процессора — $MЗ := РССП[0/31]$. Команда типа ДМК1 (0. 3. 3) вызывает в РМК0—РМК3 по адресу АБОП-7.5.2 микрокоманду ЦП и по диагностическому микроприказу АКЗЧ (закодирован в поле *А* ДМК1) активизирует поле ЗЧТ чтения на магистраль МЗ (см. рис. 61), где закодирован этот микроприказ ЦП. После считывания информации на МЗ сброс магистрали блокируется по микроприказу блокировки сброса магистралей БСМ. Следующей командой типа ДМК2 (0. 3.8) осуществляется запись в РИП с МЗ по диагностическому микроприказу $РИП := МЗ(Д)$. Адрес оперативной памяти АДР ОП, код которого С8, из поля АДРЕС поступает в БУ ОП. Одновременно поле *РЛ* формирует диагностический запрос ЗПРД (см. рис. 55) в ОП для записи информации, находящейся в РИП. В конце такта магистрали сбрасываются. Запись второго слова РССП происходит аналогично.

10.5. ТЕСТЫ ЛОКАЛИЗАЦИИ НЕИСПРАВНОСТИ

Переход на выполнение ТЛН осуществляется либо после классификации ошибки, когда в результате семикратного неуспешного повторения команды установлено, что ошибка вызвана отказом, либо при невозможности повторения команды с начала.

Исполнению ТЛН всегда предшествует анализ всех разрядов РМО (рис. 126). Он производится по определенному приоритету, установленному в порядке значимости тех или иных блоков для

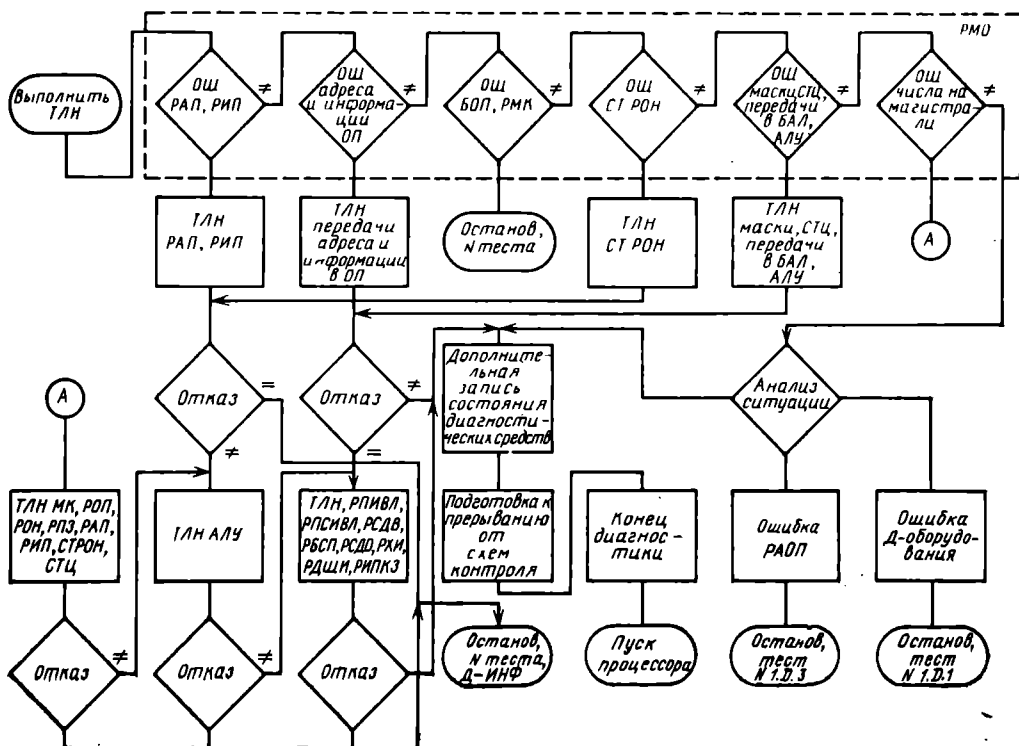


Рис. 126. Последовательность анализа РМО и выполнения ТЛН

работы процессора (см. табл. 35). В результате анализа выполняются ТЛН именно того блока, сигнал ошибки которого зафиксирован в данной группе разрядов. В случае ошибок БОП0—БОП3 и ошибок пересылки содержимого РМК0—РМК2 из БОП0—БОП2 тесты не выполняются, а происходит останов процессора, поскольку в данном случае ошибка оказывается локализованной с точностью до сменного блока.

Анализ разрядов РМО, фиксирующих ошибки РССП [40/63], производится в процедуре классификации ошибки при определении возможности повторения команды с начала. При наличии зафиксированных ошибок выполняются автономные ТЛН РССП.

Анализ разрядов РМО, фиксирующих ошибки ПКЗ, РИ ОП и РА ОП, также производится в процедуре классификации ошибки. При ошибках РИ ОП и ПКЗ (см. события 7 и 8, табл. 36) ТЛН не выполняются, а управление путем прерывания передается

программным средствам восстановления (гл. 14). При ошибках РА ОП ТЛН также не выполняются. Происходит останов ЦП, на ПУ индицируется тест № 1. D. 3, идентифицирующий ситуацию.

Если будет зафиксирована ошибка числа на магистрали, то заранее сказать, каким из регистров, связанным с магистралью по чтению, она была вызвана, нельзя. Поэтому сначала выполняются ТЛН той магистрали, на которой была зафиксирована ошибка, а потом последовательно — ТЛН всех блоков, имеющих чтение на эту магистраль.

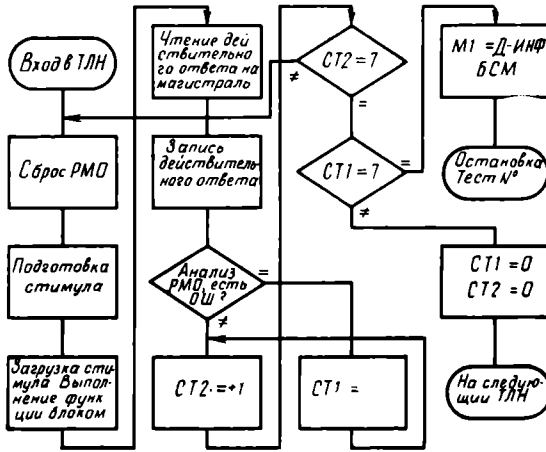


Рис. 127. Структура тестов локализации неисправности

Благодаря широкому распараллеливанию микроопераций процессора и особенности организации контроля АЛУ, в случае ошибки в последнем возможно ее распространение в другие регистры и узлы и фиксация их автономными схемами контроля. Если при анализе РМО окажется, что приоритет анализа ошибок этих узлов выше, чем приоритет анализа ошибок АЛУ и магистралей, то в первую очередь будут вы-

полняться ТЛН этих узлов. При этом отказ обнаружен не будет, поскольку ошибка была внесена извне. К таким узлам относятся РАП, РИП и СТ РОН, в которые информация может быть занесена после ее преобразования на АЛУ. Чтобы не получить ложную информацию о сбое (например, при ситуации, когда имеется отказ АЛУ, в РМО зафиксированы ошибки АЛУ и РИП, и выполнены ТЛН РИП, которыми отказ не обнаружен), после неуспешного выполнения ТЛН РАП, РИП и СТ РОН дополнительно тестируются АЛУ и его связи с магистралями.

Для поиска неисправности в функциональных узлах с помощью ТЛН используют автономные схемы контроля и схемы контроля числа на магистралях. Тестирование в этом случае заключается в том, что в проверяемый узел или блок подается определенный код, называемый стимулом, и с помощью схем контроля проверяется правильность функционирования этого блока. В соответствии с этим обобщенная схема построения ТЛН выглядит следующим образом (рис. 127).

Вначале происходит очистка РМО от сигналов ошибок, возникающих либо при нормальной работе ЦП, либо при выполнении предыдущих диагностических процедур (сброс РМО).

В общем случае для проверки информационных разрядов и цепей связи в каждый разряд проверяемого регистра необходимо последовательно записать значения «0» и «1». Для проверки цепей формирования контрольных разрядов их надо проверить с помощью комбинаций 00, 01, 10, 11. Этому условию удовлетворяет, в частности, такой набор стимулов: F1F1 F1F1, АААА АААА, 4444 4444, 1F1F 1F1F. Обычно стимул формируется на одной из магистралей путем считывания константы из БОПЗ. Затем он пересылается в проверяемый узел. В ряде случаев стимул формируется в ходе самого теста. Например, при проверке счетчиков формируется только начальный стимул 0000, а последующие стимулы 0001, 0010, 0011 и т. д. формируются с помощью АЛУ. Далее стимул поступает в проверяемый узел и этим узлом выполняется одна из его функций (запись в регистр, одна из функций АЛУ, один из сдвигов в БРСДВ и т. д.). После этого результат выполнения функции узлом (действительный ответ узла) считывается на магистраль. При этом ошибка в работе узла или его контрольного оборудования будет обнаружена схемами контроля числа на магистрали. Одновременно полученный результат (действительный ответ) записывается в один из исправных регистров, имеющих индикацию на ПУ. Это необходимо для того, чтобы в случае отказа можно было сравнить действительный ответ с эталоном. Описанная операция со стимулом повторяется семь раз, и если каждый раз при ее выполнении фиксируется ошибка, то событие классифицируется как отказ и производится останов по отказу. На ПУ индицируются информация, сформированная в ходе теста (состояние диагностических средств — Д-ИНФ), и номер самого теста, обнаружившего неисправность. Дальнейший поиск места отказа осуществляется с использованием этой информации и журнала неисправностей.

Если проверяемый узел имеет автономные схемы контроля, то ошибка функционирования будет ими обнаружена и необходимость в считывании результата на магистраль отпадает. Если этот узел имеет индикацию на ПУ, то действительный ответ сохраняется в самом узле (например, в случае РАП, РИП, СТ и др.).

В качестве иллюстрации рассмотрим упрощенный фрагмент микропрограммы одного из самых простых тестов — ТЛН РССП [32/63]. Этот регистр имеет автономные схемы контроля адресной части РССП [40/63]. Кроме того, РССП [32/63] имеет связи с МЗ по цепям считывания (п. 3.1). Таким образом, РССП [32/63] может проверяться двумя типами ТЛН: автономными и по ошибке числа на магистрали. Для обоих типов ТЛН используются общие ветви микропрограмм. Информация, записываемая в РССП [40/63], контролируется побайтно. Если в одной из восьмиразрядных групп РССП [40/63] схемами контроля будет обнаружена ошибка, то при анализе разрядов РМО, в которых фиксируются ошибки РССП [40/63] (см. табл. 35), произойдет переход на автономные ТЛН РССП. Анализ

разрядов РМО осуществляется в процедуре классификации ошибки.

В начале автономных ТЛН РССП (рис. 128) в единичное состояние устанавливается служебный триггер ТОРПЗ (триггер отказа РПЗ). Оно свидетельствует о том, что выполняются автономные ТЛН РССП.

Далее в РССП [40/63] записывается сброшенное значение магистрали МЗ (РССП [40/63] := := МЗ) для того, чтобы очистить регистр от некорректной информации. Затем в РАВД заносится константа 000.1000.00, которая, будучи дополнена справа еще двумя разрядами при считывании, содержащего РАВД, образует адрес возврата при безусловном переходе. В частности, по микроприказу ЧТ РАВД (00) два бита 00 дополняют эту константу, формируя адрес 000.1000.000 продолжения микропрограммы. По микрокоманде типа ДМК1 с адресом 0.8.0 на М1 из БОПЗ считывается константа (M1 := := F1F1 F1F1), которая является первым стимулом, проверяющим регистр. Чтобы сохра-

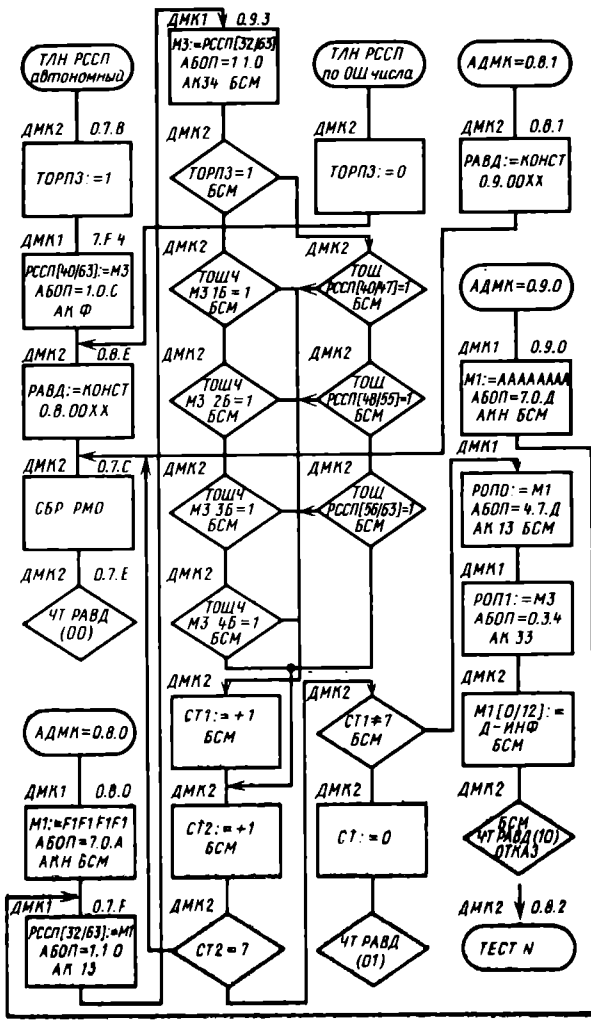


Рис. 128. Фрагмент ТЛН РССП [32/63]

нить состояние магистралей на следующий диагностический такт, в этой же ДМК1 имеется микроприказ блокировки сброса магистралей (БСМ). По следующей микрокоманде стимул с М1 записывается в РССП [32/63] (РССП [32/63] := М1), откуда действительный ответ из регистра считывается на МЗ (МЗ := РССП [32/63]) на случай выполнения ТЛН по ошибке числа. По результату анализа ТОРПЗ микропрограмма разветвляется. В случае использования автономных ТЛН далее выполняется анализ триггеров РМО, в которых могут быть зафиксированы сигналы ошибок РССП [40/63]: триггеры ТОШ РССП [40/47], ТОШ РССП [48/55] и ТОШ РССП [56/63]. Если хотя бы в одном из них за-

фиксирован сигнал ошибки, содержимое диагностического счетчика СТ1, подсчитывающего количество ошибок, обнаруженных при выполнении ТЛН, увеличивается на единицу. До начала выполнения ТЛН счетчики СТ1 и СТ2 сброшены. Счетчик СТ2 подсчитывает количество операций с одним и тем же стимулом. Операция со стимулом повторяется семь раз, после чего анализируется содержимое СТ1. Если СТ1=7, т. е. при каждой записи операнда F1 F1 F1 в РССП [40/63] возникает ошибка, событие классифицируется как отказ. Содержимое М1 (стимул, который является эталоном действительного ответа при исправном регистре) записывается в РОПО (РОПО := М1), содержимое М3 (действительный ответ регистра) записывается в РОП1. После этих действий на М1 считывается диагностическая информация, представляющая собой состояние диагностических средств (М1 [0/12] := Д-ИНФ), дается микроприказ ОТКАЗ и считывается содержимое РАВД, к которому подформировывается пара бит 10 [ЧТ РАВД (10)]. По микроприказу ОТКАЗ после выборки следующей микрокоманды происходит останов по отказу. На пульте управления загорается лампочка «ОТКАЗ». Следующая микрокоманда, адрес которой будет 0.8.2, в поле № ТЕСТА (разряды [21/31] см. рис. 123) содержит номер теста, обнаружившего отказ.

Таким образом, после останова ЦП и загорания лампочки «ОТКАЗ» оператор располагает следующей информацией на ПУ: диагностической информацией, которая после останова непосредственно индицируется на первой линейке индикации ПУ в разрядах [0/12]. В данном случае ею являются состояние ТОРПЗ, показывающего, какой исполнялся тест: автономный или по числу;

эталонный и действительный ответы. В случае ТЛН по числу эталон и действительный ответ хранятся в РОПО и РОП1 соответственно, которые индицируются на первой и второй линейках индикации ПУ (п. 3.6). В случае автономных ТЛН по числу эталон хранится в РОПО, а действительный ответ в РССП [40/63];

номер теста, обнаружившего неисправность. Он индицируется на четвертой линейке ПУ при индикации РМК6;

группа разрядов, в которой зафиксированы сигналы ошибки. Они определяются при индикации РМО.

Пользуясь индикацией РМО и сравнивая в соответствующих группах разрядов эталон и действительный ответ, можно обнаружить разряд, в котором действительный ответ не соответствует эталону. В дальнейшем, используя полученную информацию и пользуясь журналом (словарем) неисправностей, входящим в комплект документации на машину, локализируют место неисправности с точностью до ТЭЗа.

Если СТ1 \neq 7, то сбрасываются счетчики СТ1 и СТ2 (СТ := 0) и считыванием содержимого РАВД [ЧТ РАВД (01)] формируется

адрес возврата 0.8.1 По адресу ДМК 0.8.1 производится занесение в РАВД новой константы и микропрограмма повторяется. Только теперь уже при ЧТ РАВД (00) будет сформирован адрес 0.9.0. Далее формируется новый стимул, с которым и выполняется операция.

10.6. КОМАНДА ДИАГНОСТИКА

Эта команда с кодом операции 83 не предназначена для использования в основных программах и выполняется только в состоянии СУПЕРВИЗОР. Мнемонического обозначения она не имеет, формат SI (рис. 129).

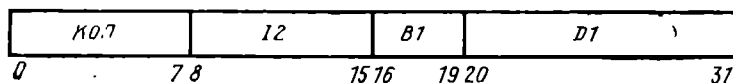


Рис. 129. Команда ДИАГНОСТИКА

Назначение полей команды следующее: старший, т. е. восьмой разряд поля I2, определяет, куда должно быть передано управление выполнением данной команды — процессору или ОК. Если

Т а б л и ц а 37

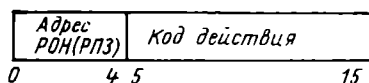
Действия по команде ДИАГНОСТИКА

| Действие | Дополнительная информация | Код диагностического действия | | | |
|---|---------------------------|-------------------------------|------|-------|--|
| | Разряды УСКД | | | | |
| | 0/4 | 5/7 | 8/11 | 12/15 | |
| 1. Установка режима «Контрольный код» | X X X X X | 101 | 0000 | 1010 | |
| 2. Сброс режима «Контрольный код» | X X X X X | 101 | 0000 | 1011 | |
| 3. Установка режима «Блокировка РМО» | X X X X X | 101 | 0000 | 1100 | |
| 4. Сброс режима «Блокировка РМО» | X X X X X | 101 | 0000 | 1101 | |
| 5. Запись информации с неправильной четностью в ОП по адресу 00 01 B8 | X X X X X | 101 | 0001 | 0111 | |
| 6. Запись информации с неправильной четностью в РОН по адресу, определяемому УСКД [0/3] | Адрес РОН X | 101 | 0001 | 1000 | |
| 7. Запись информации с неправильной четностью в РПЗ по адресу, определяемому УСКД [1/3] | X Адрес РПЗ X | 101 | 0001 | 1010 | |
| 8. Запись информации с неправильной четностью в РССП [0/31] | X X X X X | 101 | 0001 | 1001 | |
| 9. Имитация ошибок РМО | X X X X X | 101 | 0001 | 1100 | |
| 10. Запись состояния ЦП | X X X X X | 000 | 0000 | 0101 | |

этот бит равен единице, то управление передается процессору. Остальные биты поля игнорируются. Поля *B1* и *D1* определяют в ОП адрес управляющего слова команды ДИАГНОСТИКА (УСКД). Управляющее слово (рис. 130) располагается в ОП, начиная с границы слова или полуслова.

Разряды [0/4] используются не всегда. В этих случаях они могут находиться в произвольном состоянии.

Рис. 130. Управляющее слово команды ДИАГНОСТИКА



Разряды [5/15] определяют адрес начальной микрокоманды диагностической микропрограммы, выполняющей действия (табл. 37), предписанные командой ДИАГНОСТИКА.

С помощью команды ДИАГНОСТИКА можно также выполнить любую процедуру или отдельную микропрограмму диагностики. Для этого необходимо в УСКД в поле «код диагностического действия» задать адрес начальной микрокоманды этой микропрограммы.

Команда ДИАГНОСТИКА используется программными средствами восстановления.

ГЛАВА 11

ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА

11.1. ОБЩИЕ СВЕДЕНИЯ

К операциям ввода-вывода относятся передачи информации между оперативной памятью и внешними устройствами (ВУ). Те функции управления операциями ввода-вывода, которые являются общими и не зависят от конкретного типа ВУ, выполняются общими для них всех устройствами — каналами ввода-вывода. Функции управления, специфичные для конкретных ВУ, выполняются специализированными устройствами управления (УВУ). Последние связаны с центральным процессором через каналы ввода-вывода.

Система ввода-вывода предусматривает стандартизацию процесса передачи данных как при управлении, так и при подключении внешних устройств. Это является одним из принципиальных решений, положенных в основу построения ЕС ЭВМ. Рассмотрим кратко основные положения, понятия и термины. Более полные сведения об организации ввода-вывода можно найти в работах [14, 17].

К внешним устройствам относятся внешняя память, устройства ввода-вывода и средства связи между системами обработки данных. Например, внешние запоминающие устройства на магнитных дисках и лентах, устройства для перфорации и чтения с перфолент и перфокарт, печатающие устройства, устройства обработки информации, передаваемой по линиям связи и т. п. Большинство ВУ работает с внешними физическими носителями информации. Устройства, не работающие непосредственно с физическими носителями информации, состоят лишь из электронного оборудования.

Логические возможности каждого ВУ реализованы в его устройстве управления. Это позволяет независимо от характеристик ВУ стандартным образом организовывать управление ими со стороны канала. Каждое конкретное устройство управления расшифровывает команды, поступающие из канала, интерпретирует их для данного типа ВУ и формирует последовательность сигналов, необходимых для выполнения операции. Оно также обеспечивает передачу в каналы информации о состоянии ВУ. Устройство управления может быть выполнено отдельно или физически и логически составлять единое целое с ВУ. Одно устройство управления может обслуживать одно или несколько ВУ.

Канал ввода-вывода под управлением ЦП организует и осуществляет обмен информацией между ВУ и оперативной памятью. При этом в общем случае канал выполняет следующие функции:

- получает от ЦП команды ввода-вывода;

- производит логическое подключение ВУ, заданного в команде ввода-вывода;

- выбирает из оперативной памяти управляющую информацию, относящуюся к каналу, проверяет на отсутствие аппаратных и программных ошибок и расшифровывает ее;

- выбирает из оперативной памяти данные 32-разрядными словами, обеспечивает их промежуточное хранение и побайтную выдачу в ВУ;

- принимает из ВУ байты данных, komponует их в 32-разрядные слова и производит запись в ОП целыми словами;

- подсчитывает число передаваемых байтов и контролирует их по четности;

- получает от ВУ информацию об их состоянии;

- сохраняет информацию о состоянии канала;

- упорядочивает запросы на прерывание от ВУ;

- направляет запросы на прерывание в ЦП;

- пересылает, по разрешению ЦП, информацию о состоянии канала в оперативную память.

Для реализации этих функций каналы обладают определенными логическими возможностями и имеют средства для буферного запоминания передаваемых данных. Это дает возможность осуществлять обработку данных центральным процессором па-

раллельно с выполнением операций ввода-вывода. Степень параллелизма зависит от конкретной реализации оборудования канала.

Различают два режима передачи данных между оперативной памятью и внешним устройством: монополюный и мультиплексуный.

В монополюном режиме ВУ остается в логической связи с каналом, и ему принадлежат все средства управления, имеющиеся в канале, до тех пор, пока не закончится передача всей группы данных. Группа может состоять из нескольких байтов, целого блока данных или их последовательности с соответствующей управляющей информацией.

В мультиплексуном режиме канал может обслуживать несколько одновременно выполняемых операций ввода-вывода. В этом режиме каждое ВУ занимает средства канала на определенные интервалы времени, в течение каждого из которых передается только сегмент информации. Сегмент может представлять собой один или группу байтов, а также управляющую информацию для ВУ.

Группы данных в мультиплексуном и монополюном режимах имеют различную длину. Считается, что канал работает в монополюном режиме, если для передачи группы данных ВУ монополизует средства канала на время, превышающее 32 мкс.

Средства канала, необходимые для выполнения отдельной операции ввода-вывода, называются подканалом. Подканал представляет собой часть памяти канала, используемую для хранения адресов, содержимого счетчика передаваемых байтов, управляющей информации и информации о состоянии устройств ввода-вывода. Возможность работы канала в том или ином режиме определяется количеством подканалов.

Селекторный канал (СК) имеет только один подканал, работает только в монополюном режиме и не может обслуживать несколько ВУ одновременно. Он обеспечивает обмен информацией с ВУ, работающим с относительно высокой скоростью передачи данных, например, с накопителем на магнитных дисках, лентах и т. п.

Мультиплексуный канал (МК) имеет несколько подканалов и может работать как в мультиплексуном, так и в монополюном режимах. В мультиплексуном режиме МК способен одновременно обеспечить выполнение по одной операции ввода-вывода в каждом подканале при условии, что общая нагрузка на канал не превышает его пропускной способности. В этом режиме канал обменивается информацией с ВУ, работающими с относительно малой скоростью передачи данных (с устройствами ввода-вывода на перфокартах, перфолентах, печатающими устройствами и т. п.).

В монополюном режиме подканал монополизует все средства канала и проявляет себя по отношению к программе как отдельный селекторный канал до тех пор, пока не завершится интервал времени, в течение которого осуществляется работа в этом режиме.

Наличие в мультиплексном канале монопольного режима позволяет подсоединить быстродействующие ВУ к мультиплексному каналу.

11.2. ИНТЕРФЕЙС ВВОДА-ВЫВОДА

Все связи между устройствами управления ВУ и каналом осуществляются через стандартное сопряжение, получившее название интерфейс ввода-вывода (ИВВ), представляющий собой унифицированную систему связей и сигналов между каналами ввода-вывода и УВУ. Все устройства, подключаемые к каналу посредством интерфейса ввода-вывода, обычно называются абонентами.

Интерфейс ввода-вывода (табл. 38) обеспечивает единые правила обмена информацией, формат информации (байт), выработку и передачу последовательностей управляющих сигналов, унифицированные средства подключения и управления различными ВУ. Под информацией понимаются данные, подлежащие передаче команды, адреса, байты состояния ВУ.

Для выборки ВУ канал сначала выдает в шины интерфейса его адрес, а затем сигнал выборки. На это выбранное устройство отвечает выдачей своего адреса, совпавшего с принятым от канала, и тем самым определяет себя. Если УВУ инициирует последовательность сигналов, то оно должно определять себя путем передачи своего адреса перед тем, как указать цепь последовательности сигналов.

Интерфейс обеспечивает также взаимную блокировку и выработку последовательностей сигналов выполнения операций ввода-вывода и последовательностей сигналов управления. Операции ввода-вывода включают в себя одно или несколько действий: начальную выборку ВУ; выборку занятого ВУ; выборку, вводимую ВУ; передачу данных, окончание операции.

Для управления операцией ввода-вывода, передачи специальных указаний ВУ и приведения системы ввода-вывода в исходное состояние используются специальные последовательности сигналов управления:

ПРОДОЛЖИТЬ;
ОСТАНОВ;
ЗАПОМНИТЬ СОСТОЯНИЕ;
БЛОКИРОВКА ДАННЫХ;
БЛОКИРОВКА СОСТОЯНИЯ;
ОПРЕДЕЛЕНИЕ ЦЕПОЧКИ КОМАНД;
ОТКЛЮЧЕНИЕ ОТ ИНТЕРФЕЙСА;
СЕЛЕКТИВНЫЙ СБРОС;
СБРОС СИСТЕМЫ.

Более подробное описание интерфейса ввода-вывода дано в [17, 23].

Линии интерфейса

| Название линии | Условное обозначение | Назначение |
|------------------------|----------------------|---|
| Шина канала контроль | ШИН-КК | Информационные шины канала. Используются для передачи информации (данных, адресов ВУ, команд) от канала к абоненту |
| » » 0 | ШИН-К0 | |
| » » 1 | ШИН-К1 | |
| » » 2 | ШИН-К2 | |
| » » 3 | ШИН-К3 | |
| » » 4 | ШИН-К4 | |
| » » 5 | ШИН-К5 | |
| » » 6 | ШИН-К6 | |
| » » 7 | ШИН-К7 | Информационные шины абонента. Используются для передачи информации (данных, адресов ВУ, информации о состоянии абонента) от абонента к каналу |
| Шина абонента контроль | ШИН-АК | |
| » » 0 | ШИН-А0 | |
| » » 1 | ШИН-А1 | |
| » » 2 | ШИН-А2 | |
| » » 3 | ШИН-А3 | |
| » » 4 | ШИН-А4 | |
| » » 5 | ШИН-А5 | |
| » » 6 | ШИН-А6 | Линии идентификации Используются для определения вида информации, находящейся на информационных шинах канала и абонента, а также для взаимной блокировки сигналов Линии управления Используются для управления выборкой абонента, для сканирования и управления подключением абонента к каналу |
| » » 7 | ШИН-А7 | |
| Адрес от канала | АДР-К | |
| » » абонента | АДР-А | |
| Управление канала | УПР-К | |
| » абонента | УПР-А | |
| Информация канала | ИНФ-К | |
| » абонента | ИНФ-А | |
| Работа канала | РАБ-К | Линии специального управления. Используются для управления службой времени и сменной состоянием абонента |
| » абонента | РАБ-А | |
| Разрешение выборки | РВБ-К | |
| Выборка | ВБР-К | |
| Обратная выборка | ВБР-А | |
| Блокировка | БЛК-К | |
| Требование абонента | ТРБ-А | |
| Измерение от канала | ИЗМ-К | |
| » » абонента | ИЗМ-А | |
| Смена состояния | СМС-К | |

11.3. КОМАНДЫ ВВОДА-ВЫВОДА

В системе команд семейства «Ряд 1» предусмотрены четыре команды для управления вводом-выводом (прил. 4), выполняемые центральным процессором. Эти команды используются для начала и прекращения операции ввода-вывода, а также для проверки состояния канала и ВУ.

По команде НАЧАТЬ ВВОД-ВЫВОД SIO (9С) начинается любая операция ввода-вывода в ВУ и в подканале, адрес которого определяется как сумма содержимого регистра, указанного полем *V1* и содержимым поля *D1* командного слова. Команда устанавливает признак результата, используемый для определе-

ния состояния адресуемого ВУ и канала, а также для подтверждения принятия команды.

По команде ПРОВЕРИТЬ ВВОД-ВЫВОД ТЮ (9D) проверяется состояние системы ввода-вывода и устанавливается соответствующий признак результата.

По команде ОСТАНОВИТЬ ВВОД-ВЫВОД НЮ (9E) прекращается выполнение операции ввода-вывода в канале и в ВУ и в зависимости от условий прекращения устанавливается соответствующий признак результата.

По команде проверить канал ТСН (9F) проверяется адресуемый канал и в зависимости от его состояния устанавливается соответствующий признак результата. Эта команда не влияет на состояние канала.

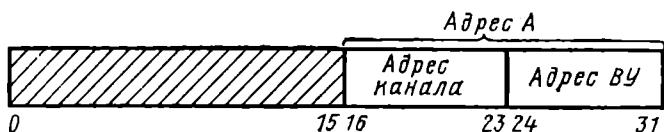


Рис. 131. Адрес ввода-вывода

Все эти команды имеют формат S1 (прил. 5). Разряды [8/15] команды не используются. Содержимое поля *В1* определяет общий регистр. Сумма, полученная сложением содержимого этого регистра с содержимым поля *Д1* команды, определяет адрес *А* ввода-вывода (рис. 131). Разряды *А* [0/15] этой суммы не используются, а в разрядах *А* [16/31] указывается адрес канала и адрес устройства в этом канале.

В модели ЕС-1033 код в разрядах *А* [22/23] определяет номер канала:

- 00 — мультиплексный канал;
- 01 — селекторный канал 1;
- 10 — селекторный канал 2;
- 11 — селекторный канал 3.

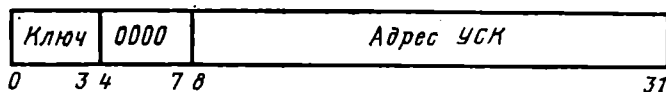
В селекторном канале код в разрядах *А* [24/27] определяет адрес ВВУ, а код в разрядах *А* [28/31] — адрес ВУ. Эти коды могут принимать значения от 0000 до 1111.

В мультиплексном канале значение разряда *А* [24] определяет тип подканала: не разделенный ($A[24] = 0$) или разделенный ($A[24] = 1$). При $A[24] = 0$ код в разрядах *А* [25/31] определяет 128 неразделенных подканалов и соответствующих им ВУ. При $A[24] = 1$ код в разрядах *А* [25/27] определяет 8 разделенных подканалов, к каждому из которых может быть подключено до 16 ВУ с адресами, определяемыми кодом в разрядах *А* [28/31]. Таким образом, теоретически неразделенные подканалы позволяют подключать ВУ с адресами от 0000 0000 до 0111 1111, и адреса ВУ в этом случае совпадают с адресами подканалов. Каждый разделенный подканал с адресом 000—111 позволяет подключить до 16 ВУ с адресами 0000—1111, каждое из

которых имеет одно общее устройство управления. В этом случае адрес УВУ совпадает с адресом подканала.

С точки зрения физической реализации из 128 неразделенных подканалов первые восемь могут быть использованы для подключения ВУ, разделяющих подканал. Следовательно, в этом случае в системе должны отсутствовать ВУ с адресами 0000 0000—0000 0111.

Рис. 132. Адресное слово канала

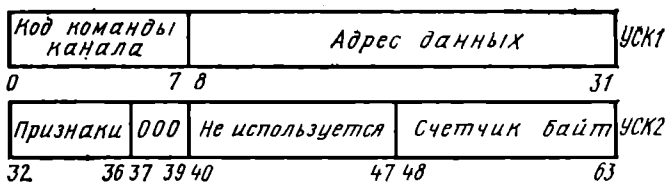


Все команды ввода-вывода являются привилегированными и выполняются только тогда, когда процессор находится в состоянии СУПЕРВИЗОР.

Программа ЦП начинает операции ввода-вывода с команды НАЧАТЬ ВВОД-ВЫВОД, в процессе выполнения которой канал выбирает из ячейки ОП с фиксированным адресом 48₁₆ (прил. 10) адресное слово канала АСК (рис. 132).

Разряды АСК [0/3] содержат ключ защиты памяти, который будет использоваться в операции ввода-вывода. Разряды АСК [4/7] должны содержать нули. АСК [8/31] указывает в ОП

Рис. 133. Управляющее слово канала



адрес управляющего слова канала УСК (рис. 133). Оно представляет собой команду канала, выполняемую каналом и ВУ, которые указаны адресом ввода-вывода. Для выполнения операций ввода-вывода может быть задана последовательность УСК, которые могут располагаться в соседних или несмежных ячейках ОП. Последовательность выполняемых УСК называется программой канала. УСК имеет формат двойного слова. Распределение полей УСК приведено в табл. 39.

Если УСК [32] = 1, то это указывает на цепочку данных. Канал после завершения передачи данных по текущему УСК выбирает из следующей ячейки ОП очередное управляющее слово и продолжает операцию, заданную предыдущим словом, но с использованием области ОП, заданной новым УСК. Если УСК [32] = 0, то текущее УСК является последним в операции ввода-вывода. В случае если УСК [33] = 1, то имеет место цепочка команд. Это означает, что после реализации данного УСК происходит переход к новому управляющему слову, содержащему

Распределение полей УСК

| Разряды | Поле | Функция |
|-------------------------|---|--|
| 0/7 8/31 | Код команды канала Адрес данных | Определяет выполняемую операцию. Определяет начало или конец области данных в ОП |
| 32/36 32 | Поле признаков Признак цепочки данных (ЦД) | Указывает на использование адресной части следующего УСК |
| 33 | Признак цепочки команд (ЦК) | Указывает на использование кода операции и адреса данных следующего УСК |
| 34 | Признак блокировки неправильной длины (БНД) | Запрещает формирование признака неправильной длины |
| 35 | Признак блокировки записи в ОП | Блокирует передачу информации в ОП |
| 36 | Признак программно управляемого прерывания | Вынуждает канал генерировать запрос на прерывание, когда он считывает УСК, содержащее этот признак |
| 37/39 40/47 48/63 | 000 Счетчик байт | — Не используются Определяет число байтов, которое должно быть передано по этому УСК |

новый код операции. Цепочка команд отсутствует, если $УСК [33] = 0$ или $УСК [32] = 1$, т. е. при наличии признака цепочки данных цепочка команд блокируется.

В канале формируется сигнал неправильной длины при несоответствии количества байтов, фактически передаваемых между ОП и ВУ, количеству байтов, заданных выполняемым УСК. При $УСК [34] = 1$ происходит блокировка этого сигнала.

Если $УСК [35] = 1$, то запись данных в ОП не производится, но продвижение счетчика происходит.

Значение $УСК [36] = 1$ вызывает программно-управляемое прерывание.

Младшие разряды кода команды канала определяют тип операции ввода-вывода, а старшие — модификацию основной операции, определяемую на уровне УВУ и ВУ и зависящую от специфики их построения и работы. Всего имеется шесть команд канала.

Основные команды ввода-вывода и соответствующие им коды приведены в табл. 40, в которой буква М обозначает разряд модификатора, а Х — состояние разряда безразлично.

Канал различает только четыре операции: ввод (команды чтения и уточнения состояния), вывод (команды записи и управления), ввод в обратном направлении (команда чтения в обратном направлении) и переход в канале.

Команды канала

| Код команды | Команда |
|--|---|
| ММММ ММ01 ММММ ММ10 ММММ ММ11 ММММ 0100 ММММ 1100 ХХХХ 000 | ЗАПИСАТЬ СЧИТАТЬ УПРАВЛЕНИЕ УТОЧНИТЬ СОСТОЯНИЕ СЧИТАТЬ В ОБРАТНОМ НАПРАВЛЕНИИ ПЕРЕХОД В КАНАЛЕ |

11.4. СОСТОЯНИЕ СИСТЕМЫ ВВОДА-ВЫВОДА

Состояние системы ввода-вывода определяется совокупностью состояний трех ее компонентов — канала, подканала и ВУ. Каждая из этих компонент может иметь одно из четырех состояний, влияющих на выполнение команды ввода-вывода, инициированной центральным процессором. Эти состояния отражены в табл. 41.

Т а б л и ц а 41

Состояние системы ввода-вывода

| Состояние | Компонент | Описание |
|-------------------------|---------------------------|--|
| «Доступен» (Д) | Канал, подканал или ВУ | Другие состояния отсутствуют |
| «Хранит прерывание» (П) | Канал | Имеется запрос на немедленное прерывание от канала |
| | Подканал | Имеется информация для прерывания в подканале |
| | ВУ | Устройство хранит информацию для прерывания |
| «Работает» (Р) | Канал | Канал работает в монопольном режиме |
| | Подканал | Подканал выполняет какую-либо операцию |
| | ВУ | ВУ выполняет операцию ввода-вывода |
| «Неработоспособен» (Н) | Канал, подканал ВУ | Неработоспособен |

Анализ состояния системы ввода-вывода позволяет установить одно из четырех возможных значений признака результата (0, 1, 2, 3) в РССП [34/35]. Установка происходит после завершения команды ввода-вывода в канале. Условия, которые принимаются во внимание при установке признака результата, приведены в табл. 42.

Каждая из позиций в графе «Условное обозначение» указывает состояние компонент системы: канала, подканала и ВУ, звездочкой отмечены случаи, когда ССК (рис. 134) или его часть запо-

Установка признака результата в операциях ввода-вывода

| Состояние системы | Условное обозначение | Признак результата | | | |
|--|----------------------|--------------------|-----|----|-----|
| | | SIO | TIO | НЮ | TCH |
| Доступен | ДДД | 0,1* | 0 | 1* | 0 |
| Хранится прерывание в ВУ | ДДП | 1* | 1* | 1* | 0 |
| ВУ работает | ДДР | 1* | 1* | 1* | 0 |
| ВУ неработоспособно | ДДН | 3 | 3 | 3 | 0 |
| Подканал хранит информацию для прерывания: | ДПХ | | | | |
| для адресуемого ВУ | | 2 | 1* | 0 | 0 |
| » другого ВУ | | 2 | 2 | 0 | 0 |
| Подканал работает: | ДРХ | | | | |
| с адресуемым ВУ | | 2 | 2 | 1* | 0 |
| с другим ВУ | | 2 | 2 | 1 | 0 |
| Подканал неработоспособен | ДНХ | 3 | 3 | 3 | 0 |
| Подканал хранит прерывание | ПХХ | 2 | — | 2 | 1 |
| Канал работает: | РХХ | | | | |
| с адресуемым ВУ | | 2 | 2 | 2 | 2 |
| с другим ВУ | | 2 | 2 | 2 | 2 |
| Канал неработоспособен | НХХ | 3 | 3 | 3 | 3 |

минается в ячейке ОП по адресу 40_{16} (прил. 10) во время выполнения команды.

Индекс X рядом с обозначениями Д, П, Р и Н указывает на то, что состояние соответствующих компонентов несущественно. Например, состояние ДПХ означает любое из состояний ДПД, ДПП, ДПР или ДПН; состояние ПХХ означает 16 состояний, причем некоторые из них никогда не возникают.

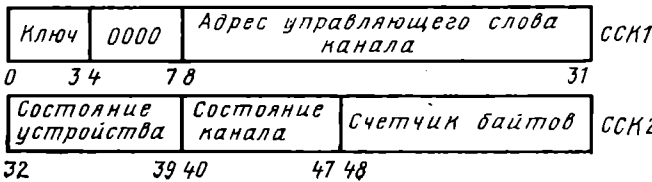


Рис. 134. Слово состояния канала

Для выполнения команд SIO, TIO, НЮ состояние канала, продолжающего хранить прерывание, является таким же, как и канала в состоянии «Доступен», и установка признака результата зависит от состояния подканала и устройства. Признаки результата для состояния ПХХ такие же, как и для состояния РХХ, где X означает состояние подканала и ВУ.

Программа определяет состояние ВУ или условия, при которых была прекращена операция ввода-вывода, по информации, содержащейся в слове состояния канала. Формирование нового ССК или отдельных его частей производится при обработке пре-

Распределение полей ССК

| Номера разрядов | Поле | Содержимое |
|-----------------|-------------------|---|
| 0/3 | Ключ защиты | Ключ защиты (из АСК), используемый подканалом Адрес последнего используемого УСК, продвинутый на +8 Характеристика состояния канала и ВУ. Значения разрядов [32/39] принимаются по интерфейсу (табл. 44), значения разрядов [40/47] формируются каналом (табл. 45) Остаточный счетчик последнего использованного УСК |
| 8/31 | Адрес УСК | |
| 32/47 | Разряды состояния | |
| 48/63 | Счетчик | |

Таблица 44

Байт состояния ВУ (U)

| Разряд ССК | Идентификатор | Признак | Условия установки |
|------------|---------------|--------------------|---|
| 32 | U0 | ВНИМАНИЕ | Специальной клавишей на пульте управления УВУ Устанавливается совместно с U3 или U5 для модификации основного значения состояния УВУ закончило операцию ввода-вывода ВУ занято выполнением предыдущей операции ввода-вывода. Если одновременно установлены U1 и U3, занято УВУ Подканал закончил свою часть операции передачи данных, ВУ к этому моменту может и не закончить операцию ввода-вывода ВУ закончило операцию ввода-вывода и готово для выполнения других операций. ВУ перешло из состояния «Не готово» в состояние «Готово» В ВУ имеется один или несколько разрядов уточненного байта состояния. Программа должна его проверить для того, чтобы определить, что произошло в ВУ В ВУ произошел особый случай, например, конец страницы на печатающем устройстве, пустой карман в устройстве ввода-вывода карт |
| 33 | U1 | МОДИФИКАТОР | |
| 34 | U2 | УВУ КОНЧИЛО | |
| 35 | U3 | ЗАНЯТО | |
| 36 | U4 | КАНАЛ КОНЧИЛ | |
| 37 | U5 | УСТРОЙСТВО КОНЧИЛО | |
| 38 | U6 | СБОЙ В УСТРОЙСТВЕ | |
| 39 | U7 | ОСОБЫЙ СЛУЧАЙ | |

Байт состояния канала (S)

| Разряд ССК | Идентификатор | Ситуация | Условия установки |
|------------|---------------|-----------------------------------|---|
| 40 | S0 | Программно-управляемое прерывание | В УСК[36] установлен признак прерывания |
| 41 | S1 | Неправильная длина | Поле счетчика УСК не равно длине блока информации на внешнем носителе и в УСК не установлен признак блокировки неправильной длины |
| 42 | S2 | Ошибка программы | Неверно АСК или УСК, указан несуществующий подканал и т. д. |
| 43 | S3 | Ошибка защиты | Ключ защиты, указанный в АСК, не сравнился с ключом основной памяти |
| 44 | S4 | Ошибка данных канала | В канале обнаружена ошибка в четности байта данных. Канал исправляет контрольный разряд при записи информации в основную память |
| 45 | S5 | Ошибка в управлении канала | В канале обнаружен неверный сигнал управления. Подканал или ВУ неисправны |
| 46 | S6 | Ошибка в управлении интерфейса | В ВУ или УВУ, подключенном к подканалу, произошла неисправность |
| 47 | S7 | Ошибка в цепочке данных | При операциях ввода в канале происходит переполнение во время работы с цепочкой данных, когда скорость передачи данных слишком высока |

рывания от ввода-вывода, а также во время выполнения команд SIO, TIO и NIO. Слово состояния канала заносится в ячейку 40₁₆ оперативной памяти.

Если ССК запоминается в результате выполнения команды ввода-вывода, то адрес ВУ, для которого ССК запомнено, соответствует указанному в команде. Если запоминание ССК является результатом прерывания от ввода-вывода, то адрес канала и ВУ определяется содержимым поля прерывания старого ССП. Распределение полей ССК показано в табл. 43—45.

11.5. ВЫПОЛНЕНИЕ ОПЕРАЦИЙ ВВОДА-ВЫВОДА. ПРЕРЫВАНИЯ

В организации выполнения операций ввода-вывода предусмотрено использование программы канала, представляющей собой совокупность управляющих слов канала. Адресное слово, относящееся к этой совокупности, находится в фиксированной ячейке 48₁₆ ОП.

Выполнение операции ввода-вывода предусматривает три стадии: начальную выборку, передачу данных и окончание.

Первая стадия, как уже указывалось, включает в себя выполнение команды НАЧАТЬ ВВОД-ВЫВОД. В процессе выполнения этой команды обеспечиваются проверка АСК и УСК на отсутствие программных ошибок, а также аппаратный контроль выполнения команды в канале. При этом обеспечивается выборка ВУ в последовательности, определяемой требованиями стандартного интерфейса. Происходит запуск ВУ и анализ его состояния, а также устанавливается признак результата в ССП, чем завершается выполнение команды НАЧАТЬ ВВОД-ВЫВОД в центральном процессоре.

Вторая стадия операции — передача данных — начинается, если она предусмотрена кодом команды канала и если канал не обнаружит программных и аппаратных ошибок, а ВУ в ходе выполнения первой стадии выдает нулевой байт состояния.

Окончание операции в подканале, как правило, характеризуется двумя признаками в байте состояния: КАНАЛ КОНЧИЛ и ВУ КОНЧИЛО. Признак КАНАЛ КОНЧИЛ показывает, что передача данных завершена с точки зрения использования средств канала. Признак ВУ КОНЧИЛО указывает на завершение операции внешним устройством, он может быть сформирован одновременно с сигналом КАНАЛ КОНЧИЛ или позже него. В операциях, связанных с использованием общего для нескольких ВУ устройства управления, после признака КАНАЛ КОНЧИЛ формируется признак УВУ КОНЧИЛО. Он отражает готовность УВУ к выполнению следующей операции.

Если в текущем УСК задан признак ЦК, то формирование признака ВУ КОНЧИЛО приводит каждый раз к организации прохождения такой же последовательности сигналов через интерфейс, как и при выполнении первой команды канала, заданной командой НАЧАТЬ ВВОД-ВЫВОД. Говорят, что в этом случае имеет место перевыборка ВУ по цепочке команд. Канал выбирает новое УСК и начинает выполнение новой команды канала с тем же ВУ. При этом сигналы, характеризующие окончание отдельной операции, программе не сообщаются до тех пор, пока текущее УСК не будет последним в цепочке.

Окончание операции в начальной стадии вызывается либо ошибками в программе, либо отказом ВУ принять команду канала. В слове состояния программы устанавливается соответствующее значение признака результата, а условия, которые помешали начать операцию, более подробно отражаются той частью ССК, которая заносится в память во время выполнения команды НАЧАТЬ ВВОД-ВЫВОД. Команда канала считается непринятой, если канал обнаружил необычную ситуацию в начальной стадии или получил от ВУ сигнал о такой ситуации без сигнала

КАНАЛ КОНЧИЛ. Если необычная ситуация вызвала отказ принять команду канала, которая не является первой в цепочке команд, то это приводит к выработке сигнала прерывания.

Существуют операции (например, некоторые операции управления), когда сразу после получения кода команды ВУ может выдать сигнал **КАНАЛ КОНЧИЛ**. Такая операция ввода-вывода, при которой сигнал **КАНАЛ КОНЧИЛ** выдается во время начальной последовательности, называется «операция с немедленным ответом». Для операций, при которых не происходит передачи данных, момент выдачи сигнала **КАНАЛ КОНЧИЛ** обычно определяется ВУ.

Прекращение операции ввода-вывода в стадии передачи данных может быть вызвано как каналом, так и ВУ. Канал посылает в ВУ указание прекратить передачу данных, если количество байтов, указанное в УСК, будет исчерпано, либо будут обнаружены ошибки в программе и нарушение защиты, либо возникнет ошибка в цепочке (табл. 45).

Значение счетчика байтов в последнем УСК, равное нулю, указывает на то, что канал передал всю информацию. Условия, возникающие из-за ошибок, вызывают преждевременное прекращение передачи данных. В любом случае прекращение передачи данных осуществляется в ответ на запрос на обслуживание от ВУ.

Каждая команда канала, за исключением команды перехода, вызывает соответствующую операцию ввода-вывода. Термин «операция ввода-вывода» относится к действиям, которые вызываются командой канала в ВУ и соответствующем подканале. Подканал участвует в выполнении операции с момента поступления команды до получения сигнала **КАНАЛ КОНЧИЛ** или, в случае цепочки команд, до получения сигнала **ВУ КОНЧИЛО** (операция в ВУ длится до появления сигнала **ВУ КОНЧИЛО**). Таким образом, процесс выполнения операции ввода-вывода и ее окончание определяются типом команды канала (наличием или отсутствием передачи данных), признаком ЦК, типом самого канала (мультиплексный или селекторный) и режимом его работы (мультиплексный или монопольный).

Операция ввода-вывода может быть завершена, кроме того, путем выполнения команды **ОСТАНОВИТЬ ВВОД-ВЫВОД**, а также в результате сбоя в работе аппаратуры канала или сброса системы ввода-вывода в исходное состояние.

Следует заметить, что ошибка в данных (S4) не вызывает прекращения их передачи. Эта ошибка становится доступна программе только в слове состояния канала при обработке прерывания после завершения операции ввода-вывода.

Прерывания от ввода-вывода и вызванная ими реакция дают возможность операционной системе определить результат выполнения операции ввода-вывода по содержимому ССК, записанному

в ОП по адресу 40₁₆. Прерывания могут быть вызваны либо программой, либо обстоятельствами, возникшими в ВУ или канале.

Если прерывания от канала не замаскированы (прил. 7) и запрос на прерывание принят центральным процессором, условие прерывания сбрасывается. Условия прерывания в канале могут быть сброшены также в процессе выполнения команды ПРОВЕРИТЬ ВВОД-ВЫВОД, а условия прерывания, возникшие в ВУ, — в процессе выполнения команды НАЧАТЬ ВВОД-ВЫВОД.

Внешнее устройство выдает в канал запрос на прерывание всякий раз, когда в нем формируется хотя бы один из признаков:

КАНАЛ КОНЧИЛ,
ВУ КОНЧИЛО,
УВУ КОНЧИЛО,
ВНИМАНИЕ,
ОШИБКА В УСТРОЙСТВЕ,
ОСОБЫЙ СЛУЧАЙ В УСТРОЙСТВЕ.

Запросы на прерывание в канале вырабатываются, если канал обнаружил ошибки в программе (S2) и защите (S3), в управлении канала (S5) и интерфейса (S6), а также в тех случаях, когда в УСК задан признак программно управляемого прерывания или в селекторном канале выполняется команда ПРОВЕРИТЬ ВВОД-ВЫВОД.

Все запросы на прерывание асинхронны по отношению к действиям, выполняемым ЦП. Но поскольку одновременно могут существовать сигналы прерывания, связанные с несколькими ВУ, подсоединенными к различным каналам, то возникает необходимость упорядочивания этих запросов. Упорядочивание происходит на двух уровнях: на первом устанавливается приоритет для сигналов прерывания, связанных с ВУ, подключенными к одному каналу, на втором — запросы от различных каналов.

Приоритеты запросов на прерывание от ВУ в каждом канале зависят от типа прерывания и от последовательности подключения устройств в кабеле интерфейса. Приоритеты для запросов от каналов устанавливаются в соответствии с типами последних. Приоритеты селекторных каналов установлены в соответствии с их номерами. Приоритет мультиплексного канала зависит от модели ЭВМ (в ЕС-1033 мультиплексный канал обладает высшим приоритетом).

При прерывании текущее ССП запоминается в ОП по адресу 38₁₆ в качестве старого ССП (прил. 10), а ССК, связанное с прерыванием, — по адресу 40₁₆. Затем из ОП по адресу 78₁₆ выбирается новое ССП, и работа продолжается в состоянии, указанном этим ССП (п. 5.2). Код прерывания идентифицируется адресом канала и устройства в старом ССП в разрядах [16/23] и [24/31] соответственно (прил. 8). В УСК, связанном с прерыванием, идентифицируется причина, вызвавшая его.

КАНАЛЫ ВВОДА-ВЫВОДА

12.1. ОРГАНИЗАЦИЯ КАНАЛОВ

ЭВМ ЕС-1033 содержит в своем составе (рис. 135) общий канал (ОК) с блоком пультовых режимов и диагностики (БПРД), мультиплексный канал (МК) с памятью мультиплексного канала на 128 неразделенных подканалов и три селекторных канала СК1, СК2, СК3. Мультиплексный канал обеспечивает в мультиплексном режиме скорость передачи данных до 50 кбайт/с и в монопольном режиме — последовательную работу устройств со скоростью до 300 кбайт/с. Любой из селекторных каналов обеспечивает последовательную работу устройств со скоростью до 800 кбайт/с.

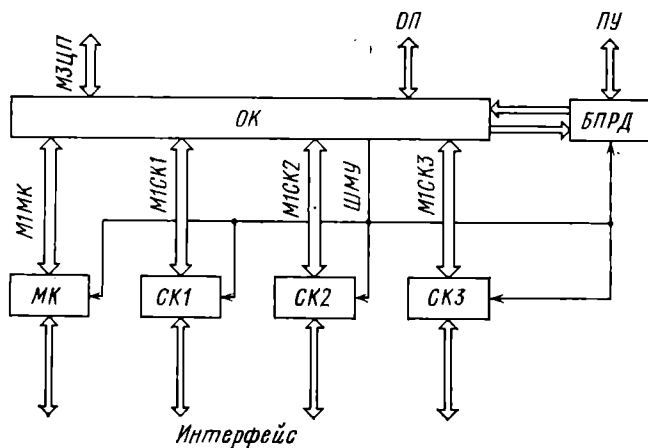


Рис. 135. Структурная схема каналов ввода-вывода

Основной особенностью структурной организации каналов модели ЕС-1033 является реализация связей с помощью нескольких информационных магистралей, через которые осуществляется связь между их узлами и блоками. Магистраль М1 является выходной внешней магистралью канала, остальные — внутренними. Через внешние магистрали М1 МК, М1 СК1, М1 СК2 и М1 СК3 мультиплексный и селекторный каналы обмениваются информацией с ОК, который, в свою очередь, обменивается информацией (данными) с ОП и командной информацией — с процессором через его информационную магистраль М3 ЦП.

Управление работой общего и мультиплексного каналов ввода-вывода осуществляется микропрограммно. В селекторном канале управление микропрограммно-аппаратурное. Управляющая микропрограммная информация хранится в постоянной памяти микропрограмм каналов (п. 1.7), которая состоит из двух унифицированных блоков БОП 2048-33 (БОП4 и БОП5). Память микропрограмм входит в состав ОК. Для микропрограммной реализации операций ввода-вывода используются стандартные для многих

операций блоки микропрограмм — подпрограммы. Каждая из них может управлять (или управляться) любой другой подпрограммой. Такой подход позволил в значительной степени сократить объем управляющей микропрограммной информации.

Аппаратурное управление реализовано с помощью последовательно включенных управляющих триггеров, образующих тактовую цепочку. Установка и сброс этих триггеров происходит по синхросериям С1 и С2. Запуск тактовой цепочки производится микропрограммно, после чего она работает независимо, вырабатывая необходимую последовательность управляющих сигналов.

Общий канал осуществляет связь каналов с ОП и ЦП, обеспечивая коммутацию потоков информации, а также связывает каналы с блоком пультовых режимов и диагностики. Кроме того, ОК обеспечивает микропрограммное управление остальными каналами. По двунаправленной шине микропрограммного управления (ШМУ) происходит передача микроприказов, информации о состоянии схем, регистров и управляющих триггеров (для их анализа с целью модификации адреса следующей микрокоманды при ветвлении микропрограммы) и другой управляющей информации. Управление в ОК реализовано микропрограммно.

Мультиплексный канал осуществляет обмен данными между ОП и ВУ через коммутатор ОК, обеспечивая одновременную работу 128 неразделенных подканалов. Для этого он хранит информацию о состоянии канала, использует и модифицирует управляющую информацию активного подканала, обеспечивает буферное запоминание текущих параметров операции ввода-вывода, начатой в подканале, а также обменивается ею с интерфейсом. Управление в МК реализовано микропрограммно.

Селекторный канал также осуществляет обмен данными между ОП и ВУ через коммутатор ОК, для чего он хранит и использует различную управляющую информацию. Он имеет только один подканал и работает в монопольном режиме. Для увеличения скорости передачи данных в СК применяется смешанное аппаратно-микропрограммное управление.

12.2. ОБЩИЙ КАНАЛ

Общий канал обеспечивает связь между процессором и каналами при выполнении команд ввода-вывода, прерываний от ввода-вывода и записи состояния при обнаружении ошибок в работе ОК. При обмене данными между ОП и каналами ОК обеспечивает обращение к ОП и подключение к ней соответствующего канала для передачи (приема) данных. В процессе реализации микропрограмм как собственных, так и каналов ввода-вывода ОК производит выборку микрокоманд из памяти микропрограмм, входящей в его состав, и модификацию базовых адресов микрокоманд в соответствии с данными для анализа, поступающими из каналов ввода-вывода.

В ОК применена магистральная организация связей как на уровне функциональных блоков, так и на уровне его внешних связей с каналами ввода-вывода и ЦП (рис. 136).

Связь между ЦП и ОК осуществляется через буферный регистр связи с процессором (РБСП) и информационную магистраль процессора МЗ ЦП. Связь между каналами ввода-вывода и ОП, а также между каналами и ОК осуществляет блок коммутации числовой информации (БК ЧИ) через внешние магистрали каналов М1 МК, М1 СК1, М1 СК2, М1 СК3. Управляющие сигналы

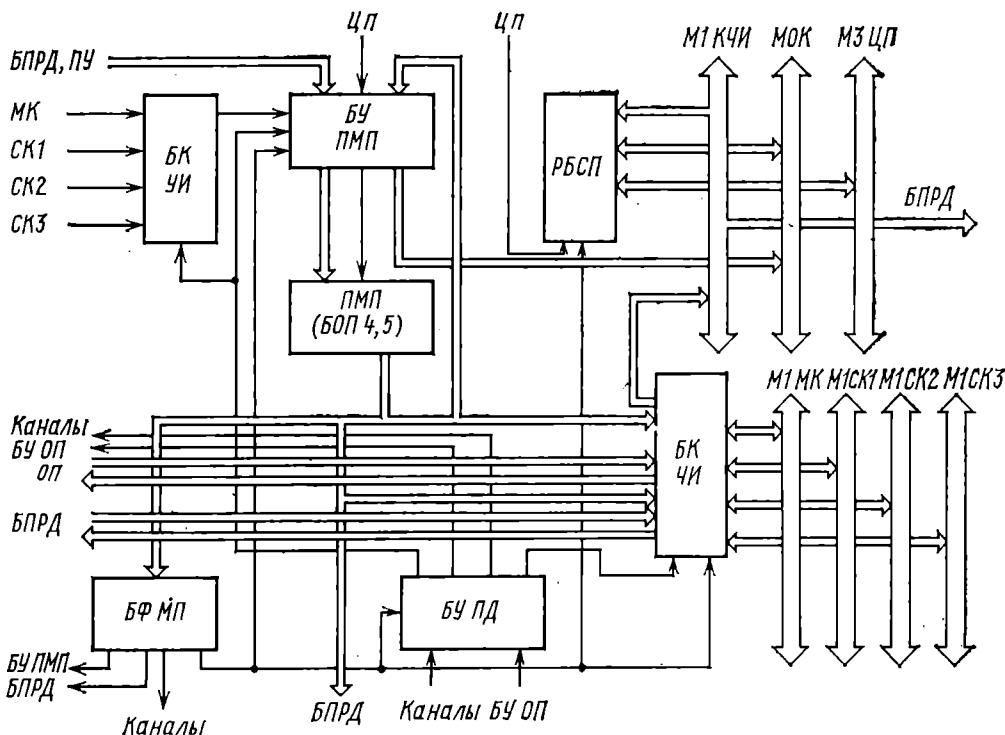


Рис. 136. Структурная схема общего канала

для его работы формируются блоком управления передачей данных (БУ ПД). Магистрали МОК и М1 КЧИ являются внутренними.

Выборка микрокоманд из памяти микропрограмм (ПМП) производится блоком управления (БУ ПМП). Содержимое операционных полей (см. ниже) считанных из ПМП микрокоманд поступает в блок формирования микроприказов (БФ МП), откуда микроприказы поступают в МК, СК, БПРД и в ОК. Базовый адрес поступает в БПРД и БУ ПМП. Последний модифицирует базовый адрес, выполняя анализ информации, поступающей от каналов ввода-вывода через блок коммутации управляющей информации (БК УИ). Константа из поля микрокоманд поступает на вход БК ЧИ и передается через него в соответствующий канал.

Блок памяти микропрограмм хранит управляющую информацию, необходимую для работы каналов ввода-вывода, и состоит

из двух унифицированных блоков БОП4 и БОП5 (п. 1.7). Микрокоманда каналов имеет 64 информационных и 2 контрольных разряда. Считается, что регистры информации блоков БОП4 и БОП5 составляют один 64-разрядный регистр микрокоманд канала (РМКК).

Микрокоманды канала разделяются на четыре типа в зависимости от значения управляющего поля Π (разряды [62/63] рис. 137).

В микрокомандах всех типов поле L (РМКК [41/49]) определяет базовый адрес, т. е. девять старших разрядов адреса следующей микрокоманды. Два младших разряда адреса следующей микрокоманды формирует БУ ПМП в зависимости от результатов проведения анализов переходов. Коды этих анализов содержатся

| Разряды | 0/5 | 6/10 | 11/15 | 16/20 | 21/25 | 26/30 | 31/35 | 36/40 | 41/49 | 50/55 | 56/61 | 62/63 |
|---------|-----------|-------------------|-------|-------|-----------|-------|-------------------|-------------------|--------------------------------------|-------------------------------|-------|-------|
| Поля | А | Б | В | Г | Д | Е | Ж | И | Л | М | Н | П |
| Поля | Константа | Операционные поля | | | | | | | Базовый адрес следующей микрокоманды | Поля М и Н анализов переходов | 00 | |
| | Константа | Операционные поля | | | | | | | | | 01 | |
| | Константа | Операционные поля | | | Константа | | Операционное поле | 10 | | | | |
| | Константа | | | | | | | Операционное поле | | | 11 | |

Рис. 137. Распределение полей микрокоманды каналов

в полях M и N (РМКК [50/55] и РМКК [56/61]) и поступают из ПМП в БУ ПМП. Туда же через БК УИ поступает информация для анализов. Каждое из полей M и N позволяет, в принципе, проверить выполнение до 64 различных условий. Сформированный таким образом адрес следующей микрокоманды запоминается в регистре адреса микрокоманды (РА МК), который входит в состав БУ ПМП.

Микрокоманда типа 1 ($\Pi = 00$) содержит поле константы в РМКК [0/5] и семь операционных полей в РМКК [6/40], содержащих коды микроприказов. Константа может определять двухразрядный номер канала, двухразрядный признак результата и его контрольный бит, двухразрядный номер слова ПМК, четырехразрядную константу для формирования маркеров начального и конечного байтов при обращении к ОП по фиксированным адресам. В этом поле также может быть задана 6-разрядная константа, определяющая функцию АЛУ и операнд B для мультиплексного канала при модификации содержимого регистров адреса УСК, данных или количества байт.

Микрокоманда типа 2 ($\Pi = 01$) содержит поле константы в РМКК [0/10] и шесть операционных полей в РМКК [11/40]. В поле константы задается адрес возврата микропрограммы. Содержимое этого поля поступает в БУ ПМП для записи его в регистры адреса возврата РАВ и РАВМ.

Микрокоманда типа 3 ($\Pi = 10$) содержит два поля констант в РМКК [0/5] и РМКК [21/35] и четыре операционных поля в РМКК [6/20] и РМКК [36/40]. Назначение первого поля константы такое же, как и в микрокоманде типа 1. Второе поле содержит 8-разрядную константу и контрольные разряды. Эта константа используется для анализа байта состояния УВУ в программе загрузки и при выполнении цепочки команд. Она выдается на магистраль М1 соответствующего канала через БК ЧИ.

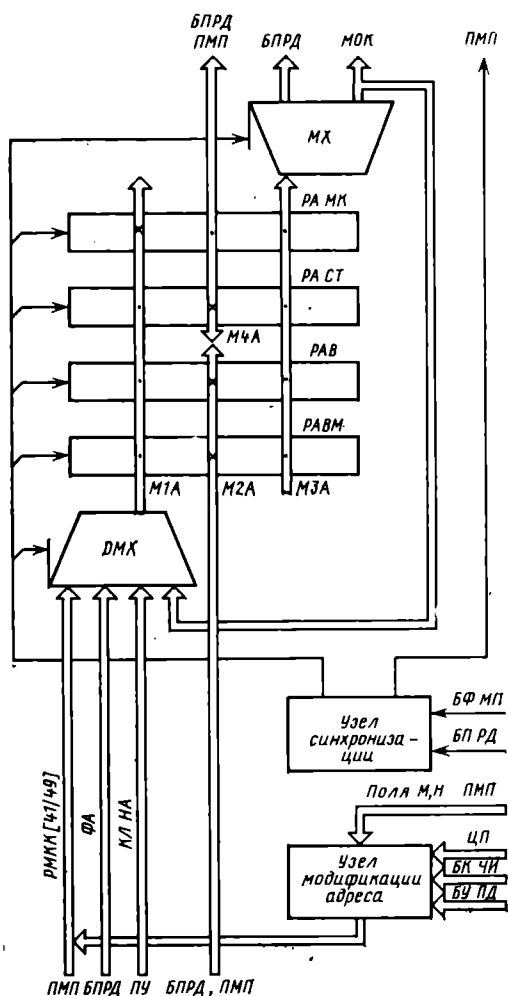


Рис. 138. Структурная схема блока управления памятью микропрограмм

Микрокоманда типа 4 ($\Pi = 11$) имеет одно операционное поле и поле РМКК [0/35] 32-разрядной константы (четыре младших разряда поля Ж определяют значения контрольных бит каждого байта константы). Эта константа используется для формирования фиксированных адресов при обращении к ОП и в тестах локализации неисправностей при автономной проверке. Константа выдается на магистраль М1 соответствующего канала через БК ЧИ.

Блок формирования микроприказов дешифрирует коды операционных полей микрокоманд, вырабатывая микроприказы, необходимые для управления работой каналов в соответствии с заданным алгоритмом. Микроприказы выдаются одновременно во все каналы на соответствующие функциональные элементы и схемы. Однако выполнение любого из них в каждом из каналов стро-

бируется соответствующим селективным сигналом работы канала РАБ МК, РАБ СК1, РАБ СК2 и РАБ СК3. Эти сигналы формируются в БУ ПД.

Блок управления ПМП (рис. 138) содержит входной ДМХ и выходной МХ коммутаторы адресной информации, регистровую адресную память, узел модификации адреса и узел синхронизации. Регистры памяти связаны между собой и коммутаторами через адресные магистрали М1А, М2А, М3А и М4А. Эти магистрали являются внутренними и не имеют элементов памяти. Сама регистровая память построена на ИС К155ХЛ1. Каждый из

регистров имеет связь по цепям записи (на рисунке эта связь отмечена крестиком) с одной магистралью и связи по цепям считывания (отмечена точкой) с двумя другими. В состав адресной памяти входят: РА МК — регистр адреса текущей микрокоманды; РАСТ — регистр старого адреса, т. е. адреса предыдущей микрокоманды; РАВ—регистр адреса возврата; РАВМ —регистр адреса возврата МК.

Код адреса текущей микрокоманды выдается в ПМП из РА МК через адресную магистраль М4А. Запись кода адреса в РА МК происходит с магистрали М1А. Информация на нее может быть либо выдана через входной коммутатор адреса, либо считана из РАСТ, РАВ и РАВМ. На входе коммутатора код адреса может быть сформирован по значению поля РМКК [41/49] (9 старших разрядов) и по результату анализа переходов (2 младших разряда). Последние формируются узлом модификации адреса, на входы которого поступают коды анализов из полей М и Н РМКК [50/61] и информация для анализа из БК УИ, БУ ПД, ЦП. На входной коммутатор код адреса может поступать также из БПРД (фиксированные адреса — ФА), с клавиатуры начального адреса пульта управления (КЛ НА) и с магистрали общего канала, на которую она поступает из РБСП.

Регистр старого адреса хранит адрес выполненной микрокоманды. Информация из РА МК переписывается в РАСТ перед записью в РА МК нового адреса. Эта информация используется для адресации отдельных процедур при автономной проверке и наладке каналов с помощью тестов локализации неисправностей, а также для хранения адреса микрокоманды, предшествовавшей пультовой процедуре (п. 13.4).

В РАВ хранится адрес начала подпрограммы, которой нужно передать управление после выполнения определенной части алгоритма. Регистр адреса возврата используется при выполнении микропрограмм селекторных каналов, микропрограммных ТЛН, а также в режиме сканирования при автономной проверке ПМП. Адрес в РАВ может быть записан из поля константы микрокоманды или с выхода узла сканирования из БПРД через магистраль М2А.

Аналогично используется при выполнении микропрограмм мультиплексного канала РАВМ.

Информация из любого из этих регистров через магистраль М3А и выходной коммутатор может быть выдана в БПРД для индикации содержимого или на МОК при записи состояния ОК или при выполнении других диагностических процедур.

Узел синхронизации формирует сигналы управления чтением и записью регистровой памяти, коммутаторами, а также формирует управляющие сигналы, необходимые для считывания микрокоманды из БОП4 и БОП5 (п. 1.7).

Буферный регистр связи с процессором РБСП (см. рис. 136) служит для передачи информации из процессора в каналы и об-

ратно при выполнении команд ввода-вывода, обработке прерываний ввода-вывода, а также при выполнении диагностических процедур. Поэтому РБСП может работать как по микроприказам, формируемым ЦП, так и по микроприказам, формируемым ОК. Регистр РБСП имеет связи по записи и чтению с информационной магистралью МЗ центрального процессора — МЗ ЦП, с магистралью общего канала и связь только по записи — с магистралью М1 коммутатора числовой информации М1 КЧИ.

Запись в РБСП с МЗ ЦП происходит по микроприказу РБСП := МЗ ЦП, который формируется в процессоре. При выполнении команд ввода-вывода в этом случае в РБСП [0/7] записывается код операции команды, а в РБСП [22/23] и РБСП [24/31] — номера канала и ВУ соответственно. При выполнении команды ДИАГНОСТИКА по этому микроприказу в РБСП записывается управляющее слово этой команды (УСКД).

Запись в РБСП с МОК происходит по микроприказу РБСП := МОК, который может быть сформирован как блоком диагностики ЦП, так и ОК. В случае ошибки ОК ЦП выдает микроприказ для того, чтобы обеспечить запись и передачу в ЦП содержимого адресных регистров ПМП и состояния служебных триггеров для записи состояния ОК в случае аппаратурной ошибки. При работе ОК запись в РБСП иницируется каналом. Например, при обработке прерывания адрес канала предварительно считывается из регистра номера канала на МОК. Далее он записывается в РБСП для передачи в ЦП.

Запись в РБСП с М1 КЧИ происходит по микроприказу РБСП := КЧИ, который формируется в ОК. Например, при обработке прерывания адрес ВУ, считанный из регистра адреса устройства МК или СК на М1 КЧИ, записывается в РБСП для передачи его в ЦП.

Содержимое РБСП может быть выдано на МЗ процессора по микроприказу МЗЦП := РБСП, который формируется в ЦП. При этом в процессор передается признак результата операции из РБСП [3/4] после выполнения команды ввода-вывода или номер канала и адрес устройства из РБСП [22/23] и РБСП [24/31] соответственно при обработке прерывания. Кроме того, через РБСП происходит передача с МОК состояния регистров и служебных триггеров для записи состояния ОК в случае ошибки в нем.

Содержимое РБСП может считываться также на МОК по микроприказу МОК := РБСП, формируемому ОК, для передачи в него кода операции, номеров канала и ВУ при выполнении команды ввода-вывода или УСКД при выполнении команды ДИАГНОСТИКА.

Коммутатор числовой информации обеспечивает ее выдачу на информационную магистраль М1 либо мультиплексного, либо одного из селекторных каналов СК1—СК3 (рис. 139). Ею могут быть либо данные или командная информация, считанные из ОП,

либо константа из ПМП или информация с клавиатуры РИ ПУ. Кроме того, КЧИ обеспечивает выдачу в ОП информации с одной из названных четырех информационных магистралей при обмене данными между ВУ и каналом или при записи ССК.

Считанная из ОП информация поступает на входной коммутатор МХ. При наличии управляющего сигнала $M1 := ОП$ и в зависимости от кода номера канала при работе с оперативной памятью (НК ОП) информация будет поступать на соответствующую магистраль. Аналогично магистраль, с которой информация

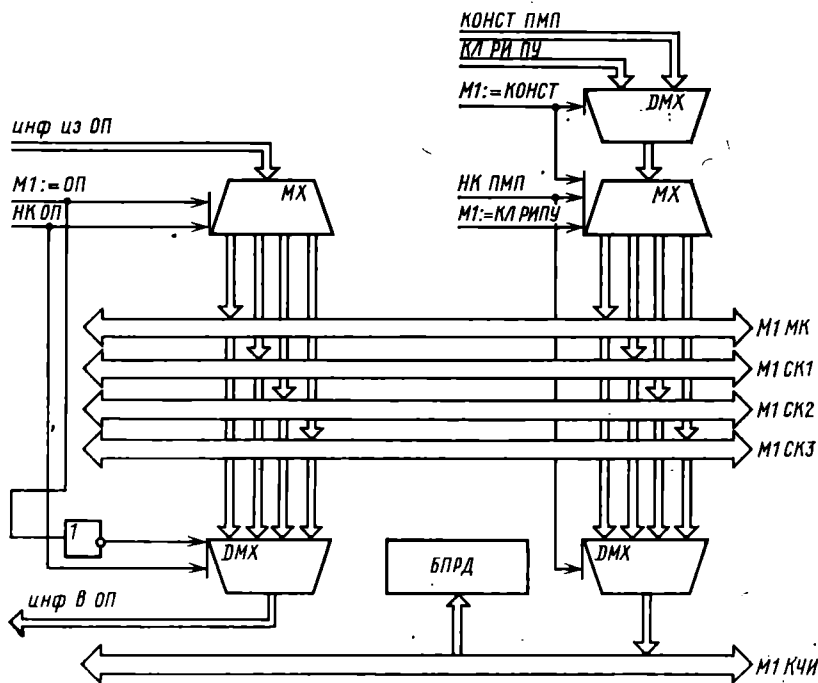


Рис. 139. Коммутатор числовой информации

поступает для записи в ОП, определяется кодом НК ОП. Этот код подается на управляющие входы выходного коммутатора DMX.

Константа из ПМП (КОНСТ ПМП) и информация с клавиатуры РИ ПУ (КЛ РИ ПУ) поступает на входы коммутатора-демультиплексора DMX два направления на одно. При наличии управляющего сигнала $M1 := КОНСТ$ на выходе коммутатора будет информация КОНСТ ПМП, в противном случае постоянно будет присутствовать информация КЛ РИ ПУ. При наличии управляющего сигнала $M1 := КОНСТ$ или $M1 := КЛ РИ ПУ$ информация через коммутатор-мультиплексор МХ одного направления на четыре будет поступать на одну из четырех информационных магистралей в зависимости от значения кода НК ПМП. Этот же код определяет, с какой из названных магистралей информация может быть передана через коммутатор-демультиплексор DMX четыре направления на одно на магистраль M1 КЧИ.

Магистраль М1 КЧИ является выходной магистралью КЧИ и обеспечивает обмен информацией между каналами и ЦП, БПРД, ПМП (см. рис. 136, 139).

Блок коммутации управляющей информации подключает к ПМП данные для анализа того канала, микропрограмма которого выполняется в данное время. Номер канала хранится в регистре номера канала, входящем в состав БУ ПД.

Блок управления передачей данных (рис. 140) формирует управляющие сигналы для БК ЧИ, а также сигналы синхронизации и управления работой каналов с ОП. Номер канала, в который выдается соответствующая информация, определяется кодом номера канала.

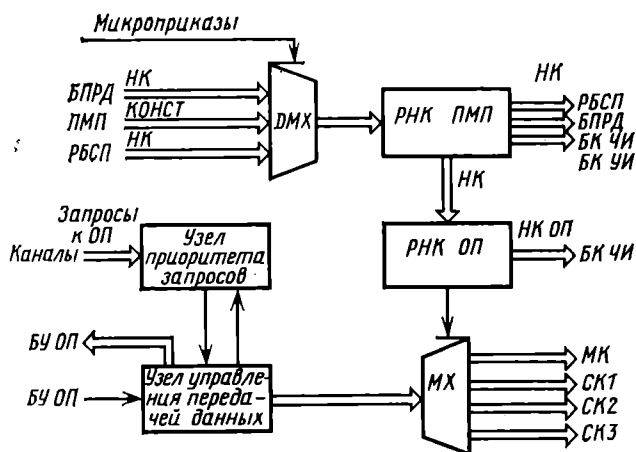


Рис. 140. Блок управления передачей данных

Код номера канала может быть записан в два регистра номера канала: один — при работе с ПМП (РНК ПМП), другой — при работе с ОП (РНК ОП). Запись в РНК ПМП происходит через входной коммутатор DMX. При этом номер канала может поступать из БПРД с клавиатуры номера канала — при выполнении пультовых режимов и ТЛН, из ПМП (из поля констант микрокоманды) — при выполнении микропрограмм обслуживания запросов от каналов и при обработке прерываний от ввода-вывода и из РБСП при выполнении команд ввода-вывода. При обработке прерываний и записи состояния ОК при ошибке в нем номер канала выдается в РБСП, для целей индикации — в БПРД, для выбора соответствующего направления коммутации в — БК ЧИ, для передачи информации из соответствующего канала в БУ ПМП, для выполнения анализов — в БК УИ и в РНК ОП — для записи в него кода номера канала при работе с ОП.

Узел приоритета запросов принимает от каналов и от микропрограммы запросы на обслуживание ОП, формирует согласно приоритету код номера канала и помещает его в РНК ОП. В случае микропрограммного запроса код номера канала поступает из РНК ПМП.

Узел управления передачей данных в зависимости от приоритета запроса к ОП формирует сигналы управления и синхронизации работы каналов с ОП и через выходной коммутатор выдает их в соответствующий канал. Одновременно он формирует и выдает в БУ ОП сигналы обращения к ОП.

Все запросы к ОП от МК, а также запросы от СК к ОП по фиксированным адресам формируются микропрограммно. Остальные запросы СК формируются аппаратурно тактовыми цепочками и могут разделяться на запросы за данными и за УСК. Высшим приоритетом обладают запросы за данными от СК1, затем запросы за данными от СК2 и СК3, запросы за УСК от СК1, СК2 и СК3. Низшим приоритетом обладают запросы от микропрограммы. Сформированный каналом запрос к ОП устанавливает в БУ ОП триггер общего запроса каналов (ТОЗК), обеспечивая тем самым приоритет каналов перед ЦП на обслуживание запроса к ОП. Адрес ОП, выдаваемый каналом, записывается в регистр адреса канала (п. 3.4), после чего БУ ОП организует обращение к ОП. При приеме запроса от канала устанавливается триггер очередности работы канала, который информирует канал о том, что его запрос обслуживается. При этом каналу разрешаются прием считанной из ОП информации, смена адреса обращения и выработка следующего запроса.

Алгоритм работы ОК реализован подпрограммой ожидания запросов (ПОЗ) и подпрограммами обслуживания запросов ЦП (рис. 141), МК СК, а также запросов на окончание выполнения команд ввода-вывода и обработки прерываний.

Запросы от процессора и каналов поступают асинхронно по отношению друг к другу и фиксируются в соответствующих триггерах. Эти триггеры входят в состав условных регистров регистровой памяти МК и СК или являются принадлежностью оборудования ЦП. Подпрограмма ожидания запросов анализирует состояние этих триггеров в три этапа: сначала состояние триггеров запросов от СК1 и МК, затем от СК2 и СК3 и, наконец, от ЦП. При наличии двух запросов на одном этапе анализа высшим приоритетом обладают запросы от СК1 и СК2. Когда будет обнаружен один из этих запросов, происходит выход из подпрограммы ожидания на подпрограмму обслуживания соответствующего запроса общим каналом. Во время ее выполнения в РНК записывается номер того канала, микропрограммы которого будут в дальнейшем выполняться под управлением ОК. При этом происходит установка триггера работы ОК с каналом ввода-вывода. Данная подпрограмма передает управление микропрограммам канала ввода-вывода, которые соответствуют типу запроса. Обработка запроса в канале ввода-вывода оканчивается сбросом ТРК и переходом на программу ожидания запросов.

По запросу от процессора каналы могут выполнять либо микропрограмму команды ввода-вывода, либо микропрограмму обслуживания прерывания.

При выполнении команды ввода-вывода ЦП записывает в РБСП код операции, номера канала и устройства, устанавливает в единичное состояние триггер ввода-вывода (ТВ/В). После этого микропрограмма процессора закидывается на анализе состояния этого триггера. После того как канал запишет в РБСП признак результата и сбросит ТВ/В, выполнение микропрограммы команды процессором будет продолжено параллельно с начавшейся операцией ввода-вывода. Процессор записывает признак результата

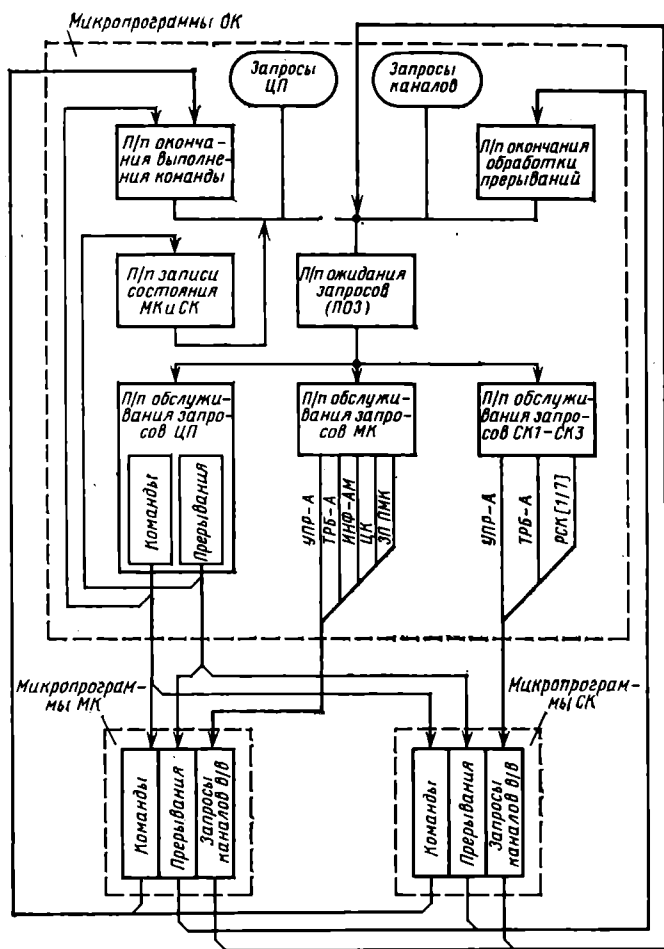


Рис. 141. Структурная схема связей микропрограмм каналов

в РБСП [34/35] и заканчивает выполнение текущей команды, переходя к выборке следующей (п. 5.3). Подпрограмма ожидания запросов ОК анализирует состояние ТВ/В и при $ТВ/В=1$ передает управление подпрограмме обслуживания запроса ЦП. Последняя идентифицирует канал по его номеру, переписывает код номера канала из РБСП в РНК ПМП и анализирует четность принятой информации. При обнаружении ошибки четности управление передается подпрограмме окончания выполнения команды, которая в этом случае формирует и записывает в РБСП код признака результата, равный трем, и сообщает ЦП об окончании вы-

полнения команды ввода-вывода сбросом ТВ/В. При отсутствии ошибок командной информации анализируется состояние канала (РСК [8, 9, 11]), после чего управление передается соответствующим подпрограммам ОК.

Каждый из каналов имеет регистр состояния канала РСК [0/11], который состоит из двух частей. РСК [0/7] содержит байт состояния канала, который формируется каналом и может быть записан в ССК[40/47], т. е. может быть доступен программе. Содержимое остальной части также формируется каналом в ходе выполнения операции ввода-вывода и используется только для микропрограммного анализа:

РСК[10] = 1 означает, что канал находится в режиме связи с системой (обрабатывает прерывание или выполняет команду ввода-вывода);

РСК[11] = 1 — канал выключен. Управление передается подпрограмме окончания выполнения команды;

РСК[9] = 1 — канал работает в монопольном режиме. Если выполняется команда НАЧАТЬ ВВОД-ВЫВОД, ПРОВЕРИТЬ ВВОД-ВЫВОД или ПРОВЕРИТЬ КАНАЛ, то управление передается подпрограмме окончания выполнения команды. Если выполняется команда ОСТАНОВИТЬ ВВОД-ВЫВОД, управление передается подпрограмме выполнения команды соответствующего канала, МК или СК1—СК3.

РСК[8] = 1 — канал хранит прерывание. Если выполняется команда ПРОВЕРИТЬ КАНАЛ, управление передается подпрограмме окончания выполнения команды ввода-вывода. В остальных случаях, когда РСК [8, 9, 11] = 0, управление передается подпрограммам выполнения команды соответствующих каналов (МК, СК1—СК3).

Если канал готов к выполнению команд ввода-вывода и выполняется команда ПРОВЕРИТЬ КАНАЛ, управление передается подпрограмме окончания выполнения команды. Эта подпрограмма формирует соответствующие признаки результата выполнения команды, записывает их в РБСП [2/3] и сообщает процессору об окончании выполнения команды ввода-вывода каналом выдачей в ЦП сигнала ТВ/В := 0. Признак результата, формируемый подпрограммой окончания выполнения команды в общем канале, имеет следующие значения:

3 — канал выключен;

2 — канал работает в монопольном режиме;

1 — канал хранит прерывание;

0 — канал свободен.

После записи признака результата в РБСП и сообщения процессору об окончании команды ввода-вывода управление передается подпрограмме ожидания запросов.

Прерывания от ввода-вывода служат для того, чтобы информировать процессор о выполнении различных этапов операции ввода-вывода или о каких-то внешних событиях в УВВ. Запросы

на прерывание воспринимаются процессором, если они разрешены маской программы в ССП [0/3] (прил. 7). В этом случае прерывания фиксируются в регистре прерываний (РФПД) процессора (п. 3.2), который начинает выполнять микропрограмму обработки прерывания (п. 4.3). При этом устанавливается в единичное состояние триггер блокировки (ТБ) и микропрограмма зацикливается на его анализе. Единичное состояние ТБ поступает в ОК в качестве сигнала запроса на прерывание от ЦП. По этому сигналу подпрограмма ожидания запросов осуществляет переход к подпрограмме обслуживания запросов процессора. В этом случае подпрограмма опрашивает состояния триггеров, в которых зафиксированы прерывания каналов, и в соответствии с приоритетом записывает адрес канала из поля констант в РНК ПМП блока управления данными. Наивысшим приоритетом обладает МК, низшим — СКЗ. Затем подпрограмма обработки прерывания анализирует состояние РСК [5/6] на отсутствие ошибок управления каналом и ошибок интерфейса. При наличии ошибок управление передается подпрограмме записи состояния соответствующего канала на момент ошибки. При отсутствии ошибок управления и интерфейса происходит переход к подпрограммам обработки прерываний соответствующих каналов.

После выполнения подпрограмм каналами управление передается ОК в подпрограмму окончания обработки прерываний. Эта подпрограмма из РНК ПМП блока управления данными записывает в РБСП номер канала, с магистрали М1 канала через БК ЧИ адрес устройства, а также сообщает процессору о завершении каналом обработки прерывания сигналом ТБ := 0. После этого управление передается подпрограмме ожидания запросов. Сигнал ТБ := 0 сбрасывает триггер блокировки, позволяя процессору продолжить микропрограмму обработки прерывания. Микропрограмма завершается записью в ОП текущего ССП и выборкой нового (прил. 10).

Мультиплексный канал формирует запросы УПР-А и ИНФ-А по сигналам идентификации УПР-А и ИНФ-А в монопольном режиме работы и, если задана ЦК, запрос на подпрограмму переборки УВУ по цепочке команд. Требование записи в ПМК мультиплексный канал выставляет в случае снятия абонентом сигнала РАБ-А, когда определен монопольный режим. Требование по сигналу «задержка» ответа мультиплексный канал выставляет при отсутствии обмена в течение 30 с в монопольном режиме работы.

Селекторный канал формирует запрос по сигналу идентификации УПР-А при получении его из интерфейса ввода-вывода после выполнения текущей команды канала и по значению РСК [1/7], если при выполнении операции ввода-вывода возникли ошибки канала.

Запрос по сигналу ТРБ-А устанавливается в каналах при получении этого сигнала из интерфейса.

12.3. МУЛЬТИПЛЕКСНЫЙ КАНАЛ

Мультиплексный канал предназначен для параллельного обслуживания медленно действующих ВУ в режиме разделения времени и имеет 128 неразделенных подканалов. Все подканалы равноценны по отношению друг к другу. Для каждого из них в памяти МК отведена область, называемая памятью подканала. Управление МК микропрограмное, управляющая информация хранится в памяти микропрограмм, входящей в состав ОК.

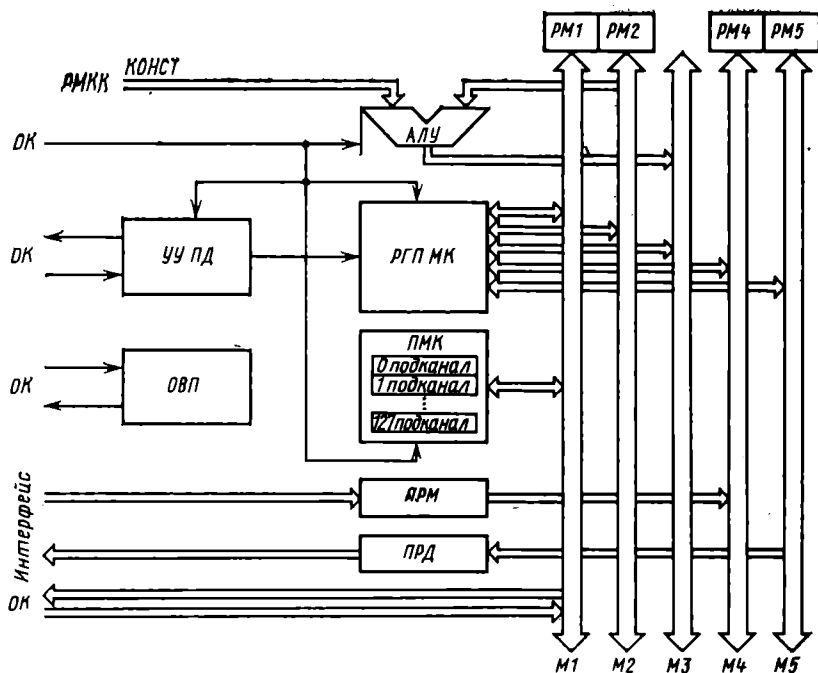


Рис. 142. Структурная схема мультиплексного канала

Канал имеет магистральную организацию связей как на уровне функциональных блоков, так и на уровне связей с ОК. Все передачи информации внутри МК осуществляются через информационные магистрали М1—М5 (рис. 142). Элементами памяти магистралей являются четыре магистральных регистра РМ1, РМ2, РМ4, РМ5, которые осуществляют прием считанной на магистраль информации и ее хранение. В исходное состояние магистрали приводятся специальным микроприказом. Магистраль М3 не имеет элементов памяти. Магистраль М1 является внешней магистралью, через которую МК обменивается информацией с ОК. Обмен информацией канала с ВУ при приеме информации осуществляется через стандартный интерфейс ввода-вывода, приемники ПРМ и магистраль М4, а при передаче информации в ВУ — через интерфейс, передатчики ПРД, магистраль М5.

Регистровая память (РГП) канала хранит всю информацию, необходимую для работы канала и активного в данное время под-

канала, а также передаваемые (принимаемые в ВУ и из ВУ) данные. Память МК (ПМК) хранит информацию, относящуюся к каждому из подканалов. Передачей данных управляет блок управления БУ ПД. Модификация адресов обращения к ОП осуществляется с помощью АЛУ. Его выходной магистралью является МЗ. В состав МК входит также ограничитель времени

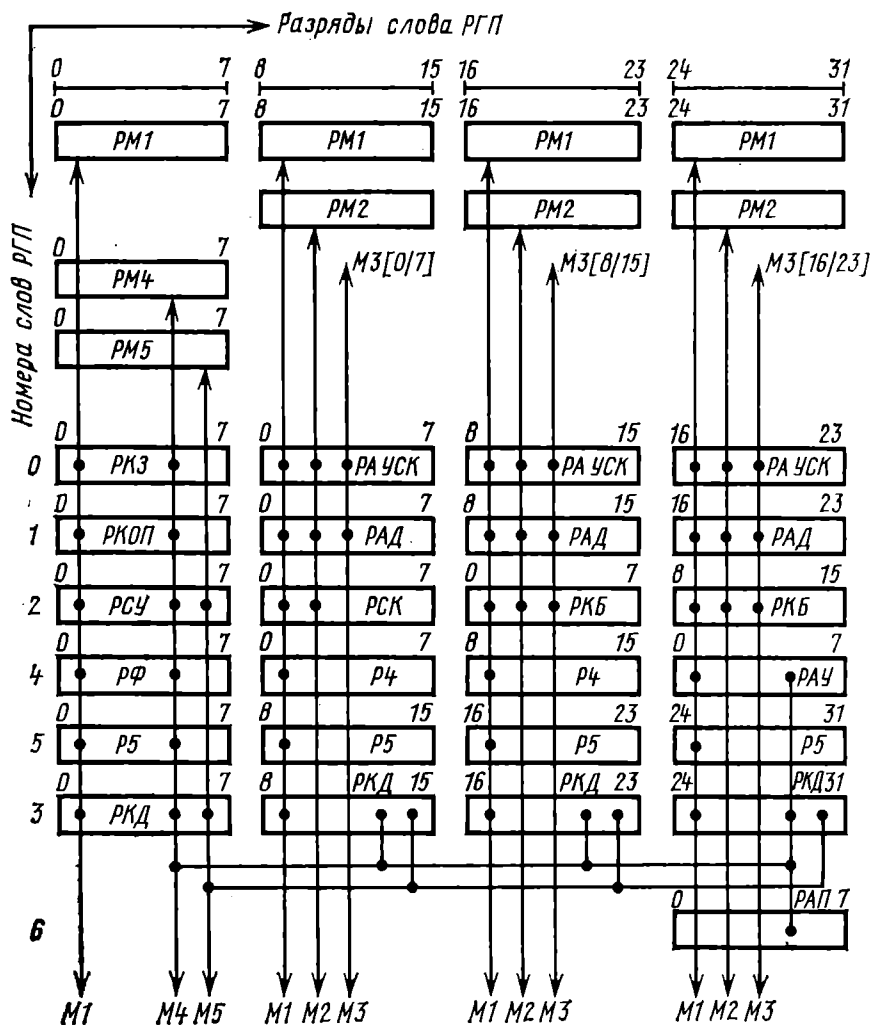


Рис. 143. Структура регистровой памяти МК

простая ОВП. Центральным блоком МК является РГП вместе с информационными магистралями и РМ (рис. 143). Регистровая память построена на ИС К155ХЛ1. Связь регистра с соответствующей магистралью на рисунке показана точкой.

Магистраль М1 имеет 32 информационных разряда и связана со всеми регистрами МК. Магистраль М3 имеет 24, а М4 — 8 информационных разрядов. Каждые восемь разрядов магистрали снабжены контрольным разрядом. Содержимое РГП пословно

через магистраль М1 и БПРД может быть проиндицировано на соответствующие линейки пульта управления.

Регистровая память имеет следующий состав:

РКЗ [0/7] — регистр ключей защиты (нулевое слово РГП — РГП0/7]). В него засылается код ключа защиты из соответствующей зоны АСК при считывании последнего из ОП. Ключ защиты передается в БУ ОП при каждом обращении канала в ОП, кроме обращения по фиксированным адресам;

РАУСК [0/23] — регистр адреса УСК (РГП0 [0/23]). После выборки АСК из ОП в РА УСК пересылается код адреса первого УСК в цепочке команд. Для выборки УСК из ОП код адреса через магистраль М1 и коммутатор ОК передается в БУ ОП, откуда он поступает в ОП. Если исполняется цепочка команд, то при выборке последующих УСК код адреса УСК модифицируется. Так как УСК имеет формат двойного слова, а формат обращения к ОП составляет одно слово, модификация кода адреса выполняется в два этапа. Сначала происходит модификация кода адреса на +4, выбирается первое слово УСК и записывается в соответствующие регистры РГП. После этого код адреса дополнительно модифицируется на +4, выбирается второе слово УСК и также записывается в РГП. Для модификации код адреса считывается из РА УСК на М2 и с нее поступает на один из входов АЛУ. На другой его вход поступает константа 0100 из РМКК. Код, определяющий функцию АЛУ $F := A + B$, поступает также из РМКК. Модифицированный код с выхода АЛУ через его выходную магистраль М3 снова записывается в РА УСК;

РКОП [0/7] — регистр кода операции (РГП1 [0/7]). Это условный регистр, объединяющий ряд регистров и триггеров. Непосредственно код операции определяют ТПН и ТОН — триггеры прямого и обратного направлений (РГП1 [3/4]). Кроме того, в состав РКОП входят регистр положения байтов в слове ОП — РПБ [0/1] (РГП1 [0/1]), регистр состояния подканала РП [8/9] (РГП1 [6/7]) и несколько служебных триггеров. Регистр состояния подканала является как бы продолжением регистра состояния устройства РСУ (см. ниже). Он устанавливается и анализируется только микропрограммно. В его разрядах запоминается следующая информация:

РП [8] = 1 — подканал хранит прерывание;

РП [9] = 1 — подканал находится в режиме передачи данных;

РАД [0/23] — регистр адреса данных (РГП1 [8/31]). Он хранит текущий адрес данных. После выборки УСК из ОП в него пересылается код адреса данных из УСК. Так как обмен данными между ОП и каналом происходит полными словами, то код адреса данных при этом модифицируется на +4 или -4 (при обратном чтении). Модификация происходит аналогично модификации кода адреса УСК;

РСУ [0/7] — регистр состояния устройства (РГП2 [0/7]). В этом регистре запоминается байт состояния устройства, полу-

ченный из ВУ. В подпрограмме записи ССК содержимое этого регистра записывается в ОП в ССК [32/39];

РСК [0/7] — регистр состояния канала (РГП2 [8/15]). Его содержимое формируется в ходе выполнения операции ввода-вывода и записывается в ССК [40/47] подпрограммой записи ССК;

РКБ [0/15] — регистр количества байтов (РГП2 [16/31]). После выборки УСК в нем запоминается содержимое поля счетчика — УСК [48/63]. В процессе обмена данными между ВУ и каналом его содержимое модифицируется на — 1 при передаче каждого байта. Модификация происходит аналогично модификации кода адреса УСК; РКД [0/31] — регистр коммутации данных (РГП3 [0/31]). Он предназначен для буферного хранения данных в операции ввода-вывода. При выводе из ОП в ВУ слово данных принимается из ОП в РКД. Прием происходит через коммутатор ОК и магистраль М1. Передача данных происходит из РКД побайтно через М5 в интерфейс. При этом модифицируется РПБ [0/1], отмечая положение переданного байта. Для передачи очередного слова данных модифицируется код адреса, слово из ОП принимается в РКД и процесс продолжается. При вводе данных в ОП происходит побайтное заполнение РКД из интерфейса через магистраль М5 и запись данных в ОП из РКД полным словом;

РФ [0/7] — регистр флажков (РГП4 [0/7]). Собственно сам регистр флажков занимает РГП4 [0/4], и в него записывается содержимое зоны признаков из УСК [32/36]. В РФ включен также триггер первого обращения к ОП—ТПО, который управляет регистром маркеров. Регистр маркеров указывает положение байтов, записываемых в ОП. РГП4 [5/6] содержит два плеча ТПО для того, чтобы его состояние не влияло на значение контрольного разряда РФ;

Р4 [0/15] — условный регистр (РГП4 [8/23]). В него включены триггеры, фиксирующие сигналы идентификации интерфейса, а также триггеры цепочки данных ТЦД, цепочки команд ТЦК и блокировки неправильной длины ТБНД;

РАУ [0/7] — регистр адреса устройства (РГП4 [24/31]). Он хранит код адреса устройства текущей операции ввода-вывода;

Р5 [0/31] — условный регистр (РГП5 [0/31]). Он состоит из триггеров фиксации ошибок байтов магистралей — ТОМ1 (0) — ТОМ1 [3], ТОМ2 (0) — ТОМ2 [2], ТОМ4, ТОМ5; триггеров ошибок АЛУ — ТО АЛУ (0) — ТОАЛУ [2], ТОВАЛУ (0) — ТОВАЛУ [2] (п. 13.2); регистра состояния канала РСК [8/11] и ряда служебных триггеров в РГП5 [28/31];

РАП [0/7] — регистр адреса подканала (РГП6 [24/31]). Регистр хранит код адреса того подканала, который вызывает прерывание в канале путем установки в единичное состояние РСК [8]. Содержимое этого регистра (код адреса ВУ) выдается в ЦП для записи в РССП [24/31] при обработке прерывания от ввода-вывода. Одновременно другие подканалы могут находиться в состоянии «Хранит прерывание», что определяется единичным значением

РП[8] в соответствующей этому подканалу области ПМК. В этом случае ВУ хранит байт состояния, отличный от нулевого, и выдает в канал запрос на прерывание от подканала сигналом ТРБ-А. Этот запрос будет воспринят каналом по окончании обработки предыдущего прерывания после сброса РСК [8].

Память мультиплексного канала (ПМК) предназначена для хранения текущей управляющей информации и состояний подканала в операциях ввода-вывода. Она построена на ИС К155РУ2 (п. 1.5), и ее объем соответствует количеству подканалов в МК. Управляющая информация и коды состояния одного подканала занимают в ПМК четыре 32-разрядных слова. После окончания сеанса связи ВУ с каналом в ПМК записывается: в нулевое слово — РГП0, в первое слово — РКЗ и РА УСК, во второе слово — РФ[0/7] из РГП4[0/7], РСК[0/7] из РГП2[8/15] и РКБ[0/15] из РГП2[16/31], в третье слово — РКД. После того как ВУ снова установит связь с каналом, содержимое ПМК подканала, к которому подключено это ВУ, пересылается в соответствующие регистры РГП.

Узел управления передачей данных (УУ ПД) управляет их передачей между ОП и каналом, с одной стороны, и каналом и ВУ — с другой. При выводе данных он осуществляет прием 32-разрядного слова из ОП через ОК и магистраль М1 в РКД. Затем данные выдаются в ВУ из РКД через магистраль М5. При вводе байты данных принимаются из ВУ через магистраль М4 и помещаются в соответствующие позиции РКД. Сформированные полные слова из РКД через М1 и ОК записываются в ОП. Границы байтов в слове указываются кодом в четырехразрядном регистре маркеров, первые два разряда которого содержат код позиции начального байта, а два последних — код позиции конечного байта в слове ОП. При передаче данных в ВУ из ОП считывается полное слово, содержащее необходимые байты, которые в соответствии с содержимым регистра маркеров передаются в ВУ. При вводе данных в ОП коды позиции крайних байтов поступают в ВУ ОП и далее в ОП для записи байтов, подлежащих вводу, в нужные позиции слова ОП.

Ограничитель времени простоя (ОВП) предназначен для ликвидации такой ситуации при монопольной передаче данных, когда временной интервал между ответом канала ИНФ-К и последующим сигналом идентификации абонента ИНФ-А или УПР-А превышает 30 с. В этом случае канал фиксирует ошибку в управлении интерфейсом.

Для микропрограммной реализации операций ввода-вывода в МК используются следующие подпрограммы:

Чтение ПМК. Подпрограмма управляет считыванием информации из четырех ячеек ПМК, соответствующих тому подканалу, который становится активным, и записывает ее в регистры РГП.

Запись ПМК. Подпрограмма осуществляет запись управляющей информации из соответствующих регистров РГП в ячейки

ПМП того подканала, который перестает быть активным. Обе подпрограммы обеспечивают мультиплексный режим в канале.

Выборка АСК и УСК. Подпрограмма используется при выполнении команды НАЧАТЬ ВВОД-ВЫВОД. Она производит выборку управляющей информации из ОП, анализирует ее на отсутствие программных и аппаратурных ошибок, производит модификацию адресов УСК, записывает управляющую информацию в регистры РГП и устанавливает канал в состояние «Хранит прерывание» при наличии признака программно-управляемого прерывания.

Запись ССК. При выполнении команд НАЧАТЬ ВВОД-ВЫВОД и ОСТАНОВИТЬ ВВОД-ВЫВОД, когда признак результата равен 1, запись ССК заключается в записи в ОП байта состояния устройства в ССК [32/39] и байта состояния канала в ССК [40/47]. При этом в ОП записывается соответственно содержимое РСУ [0/7] и РСК [0/7]. При обработке прерываний от канала и при выполнении команды ПРОВЕРИТЬ ВВОД-ВЫВОД в том случае, когда признак результата равен 1, происходит запись в ОП полного двойного слова ССК. В этом случае в ССК [0/31] записывается содержимое РКЗ и РА УСК, а в ССК [32/63] --- содержимое РСУ, РСК и РКБ. -

Анализ байта состояния устройства (БСУ). Подпрограмма производит анализ БСУ как при начальной выборке ВУ, так и при окончании операции ввода-вывода.

Выборка ВУ. Подпрограмма формирует и контролирует последовательность сигналов идентификации канала и абонента, их взаимную блокировку, четность принимаемой из интерфейса и передаваемой в него информации и производит логическое подключение ВУ. При обнаружении ошибок управление передается подпрограмме записи состояния МК, если выполнялась команда ввода-вывода или обрабатывалось прерывание от канала. Если выполнялась перевыборка ВУ по цепочке команд или выборка ВУ по требованию абонента, то в случае ошибок устанавливается прерывание в канале. Управление передается в ОК подпрограмме ожидания запросов. Подпрограмма отличает выборку ВУ по инициативе канала (начальную выборку) от выборки ВУ по инициативе абонента.

Кольцо передачи данных. Подпрограмма организует передачу данных между ОП и ВУ.

Отсоединение ВУ. Подпрограмма после снятия сигнала идентификации РАБ-А выполняет логическое отсоединение ВУ и сброс всех сигналов идентификации канала. При этом она формирует сигналы идентификации интерфейса и контролирует временные соотношения между ними. При обнаружении ошибки, если выполнялась команда ввода-вывода или обрабатывалось прерывание канала, управление передается подпрограмме записи состояния МК. В остальных случаях управление передается в ОК подпрограмме ожидания запросов.

При выполнении команды НАЧАТЬ ВВОД-ВЫВОД после анализа состояния канала (рис. 144) управление передается из подпрограммы обслуживания запросов ЦП микропрограммам МК. По номеру подканала производится чтение содержимого ПМК и передача управляющей информации в РГП. После этого анализируется состояние подканала. Если подканал работает или хранит прерывание, управление передается в ОК подпрограмме окончания выполнения команды. Если же подканал свободен, то из ОП выбирается АСК по адресу 48_{10} . Код адреса УСК записывается в РА УСК, и происходит переход на подпрограмму выборки из оперативной памяти АСК и УСК. Выбранная командная информация помещается в соответствующие регистры РГП и управление передается подпрограмме начальной выборки ВУ.

При начальной выборке ВУ мультиплексный канал выдает в интерфейс на ШИН-К код адреса ВУ, сопровождая его сигналом АДР-К. Каждое ВУ, подсоединенное к каналу, дешифрирует код адреса, но только одно ВУ опознает его как свой номер. Затем канал выдает сигнал ВБР-К, который последовательно проходит через все устройства, пока не достигнет адресуемого. Если оно неисправно или не подключено, по линии обратной выборки приходит сигнал ВБР-А. В этом случае управление передается в ОК подпрограмме окончания выполнения команды. Если в ответ на выборку из устройства приходит сигнал УПР-А, это означает, что УВУ в данный момент времени находится в состоянии «Работает» и байт состояния устройства, передаваемый из ВУ, будет содержать состояния «Модификатор» и «Занято» (см. табл. 44). Байт состояния запоминается в РСУ. Мультиплексный канал записывает ССК в фиксированную ячейку ОП, и управление передается в ОК подпрограмме окончания выполнения команды.

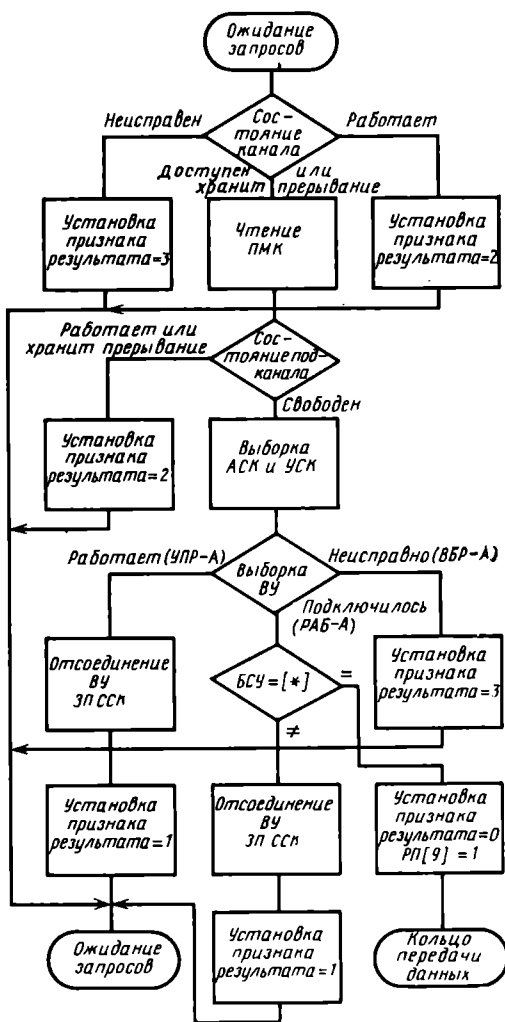


Рис. 144. Схема микропрограммы команды НАЧАТЬ ВВОД-ВЫВОД в МК:
[*] — БСУ = 00 ∨ 08 ∨ 0С ∨ 4С

Если ВУ свободно, то в ответ на выборку из него приходит сигнал РАБ-А. Одновременно ВУ на ШИН-А выдает свой адрес в сопровождении сигнала АДР-А. Канал сравнивает посланный и полученный коды адреса. При их совпадении происходит логическое подключение ВУ к каналу. Затем в ВУ выдается код операции, а оно посылает в ответ байт состояния. На этом начальная выборка ВУ завершается. Канал анализирует байт состояния устройства. Если он равен 00, 08, 0С или 4С (см. табл. 44), то устанавливается признак работы подканала $РП[9] = 1$ и признак результата, равный 0. Последний выдается в ЦП. Процессор, приняв нулевой признак результата, записывает его в РССП[34/35] и на этом завершает команду НАЧАТЬ ВВОД-ВЫВОД, а канал освобождается для работы с другими ВУ. Управление передается подпрограмме кольца передачи данных. Если байт состояния имеет иное значение, то происходит отсоединение ВУ, запись ССК и переход в ОК к подпрограмме окончания выполнения команды.

Выборка ВУ по инициативе абонента (рис. 145) происходит в том случае, если ВУ готово к приему (передаче) информации и требуется начать этот процесс. Это может быть, например, тогда, когда после начальной выборки ВУ произошло включение его механизмов и оно оказалось приведенным в состояние готовности, или ВУ окончило печать очередной «порции» данных и т. п. В этом случае ВУ посылает в канал сигнал ТРБ-А, на что канал отвечает сигналом ВБР-К. Этот сигнал проходит последовательно по всем ВУ до тех пор, пока не достигнет ВУ, пославшего сигнал ТРБ-А. Выбранное ВУ отвечает сигналом РАБ-А и выдает в интерфейс код своего адреса в сопровождении сигнала АДР-А, после чего происходит логическое подключение ВУ к каналу. Канал посылает в ВУ код операции, равный нулю, означающий продолжение работы, связанной с текущей операцией ввода-вывода, и сигнал УПР-К. Затем микропрограмма канала анализирует состояние регистра прерываний РП[9]. Его единичное состояние означает, что подканал находится в состоянии «Работает». При этом управление передается в подпрограмму «Кольцо передачи данных». Если же канал находится в каком-либо другом состоянии, то в устройство передается приказ ЗАПОМНИТЬ СОСТОЯНИЕ. В этом случае управление передается в ОК подпрограмме ожидания запросов.

При выполнении подпрограммы «Кольцо передачи данных» канал ожидает прихода сигналов ИНФ-А или УПР-А от ВУ. Если приходит сигнал УПР-А, то это означает, что по линиям ШИН-А ВУ выдает байт своего состояния и происходит переход к подпрограмме анализа этого байта. Если от ВУ приходит сигнал ИНФ-А, то это означает, что канал выдает (принимает) в интерфейс (из интерфейса) байт данных. Передача (прием) данных происходит из РКД. После передачи (приема) очередного байта из РКД в интерфейс происходит модификация и проверка со-

держимого регистра количества байт. Если РКБ = 0 и отсутствует признак цепочки данных, то канал прекращает передачу. Подпрограмма уходит на ожидание запросов. Если же имеется признак ЦД, то управление передается подпрограмме выборки нового УСК.

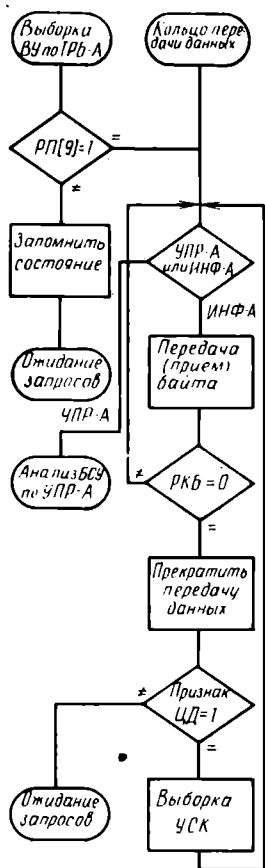


Рис. 145. Схема микропрограммы кольца передачи данных и цепочки данных в МК

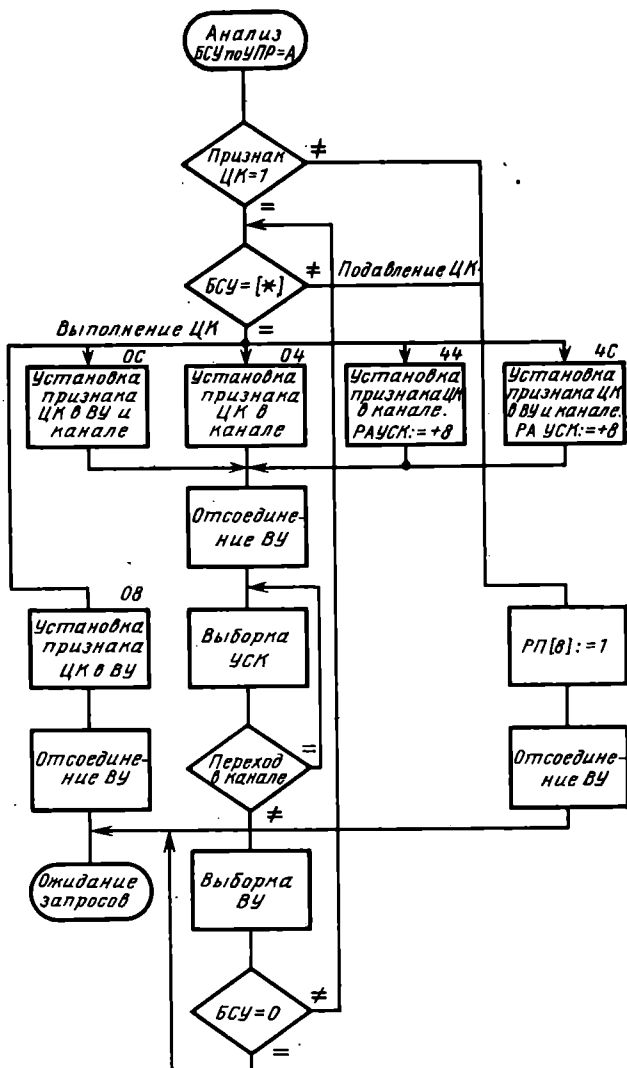


Рис. 146. Схема микропрограммы анализа байта состояния устройства и выполнения цепочки команд в МК: [*] — БСУ = 08 ∨ 0C ∨ 04 ∨ 44 ∨ 4C

Анализ байта состояния устройства начинается с анализа признака цепочки команд в текущем УСК (рис. 146). Так как переход к анализу БСУ происходит всегда при ненулевом значении признака, то при его отсутствии устанавливается признак прерывания в подканале (РП[8]: = 1). Если имеется признак ЦК, то подпрограмма анализирует значение байта состояния устройства.

Выполнение цепочки команд возможно при следующих значениях признаков в БСУ (см. табл. 44):

КАНАЛ КОНЧИЛ (БСУ = 08) — операция в подканале прекращается и устанавливается признак ЦК в ВУ. Происходит отсоединение ВУ и передача управления в ОК подпрограмме ожидания запросов;

КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО (БСУ = 0С) — операция в подканале прекращается и устанавливается признак ЦК в ВУ и канале;

ВУ КОНЧИЛО (БСУ = 04) — устанавливается признак ЦК в канале;

МОДИФИКАТОР, ВУ КОНЧИЛО (БСУ = 44) — адрес УСК модифицируется на +8, в канале устанавливается признак ЦК и прекращается операция подканала;

МОДИФИКАТОР, КАНАЛ КОНЧИЛ, ВУ КОНЧИЛО (БСУ = 4С) — адрес УСК модифицируется на +8, устанавливается признак ЦК в ВУ и канале, прекращается операция в подканале.

В последних четырех случаях происходит отсоединение ВУ и выборка нового УСК. Если в нем задан код операции ПЕРЕХОД В КАНАЛЕ,

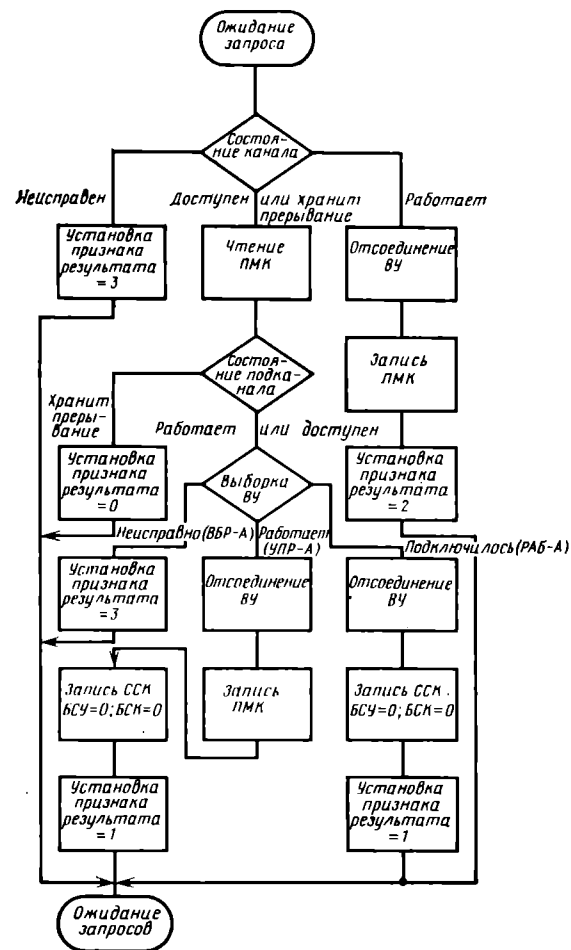


Рис. 147. Схема микропрограммы команды ОСТАНОВИТЬ ВВОД-ВЫВОД В МК

то происходит выборка нового УСК. В противном случае происходит перевыборка ВУ по цепочке команд.

Если БСУ не совпадает с рассмотренными выше значениями, то операция в подканале прекращается с одновременным подавлением цепочки команд.

Для принудительного прекращения операции ввода-вывода используется команда ОСТАНОВИТЬ ВВОД-ВЫВОД. Алгоритм ее выполнения представлен на рис. 147.

Если канал работает, происходят отсоединение ВУ и запись управляющей информации активного подканала в ПМК подпрограммой записи ПМК. После этого управление передается в ОК. Если канал доступен или хранит прерывание, производится акти-

визация подканала путем чтения из ПМК и передач в РГП его управляющей информации. После этого проверяется состояние подканала, если подканал доступен или работает, происходит выборка ВУ. Если ВУ свободно и подключилось, происходит его отсоединение, а в ССК записываются нулевые значения байт

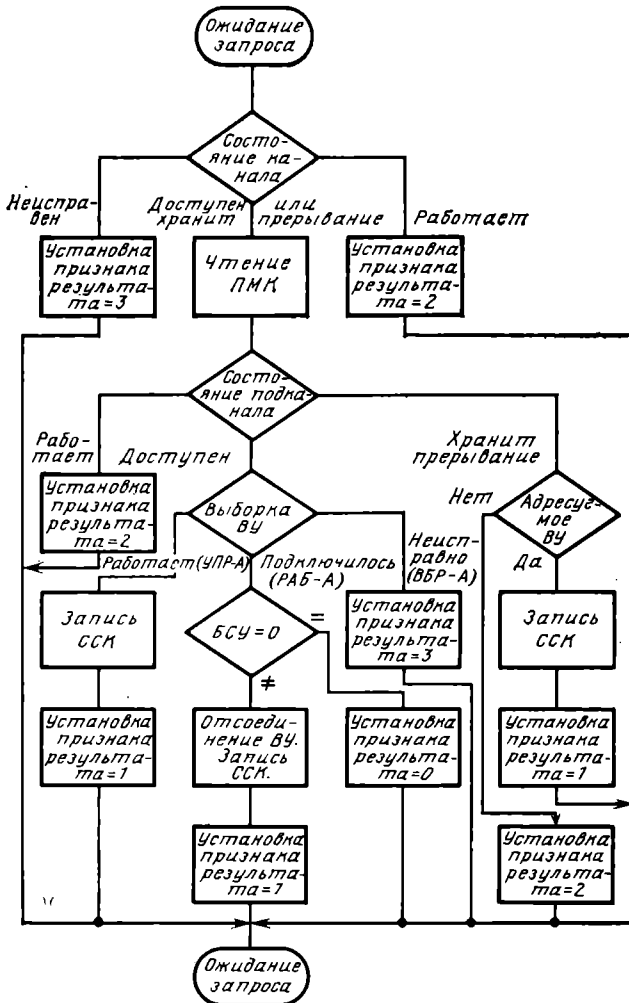


Рис. 148. Схема микропрограммы команды ПРО-
ВЕРИТЬ ВВОД-ВЫВОД в МК

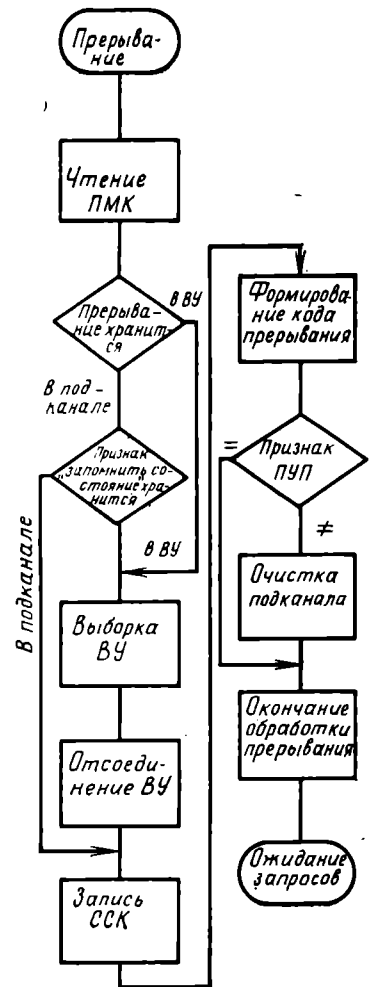


Рис. 149. Схема микропрограммы обработки прерываний канала и подканала в МК

состояний устройства и канала. Если ВУ было занято операцией ввода-вывода, то в ответ на сигнал выборки ВБР-К оно выдает байт своего состояния с признаками МОДИФИКАТОР и ЗАНЯТО, сопровождая этот байт сигналом идентификации УПР-А. Канал производит отсоединение ВУ и запись содержимого РГП МК в область ПМК, соответствующую данному подканалу. В ССК [32/47] записывается нулевое значение байта состояния канала и значение байта состояния устройства. Выполнение операции ввода-вывода прекращается.

В командах ПРОВЕРИТЬ ВВОД-ВЫВОД и ПРОВЕРИТЬ КАНАЛ также используются описанные выше подпрограммы (рис. 148). Если подканал находится в состоянии «Хранит прерывание», то значение признака результата в команде ПРОВЕРИТЬ ВВОД-ВЫВОД зависит от того, хранит ли подканал адрес устройства, адресуемого командой, или какой-либо другой.

По команде ПРОВЕРИТЬ КАНАЛ проверяется только состояние адресуемого канала и устанавливается соответствующий признак результата. При выполнении команды ВУ не выбирается и подканалы не опрашиваются.

Обработка прерывания от ввода-вывода (рис. 149) начинается чтением содержимого ПМК, после чего определяется, где хранится прерывание — в ВУ или в подканале. Если прерывание хранится в ВУ, то происходит его выборка и логическое отсоединение. После записи ССК и формирования кода прерывания проверяется вид прерывания. Если обрабатывается программно управляемое прерывание, то очистки подканала не происходит. Если прерывание хранится в подканале, то проверяется, где хранится признак ЗАПОМНИТЬ СОСТОЯНИЕ. Если он хранится в ВУ, то происходит выборка этого устройства. Если же в подканале, то — переход на запись ССК. Прерывания в МК завершаются переходом в ОК на подпрограмму окончания обработки прерывания.

12.4. СЕЛЕКТОРНЫЙ КАНАЛ

Модель ЕС-1033 имеет три селекторных канала СК1, СК2 и СК3. Все они идентичны, однако их приоритет при обработке прерываний и при обращении к ОП различен. Селекторные каналы служат для подключения и обслуживания внешних запоминающих устройств и работают всегда в монопольном режиме, имея в своем составе по одному подканалу. Управление в СК — аппаратно-микропрограммное, т. е. часть действий реализована микропрограммным путем, а часть — аппаратурным. Микропрограммная информация хранится в памяти микропрограмм (ПМП), входящей в состав ОК. Аппаратурное управление реализовано с помощью триггерных тактовых цепочек, которые вырабатывают необходимую последовательность управляющих сигналов. Запуск тактовых цепочек происходит микропрограммно

При построении СК, как и МК, применена магистральная организация связей между узлами и блоками канала. Вся передача информации происходит через информационные магистрали М1, М2, М3, М4 и М5 (рис. 150). Магистраль М1 является внешней магистралью. Она служит для связи СК с коммутатором числовой информации ОК, и через нее осуществляется обмен данными между ОП и СК. Данные передаются 32-разрядными словами. Остальные магистрали являются внутренними магистралями СК.

Все магистрали, кроме М3, имеют элементы памяти — магистральные регистры (РМ), которые осуществляют прием информации, хранение ее в течение необходимого времени и сброс магистралей в исходное состояние. Сброшенная магистраль имеет единицы во всех разрядах, включая контрольные. Контрольным разрядом снабжаются каждые восемь информационных разрядов магистрали.

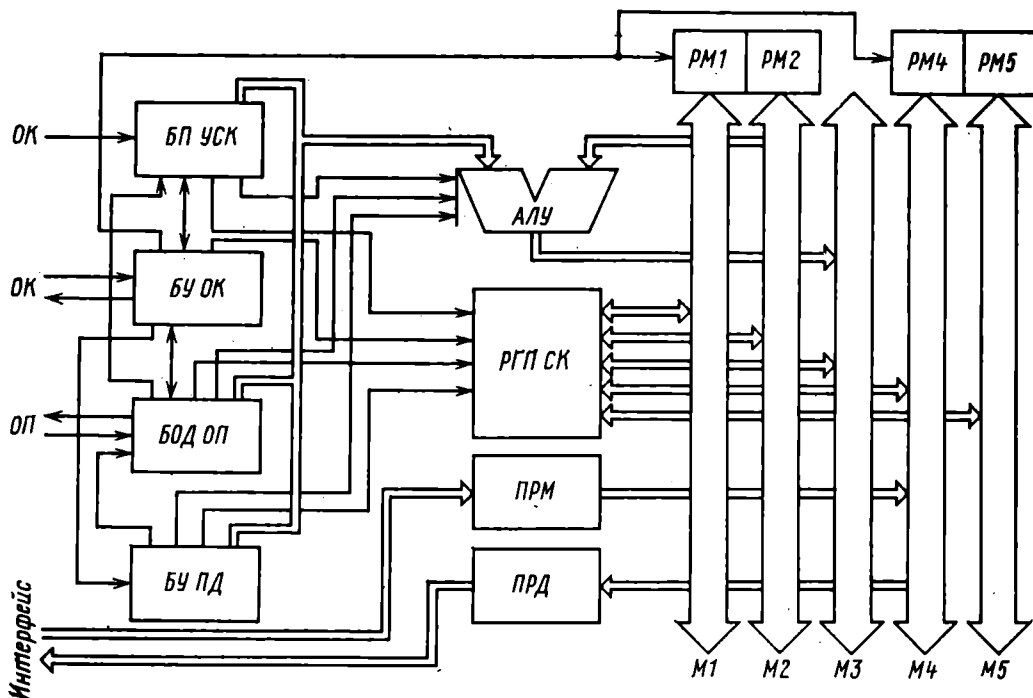


Рис. 150. Структурная схема селекторного канала

Центральным блоком СК является регистровая память (РГП СК), функционирующая совместно с информационными магистралями и РМ. В ней хранится вся информация, необходимая для работы СК. Память построена на ИС К155ХЛ1, что позволило выполнить ее по магистральному принципу. Управление записью и чтением содержимого регистров памяти происходит под воздействием управляющих сигналов, которые могут быть сформированы как микропрограммно, так и аппаратурно. Блок приема УСК осуществляет обращение в ОП за этим словом, модификацию его адресов и предварительную выборку. Блок обмена данными (БОО ОП) служит для управления процессом обмена данными между каналом и ОП. Блок управления передачей данных (БУ ПД) предназначен для управления передачей данных между каналом и ВУ. Эти три блока обеспечивают аппаратурное управление вводом и выводом данных и выборкой из ОП командной информации канала путем формирования необходимых последовательностей управляющих сигналов. Инициирование этих блоков

осуществляется микропрограммно, после чего дальнейшая их работа происходит автономно от микропрограммы. Блок управления от общего канала (БУ ОК) производит активизацию микро-

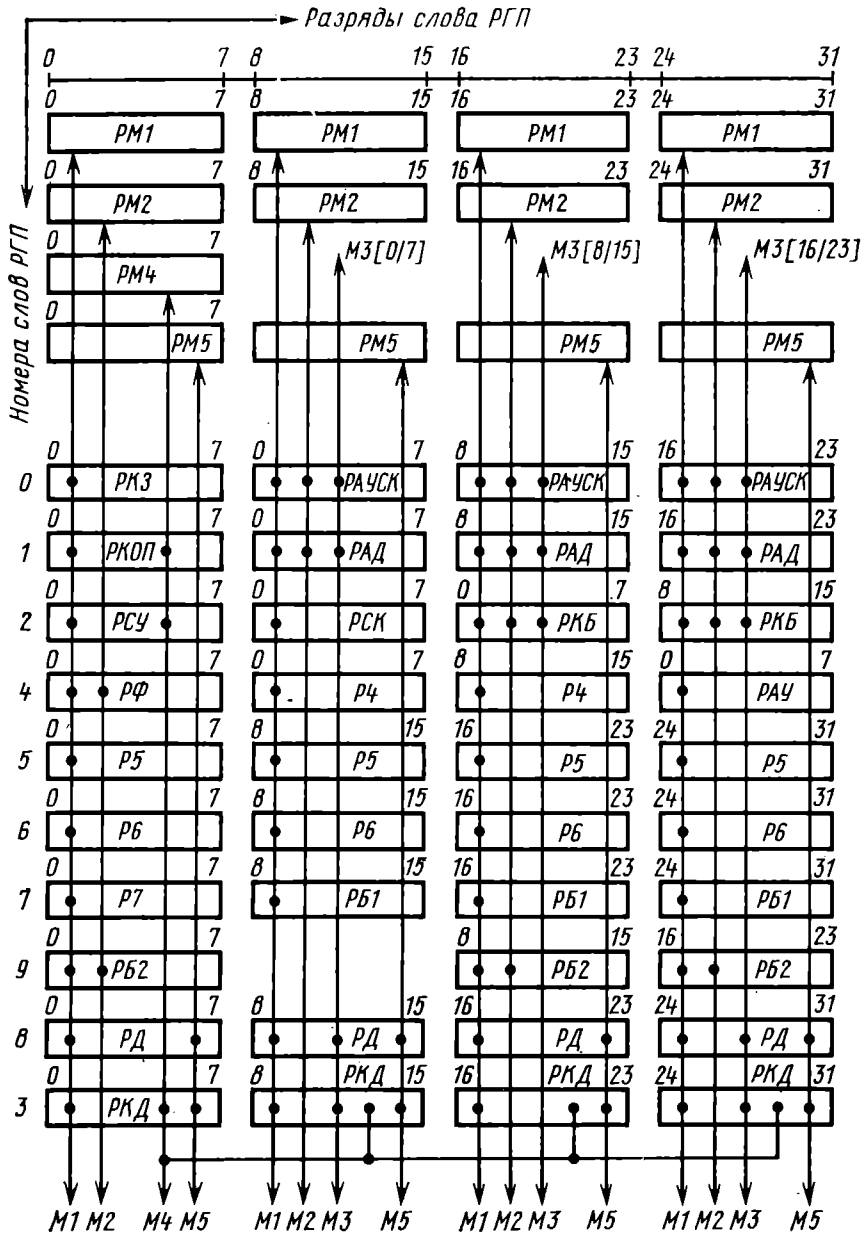


Рис. 151. Структура регистровой памяти СК

приказов, поступающих из ОК в СК для реализации микропрограмм последнего.

Структура регистровой памяти и связи регистров с магистралями представлена на рис. 151. Точками на пересечении изображений регистра и магистрали обозначено наличие связи между ними. Содержимое РГП СК через магистраль М1 может быть про-

индицировано на пульте управления (п. 13.1). На линейке индикации одновременно может быть проиндицировано только одно слово РГП.

Большинство регистров РГП СК по своим функциям аналогичны соответствующим регистрам РГП МК. Это регистры РКЗ, РА УСК, РКОП, РАД, РСУ, РСК, РКБ, РФ. Назначение и содержимое остальных регистров следующее:

Р4[0/15] — условный регистр (РГП4[8/23]). Содержит триггеры, в которых фиксируются сигналы ошибок байт числа на магистрали: ТОМ1[0] — ТОМ1[8], ТОМ5[0] — ТОМ5[3] и ТОМ4; триггеры останова ТОСТ* и ТОСТ, регистр положения байта РПБ[0/1] в слове ОП.

Р5[0/31] — условный регистр (РГП5[0/31]). Содержит триггеры тактовой цепочки, которая осуществляет выработку последовательности управляющих сигналов при аппаратурном управлении, триггеры ошибки операнда на входе АЛУ (ТОВ АЛУ), триггеры ошибки результата АЛУ (ТО АЛУ) и ряд служебных триггеров;

Р6[0/31] — условный регистр (РГП6[0/31]). Содержит ряд служебных триггеров, а также триггеры, хранящие сигналы идентификации интерфейса.

Р7[0/7] — условный регистр (РГП7[0/7]). Также содержит ряд служебных триггеров.

РБ1[0/23] — регистр буферный 1 (РГП7[8/31]). В этот регистр записывается код адреса УСК, увеличенный на 8. Это обеспечивает сохранение продвинутого адреса УСК в том случае, когда при выполнении операции ПЕРЕХОД В КАНАЛЕ происходит ошибка при выборе нового УСК.

РБ2[0/31] — регистр буферный 2 (РГП9[0/31]). При предварительной выборке УСК в него помещается второе слово УСК.

РД[0/31] — регистр данных (РГП8[0/31]). Предназначен для хранения данных при обмене между каналом и ОП в операциях ввода-вывода.

РКД[0/31] — регистр коммутации данных (РГП3[0/31]). Этот регистр осуществляет упаковку байтов данных в слово и распаковку слова на байты при обмене данными между каналом и ВУ.

Для микропрограммной реализации операций ввода-вывода в СК используются следующие подпрограммы:

Анализ состояния подканала. Эта подпрограмма получает управление от микропрограмм ОК при выполнении команд ввода-вывода, при обработке прерываний от ввода-вывода и по запросу канала на прием байта состояния ВУ по сигналу идентификации УПР-А.

Выборка АСК и УСК. Подпрограмма выбирает АСК из ОП по фиксированному адресу, записывает код ключа защиты в РКЗ, код адреса УСК в РА УСК, после чего запускает тактовую цепочку БП УСК. Последняя формирует последовательность управляющих сигналов, необходимых для выборки УСК из ОП. Таким

образом, выборка из ОП АСК осуществляется микропрограммно, а выборка УСК и размещение его в РГП — аппаратурно. После выборки УСК подпрограмма анализирует принятую управляющую информацию на отсутствие аппаратурных и программных ошибок и передает управление подпрограмме выборки ВУ. Если обнаруживаются аппаратурные ошибки, то управление передается подпрограмме записи состояния СК, если программные — подпрограмме записи ССК. Если ошибки обнаруживаются при перевыборке УСК по цепочке команд, то в канале устанавливается прерывание и управление передается в ОК подпрограмме ожидания запросов.

Обращение к ОП за АСК по фиксированному адресу, его выборка и запись адресной информации в РГП происходят микропрограммно. При аппаратурной выборке УСК после запуска тактовой цепочки БП УСК формирует: запрос к ОП за УСК, управляющие сигналы модификации содержимого РА УСК и константу модификации, равную 4. Содержимое РА УСК считывается на магистрали М1 и М2. С магистрали М1 код адреса поступает через ОК в ОП, а с магистрали М2 — на один из входов АЛУ. На другой вход АЛУ поступает константа модификации, а на управляющие входы — код, обеспечивающий функцию АЛУ $F := A + B$, после чего увеличенный на 4 адрес УСК через М3 снова записывается в РА УСК. Затем формируется запрос за вторым словом УСК, а принятое из ОП первое слово проверяется на отсутствие ошибок и помещается в РК ОП и РАД. При наличии программных ошибок производится дополнительная модификация кода адреса УСК на 4 и увеличенный на 8 код адреса заносится в РБ1 для последующего сообщения его системе. Принятое из ОП второе слово УСК помещается в буферный регистр РБ2, откуда потом передается в РФ и РКБ.

При выводе данных из ОП в ВУ в том случае, когда в текущем УСК имеется признак цепочки данных, темп поступления информации в интерфейс не должен замедляться в моменты выборки управляющих слов. Для этой цели осуществляется предварительная выборка УСК. При выводе данных из канала в регистрах РКД и РД хранятся очередные два слова данных, считанные из ОП. Если по текущему УСК осталось передать в ВУ восемь или меньше байт, то к началу передачи они оказываются уже выбранными из ОП. В этом случае при наличии признака цепочки данных аппаратно инициируется предварительная выборка следующего УСК. Выбранное двойное слово размещается в РБ1 и РБ2. После того как начнется передача байтов из последнего слова данных, находящегося в РКД, первое слово УСК из РБ1 помещается в РК ОП и РАД. Далее происходит обращение к ОП за первым словом данных, его прием и запись в РД. После того как передача байтов из РКД в ВУ по текущему УСК закончится второе, уже выбранное из ОП слово УСК из РБ2 перемещается в РФ и РКБ.

При вводе данных из ВУ в ОП, когда имеется признак цепочки данных, прием их из интерфейса и обращения к ОП со стороны каналов на время перевыборки следующего УСК блокируется.

Подпрограмма начальной загрузки программы. Ее функции рассмотрим несколько позднее в ходе описания работы канала.

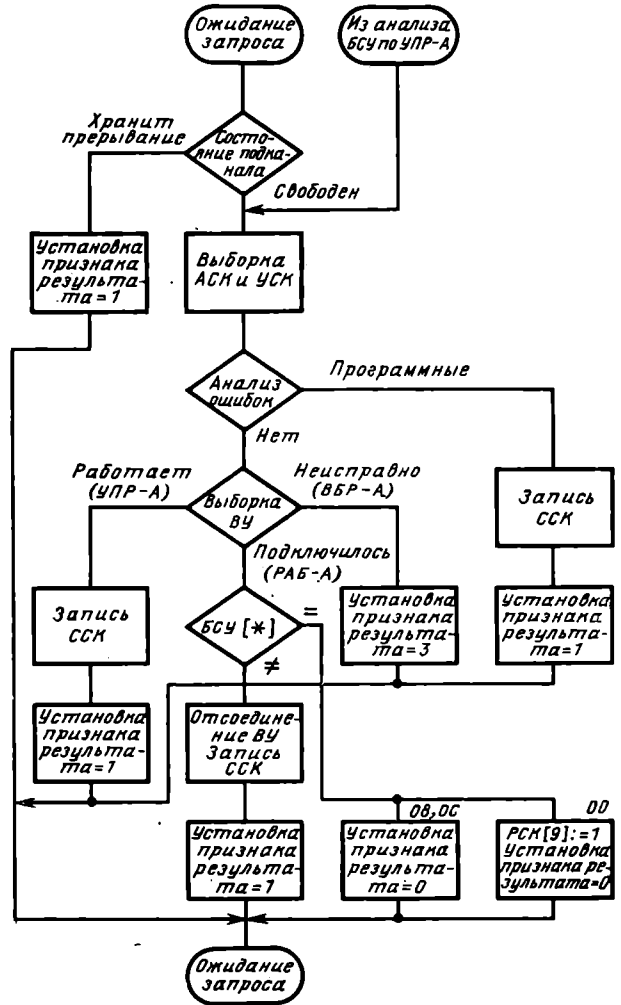


Рис. 152. Схема алгоритма выполнения команды НАЧАТЬ ВВОД-ВЫВОД в СК: [*] — БСУ = 00 ∨ 08 ∨ 0C

Подпрограммы анализа байта состояния устройства, выборки ВУ, отсоединения ВУ и записи ССК выполняют те же функции, что и аналогичные подпрограммы в МК.

Работа селекторного канала начинается по команде НАЧАТЬ ВВОД-ВЫВОД, которая, в свою очередь, начинается подпрограммой анализа состояния подканала (рис. 152). Если подканал хранит прерывание, управление передается в ОК подпрограмме окончания команды. Если подканал свободен, то подпрограмма снимает прерывание в канале, если оно имеется, переписывает код адреса ВУ из РБСП в РАУ канала и передает управление подпрограмме выборки АСК и УСК. Последние выбираются из ОП и, если не обнаружено при этом никаких ошибок, то осуществ-

вляется начальная выборка ВУ. Занятое ВУ отвечает сигналом идентификации УПР-А и выдает свой байт состояния. Происходит отсоединение ВУ, запись ССК и управление передается в ОК подпрограмме окончания команды. Если ВУ не занято и подключилось, оно отвечает сигналом РАБ-А. В этом случае анализируется байт состояния устройства. Если $БСУ \neq 00,08$ или $0С$ (см. табл. 44), то происходит отсоединение опрашиваемого устройства, запись ССК и передача управления в ОК подпрограммам окончания команды. Если $БСУ = 08$ или $0С$, то ВУ подготавливается к работе и управление передается в ОК. Если $БСУ = 00$, то это означает, что устройство готово к выполнению операции ввода-вывода. При этом, если выполняемая команда требует передачи данных из ОП, то подпрограмма выборки ВУ формирует и выдает в канал управляющий сигнал, по которому запускается тактовая цепочка блока обмена данными с ОП. Последняя начинает схемное формирование последовательности управляющих сигналов. Сначала из ОП считывается первое слово данных по адресу, код которого находится в РАД. Оно поступает в РКД. В зависимости от количества передаваемых байтов и расположения первого из них в слове БОД ОП выбирает второе слово данных (если это необходимо) и помещает его в РД. При этом БОД ОП выполняет все действия, связанные с модификацией кода адреса данных аналогично тому, как это делается при модификации кода адреса УСК. После заполнения РКД и РД канал устанавливается в режим передачи данных установкой $РСК[9] := 1$. Если же канал должен вести прием данных из ВУ в ОП, то работа БОД ОП не инициируется, а канал сразу же устанавливается в режим приема данных.

Выполнение команды **ОСТАНОВИТЬ ВВОД-ВЫВОД** начинается в СК после передачи управления из ОК, если канал работает в монопольном режиме, хранит прерывание или свободен (рис. 153). Если при этом подпрограмма анализа состояния подканала определит, что последний свободен, то происходит выборка ВУ и его отсоединение с формированием соответствующего признака результата. При записи ССК в ОП записываются только байты состояния устройства и канала.

Команда **ПРОВЕРИТЬ ВВОД-ВЫВОД** (рис. 154) начинается выполняться с подпрограммы анализа состояния подканала. В случаях, когда канал и подканал свободны, канал хранит прерывание, но подканал свободен или канал и подканал хранят прерывание — подпрограмма анализа состояния подканала передает управление подпрограмме выборки ВУ. Если подканал свободен, то происходит подключение ВУ и передача в него кода 00 операции, **ПРОДОЛЖИТЬ**. После этого анализируется байт состояния устройства. Если $БСУ = 00$, то происходит отсоединение ВУ и передача управления в ОК подпрограмме окончания команды. При занятом ВУ и при $БСУ \neq 00$ перед передачей управления в ОК подпрограмме окончания выполнения команды

происходит запись ССК. В этом случае в ОП записывается двойное слово ССК, сформированное в ходе выполнения команды, по которой выполняется операция ввода-вывода.

Выполнение команды ПРОВЕРИТЬ КАНАЛ в селекторном канале происходит так же, как и в мультиплексном канале.

Обработка прерывания начинается с определения его вида подпрограммой анализа состояния подканала: прерывание канала, подканала или программно-управляемое прерывание

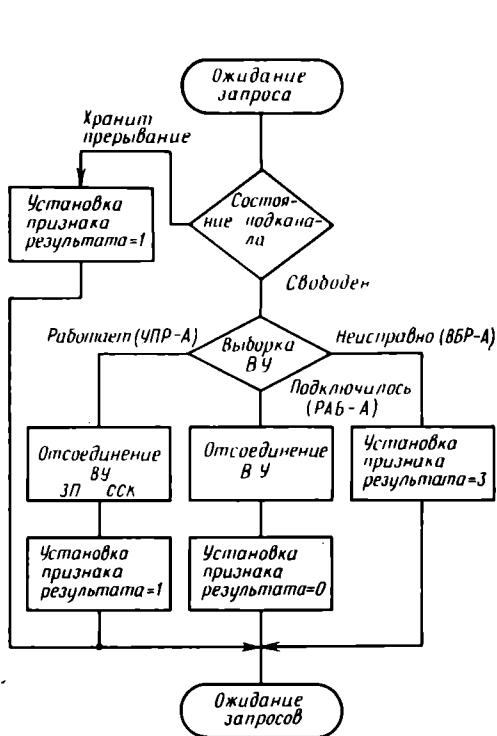


Рис. 153. Схема алгоритма выполнения команды ОСТАНОВИТЬ ВВОД-ВЫВОД в СК

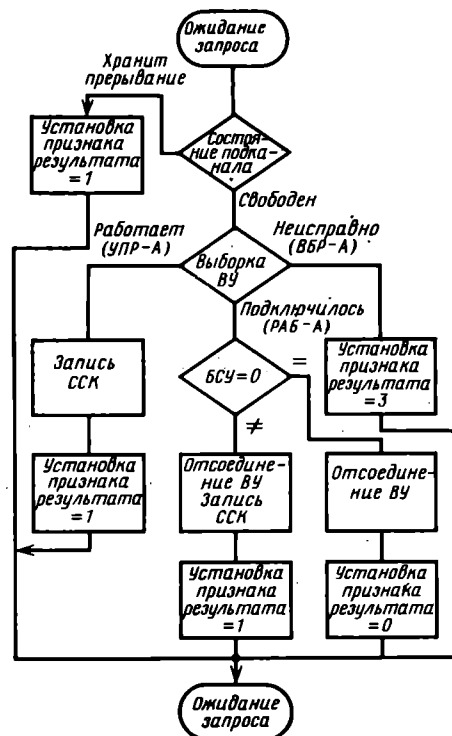


Рис. 154. Схема микропрограммы команды ПРОВЕРИТЬ ВВОД-ВЫВОД в СК

(рис. 155). При обработке прерывания канала управление передается подпрограмме выборки ВУ. После подключения ВУ в него передается код 00 операции ПРОДОЛЖИТЬ. Когда будет получен байт состояния устройства, произойдет запись в ОП двойного слова ССК и управление будет передано в ОК подпрограмме окончания обработки прерываний. При прерываниях подканала управление сразу передается подпрограмме записи ССК.

При обработке программно-управляемого прерывания (признак цепочки команд задан) анализируется байт состояния устройства. Если БСУ \neq 08, 04, 0С или 4С, то происходит установка прерываний в канале (РСК [8]: = 1), отсоединение ВУ и передача управления в ОК. В противном случае анализируется состояние РСК [0]. При РСК [0] = 0 происходит перевыборка УСК по цепочке команд (рис. 156). При РСК [0] = 1 устанавливается при-

знак прерывания в канале (РСК [8]: = 1), в РСУ заносится нулевой код и канал ожидает сигнал разрешения из процессора для обработки прерывания. Если в течение 10 мкс разрешения нет, то запрос прерывания в канале гасится и происходит переход к пере-выборке УСК по цепочке команд. Если канал дает разрешение раньше, чем через 10 мкс, то происходит запись полного ССК и переход на перевыборку УСК по цепочке команд.

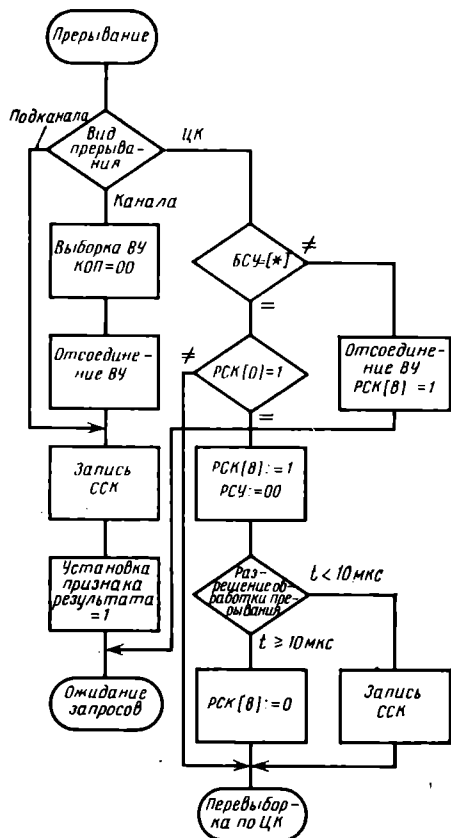
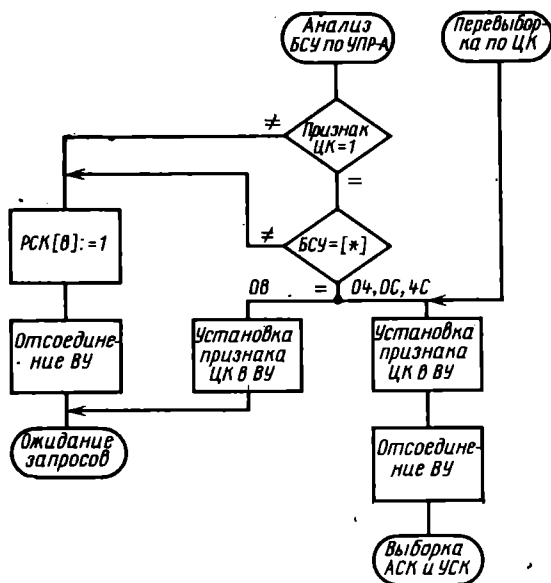


Рис. 155. Обработка прерываний в СК: [*] — БСУ = 08 ∨ 04 ∨ 0С ∨ 4С

Рис. 156. Схема алгоритма анализа БСУ и выполнения цепочки команд в СК: [*] — БСУ = 08 ∨ 04 ∨ 0С ∨ 4С



Когда канал запрашивает для анализа байт состояния устройства, то по сигналу идентификации УПР-А проверяется наличие признака цепочки команд. Если он установлен, но БСУ ≠ 08, 04, 0С или 4С, то цепочка команд подавляется. При этом, а также когда признак цепочки команд отсутствует, происходит отсоединение ВУ, установка признака прерывания в канале (РСК [8]: = := 1) и управление передается в ОК подпрограмме ожидания запросов.

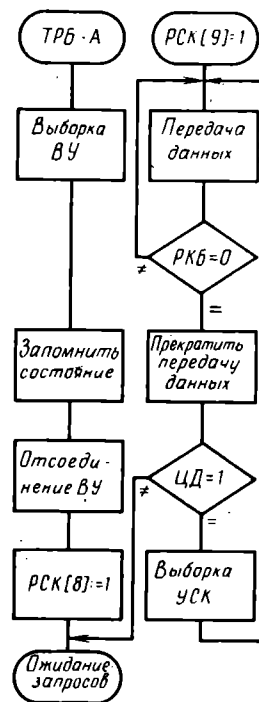
Если БСУ = 08 (КАНАЛ КОНЧИЛ), то происходит установка признака цепочки команд в ВУ и переход на подпрограмму ожидания запросов. При БСУ = 04, 0С или 4С в ВУ устанавливается признак цепочки команд, оно отсоединяется и управление передается подпрограмме выборки АСК и УСК. Начинается пере-выборка УСК по цепочке команд.

При обработке запросов канала по сигналу идентификации ТРБ-А (выборка ВУ по требованию абонента) после подключения ВУ и получения от него байта состояния в интерфейс выдается приказ ЗАПОМНИТЬ СОСТОЯНИЕ и в канале устанавливается признак прерывания ($РСК[8] = 1$, рис. 157). В зависимости от кода операции в команде канала передача данных каналом (признак $РСК[9] = 1$) заключается в их вводе или выводе в ОП или из ОП соответственно.

Рис. 157. Схема алгоритма передачи данных в СК

При выводе блок обмена данными с ОП производит считывание кода ключа защиты и кода адреса данных из РКЗ и РАД на магистраль М1, организует модификацию кода адреса данных и запись модифицированного кода в РАД. Коды ключа защиты и адреса данных с магистрали М1 через коммутатор ОК и БУ ОП пересылаются в ОП вместе с запросом СК за данными. Считанное из ОП слово данных помещается в РКД. Если при анализе количества выводимых данных обнаружена необходимость обращения в ОП за вторым словом данных, то снова запускается тактовая цепочка БОД ОП и аналогичным образом выбирается второе слово данных. В этом случае оно помещается в РД.

Затем запускается тактовая цепочка БУ ПД. При этом один байт данных, код позиции которого определяется содержимым регистра положения байтов РПБ, считывается из РКД и через магистраль М4 выдается в интерфейс. Начальная запись в РПБ управляется значениями двух младших разрядов первоначального кода адреса данных. При передаче байта данных в интерфейс формируется сигнал идентификации ИНФ-К, по которому ВУ принимает байт и сбрасывает сигнал ИНФ-К, после чего БУ ПД модифицирует содержимое РКБ на -1 , а содержимое РПБ — на $+1$ (или -1 , в зависимости от направления записи — чтения). В дальнейшем процесс передачи байтов в ВУ из РКД повторяется до тех пор, пока не будет передан последний байт. После завершения этого процесса второе слово данных через магистраль М5 переписывается из РД в РКД, и процесс передачи байтов в ВУ возобновляется. Если необходимо, то из ОП считывается очередное слово и помещается в РД. Вывод данных из ОП в канал и из канала в ВУ повторяется до нормального окончания операции по значению $РКБ = 0$, либо до поступления команды ОСТАНОВИТЬ ВВОД-ВЫВОД, либо до завершения операции по инициативе абонента.



При выводе по цепочке данных производится анализ количества байтов, подлежащих передаче в интерфейс. Когда оно равно или меньше восьми, прекращается работа БОД ОП и инициируется работа БП УСК для предварительной выборки УСК.

При вводе БУ ПД осуществляет прием байта данных из интерфейса и производит его запись в РКД. Позиция этого байта в слове данных определяется содержимым РПБ. Затем происходит модификация кода в РПБ и РКБ, и процесс повторяется до заполнения РКД. После заполнения РКД инициируется работа БОД ОП. Этот блок переписывает слово данных из РКД в РД. Затем он формирует запрос к ОП на запись слова данных, передает в ОП коды ключа защиты и адреса данных и производит модификацию содержимого РАД.

При вводе по цепочке данных после окончания операции по текущему УСК при РКБ = 0 блокируется прием данных из интерфейса на время, необходимое для перевыборки УСК.

Канал формирует запрос на обработку программных ошибок ввода-вывода при РСК [1/4,7] \neq 0. При обработке этого запроса происходит отсоединение ВУ, фиксируется прерывание в канале (РСК [8]: = 1) и управление передается подпрограмме ожидания запросов.

Подпрограмма начальной загрузки выполняется по команде с пульта управления машиной при нажатии кнопки «НАЧАЛЬНАЯ ЗАГРУЗКА». При этом процессор сначала выполняет микропрограмму сброса системы (п. 4.2), после чего устанавливает триггер начальной загрузки (ТНЗ: = 1) и микропрограмма зацикливается на его анализе. Единичное состояние ТНЗ инициирует выполнение подпрограммы начальной загрузки в канале. При этом в РНК заносится код адреса канала с клавиатуры пульта управления, а в РГП канала поступает следующая информация (в шестнадцатиричных кодах):

РАУ — информация с клавиатуры пульта управления
(адрес ВУ);
РКЗ — 00;
РА УСК — 00 0008;
РФ — 60;
РАД — 00 0000;
РКОП — 02;
РКБ — 0018.

Код адреса ВУ посылается в интерфейс в сопровождении сигнала идентификации АДР-К, и происходит начальная выборка заданного ВУ, которое отвечает каналу байтом состояния. Если БСУ = 00, то подпрограмма выходит на передачу данных так же, как и при выполнении цепочки команд. После успешного завершения загрузки начальной программы происходит отсоединение ВУ с выдачей байта состояния, запись в РБСП кодов адресов канала и ВУ, установка канала и подканала в состояние «Доступен»,

сброс ТНЗ и переход к подпрограмме ожидания запросов ОК. По нулевому значению ТНЗ процессор завершает свою микропрограмму загрузки начальной программы.

ГЛАВА 13

СРЕДСТВА КОНТРОЛЯ И ДИАГНОСТИКИ КАНАЛОВ ВВОДА-ВЫВОДА

13.1. ПУЛЬТ УПРАВЛЕНИЯ

Пульт управления каналов ввода-вывода является частью пульта управления машины (см. рис. 63) и состоит из панелей индикации и управления. Панель индикации расположена в нижней половине верхней панели пульта, а панель управления — в нижней левой части нижней панели пульта управления машины.

Панель индикации содержит три линейки индикации. В первых 16 индикационных лампах, объединенных надписью «ЗАПРОС БОП», индицируются виды запросов каждого из каналов, процессора и пульта управления. Эти запросы вызывают из памяти микропрограмм соответствующую подпрограмму обработки запроса.

Лампы «ТРБ-А» предназначены для индикации требования абонента на связь с каналом для передачи данных или информации о состоянии абонента по сигналу идентификации ТРБ-А.

Лампы «РСК [8]» сигнализируют о том, что запрос каналов на микропрограмму обслуживания прерывания по значению $РСК [8] = 1$ принят общим каналом. Этот запрос поступает в процессор в качестве сигнала управления программы от ввода-вывода.

Лампы «УПР-А» селекторных каналов предназначены для индикации запроса на микропрограмму обработки байта состояния устройства по сигналу идентификации УПР-А.

Лампы «РСК» селекторных каналов предназначены для индикации запросов на микропрограммы обслуживания прерываний, возникающих в каналах, по состоянию $РСК [1/4, 7] \neq 0$.

Лампа «ТВ/В» состояния триггера ввода-вывода предназначена для индикации запроса процессора на микропрограммы обслуживания команд ввода-вывода (SIO, NIO, TIO, TCH).

Лампа «ТНЗ» состояния триггера начальной загрузки индицирует выполнение операции начальной загрузки программы.

Лампа «ТФА» состояния триггера фиксированного адреса предназначена для индикации обращения общего канала к ОП по фиксированному адресу, например, за АСК, ССК и т. д.

Лампы «№ КАНАЛА» предназначены для индикации номера канала (разряды 0 и 1), который связан в данный момент с общим каналом. Здесь же индицируется значение контрольного разряда кода номера канала.

Лампы «ОШ БОП4» и «ОШ БОП5» предназначены для индикации ошибок БОП каналов.

Лампа «РСК [11]» состояния РСК [11] предназначена для индикации нерабочего состояния МК.

Лампа «БЛ АК» блокировки адреса канала предназначена для индикации прерывания микропрограммы обслуживания запросов СК запросами МК на передачу данных в монополярном режиме.

Лампа «ТРК» триггера работы каналов предназначена для индикации связи ОК с одним из каналов. Лампа погашена, когда ОК находится в режиме ожидания запросов.

Группа ламп, объединенных общим названием «РЕЖИМ ПРОВЕРКИ», предназначена для индикации автономных и диагностических режимов работы ОК:

«СКАН» — режима сканирования БОП каналов;

«БЛ РАСТ» — блокировки РАСТ. Она указывает на выполнение пультовых операций (в РАСТ сохраняется значение адреса микрокоманды, предшествующей пультовой операции);

«ТД» — состояние триггера команды ДИАГНОСТИКА и указывает на выполнение этой команды в каналах;

лампа останова «ОСТ» предназначена для индикации состояния «Стоп», при котором блокируется работа цепей синхронизации ОК и МК.

Вторая и третья линейки предназначены для индикации состояния аппаратуры каналов. Конкретные узлы, состояния которых индицируются, определяются кодами, которые набраны на клавиатуре «№ РЕГИСТРА» панели управления каналов, и от того, нажата или нет одна из кнопок «ЧТЕНИЕ РГП» или «ЧТЕНИЕ ПМК».

На второй линейке могут быть проиндицированы состояния: триггера ошибки БОП, в котором фиксируется ошибка любого из блоков БОП4 или БОП5, входящих в состав ПМП, триггера тактового режима СК и одного из адресных регистров БУ ПМП (табл. 46). На третьей линейке могут быть проиндицированы либо одно из слов памяти канала (РГП или ПМК), либо состояние магистрали М1 канала. Коды 0000—1001 клавиатуры «№ РЕГИСТРА» служат для вызова на индикационные элементы третьей линейки одного из слов РГП МК или РГП СК. Для этого нужно на клавиатуре «АДРЕС КАНАЛА», расположенной в правой части нижней панели пульта, набрать код номера канала, на клавиатуре «№ РЕГИСТРА» набрать код номера регистра РГП и нажать кнопку «ЧТЕНИЕ РГП». При этом на второй линейке будет индицироваться информация в соответствии с табл. 46, а на третьей линейке — выбранное слово РГП. На второй линейке при кодах 0000—1001 постоянно индицируется один и тот же тип

Индицирование информации каналов

| Код клавиатуры «№ РЕГИСТРА» | Вторая линейка | | Третья линейка | |
|-----------------------------|---|---------------------------|------------------------------|---------|
| | информация | разряды | информация | разряды |
| 0000—1001 | РАМК[0/10] Триггер ошибки БОП Триггер тактового режима Диагностический счетчик | 1/11 26 27 28/31 | РГП | 0/31 |
| 1010 | РМКК[0/31], КР | 0/31, К | ПМК или М1 ка- нала | 0/31 |
| 1011 | РМКК[32/63], КР | 0/31, К | | |
| 1100 | Соответствует кодам 0000—1001 | | | |
| 1101 | РАСТ[0/10] | 1/11 | | |
| 1110 | РАВМ[0/10] | 1/11 | | |
| 1111 | РАВ[0/10] | 1/11 | | |

информации независимо от того, нажата или нет кнопка «ЧТЕНИЕ РГП».

Коды 1010—1111 клавиатуры «№ РЕГИСТРА» являются запрещенными для чтения РГП (максимальный номер слова РГП СК равен 1001) и используются для индикации одного из слов ПМК на третьей линейке. Для этого необходимо на клавиатуре «АДРЕС УСТРОЙСТВА», расположенной в правой нижней части нижней панели пульта, набрать код адреса подканала, на клавиатуре «№ РЕГИСТРА» в двух ее младших разрядах набрать код номера слова ПМК и нажать кнопку «ЧТЕНИЕ ПМК». При этом на третьей линейке будет индицироваться выбранное слово ПМК. На второй линейке при кодах 1010—1111 индицируются адресные регистры БУ ПМП в соответствии с табл. 46 независимо от того, нажата или нет кнопка «ЧТЕНИЕ ПМК». На третьей линейке при этих кодах и не нажатой кнопке «ЧТЕНИЕ ПМК» постоянно индицируется магистраль М1 того канала, код номера которого набран на клавиатуре «АДРЕС КАНАЛА».

На пульте управления расположены органы, обеспечивающие управление каналами и выполнение пультовых процедур.

Кнопка сброса каналов «СБРОС КАН.» В автономном режиме работы нажатие этой кнопки вызывает процедуру сброса. Последняя заключается в сбросе аппаратуры каналов, ВУ и неразделенных УВУ в исходное состояние. Процедура сброса реализована микропрограммно. Сигнал сброса формируется также при нажатии

кнопок «СБРОС» (сброс системы), «ЗАГРУЗКА» (при начальной загрузке программы), а также микропрограммой включения питания.

Кнопка останова каналов «СТОП КАН.» Нажатием этой кнопки обеспечивается включение схемы останова общего канала по синхросерии С2. Схема останова блокирует поступление сигналов синхронизации С1 и С2 от генератора в аппаратуру ОК, переводя ОК в состояние «Стоп». Текущий такт завершается; блокировка осуществляется, начиная с серии С1 следующего такта. При этом на РА МК находится адрес микрокоманды, считанной из БОП, но выполнение которой заблокировано. В зоне индикации каналов на пульте управления системы загорается лампочка «ОСТ». Вывод каналов из состояния «Стоп» осуществляется нажатием кнопки «ПУСК».

Кнопка пуска каналов «ПУСК КАН.» Нажатием этой кнопки по синхроимпульсу С2 сбрасывается схема останова и, начиная с С1, возобновляется поступление синхросерий в ОК, обеспечивая тем самым продолжение его работы с той микрокоманды, выполнение которой было заблокировано при останове. Лампочка «ОСТ» при этом гаснет.

Кнопки «ЧТЕНИЕ РГП» и «ЧТЕНИЕ ПМК». Нажатие одной из этих кнопок вызывает соответствующую пультовую микропрограмму, которая обеспечивает индикацию слова памяти. Выбор адреса памяти и адреса слова в ней описано выше.

Кнопки «ЗАПИСЬ РГП», «ЗАПИСЬ ПМК». Нажатие одной из этих кнопок вызывает соответствующую пультовую микропрограмму. Последняя осуществляет запись в слово РГП или ПМК информации, набранной на регистре информации пульта управления машины (РИ ПУ расположен в нижней части средней панели). При записи адрес слова выбирается так же, как и при чтении.

Кнопка режима сканирования «СКАН». Нажатие этой кнопки обеспечивает последовательное считывание из БОП4 и БОП5 микрокоманд и проверку их содержимого на четность. Начальный адрес сканирования определяется кодом, набранным на клавиатуре «НАЧАЛЬНЫЙ АДРЕС». Сканирование происходит циклически — после считывания микрокоманды по адресу 7FF, следующей считывается микрокоманда по адресу 000. Слово по адресу 7FF содержит информацию с неправильной четностью и используется для имитации ошибки ОК. Выход из цикла сканирования происходит при нажатии кнопки «СБРОС КАН».

Кнопка пуска по адресу «ПУСК ПО АДР». Нажатием этой кнопки обеспечивается выполнение микропрограммы начиная с адреса, код которого задан клавиатурой «НАЧАЛЬНЫЙ АДРЕС».

Клавиша «ЦИКЛ». Обеспечивает исполнение фрагмента микропрограммы, начальный и конечный адреса которого заданы клавишами клавиатуры «НАЧАЛЬНЫЙ АДРЕС» и «КОНЕЧНЫЙ

АДРЕС». Исполнение фрагмента происходит начиная с начального адреса при нажатии кнопки «ПУСК по АДР».

Клавиша «ТАКТ». Обеспечивает выполнение тактового режима, при котором после нажатия кнопки «ПУСК» исполняется только одна микрокоманда.

Клавиша останова по адресу «ОСТ ПО АДР». Обеспечивает останов при совпадении кода адреса текущей микрокоманды с кодом, набранным на клавиатуре «КОНЕЧНЫЙ АДРЕС».

Клавиша останова по ошибке «ОСТ ПО ОШ». Нажатие этой клавиши обеспечивает останов каналов при возникновении ошибок в данных, в управлении, в сопряжении и при ошибке в цепочке (фиксируются соответственно в РСК[4/7]).

Клавиша блокировки ошибки «БЛ. ОШ». Обеспечивает фиксацию ошибок БОП и фиксацию ошибок в РСК [1/7], но исключает остановки по этим ошибкам. Эта клавиша обеспечивает также блокировку выдачи общим каналом сигнала прерывания при $РСК[8] = 1$.

Клавиша отключение интерфейса «ОТКЛ. ИНТЕРФ». При нажатии этой клавиши блокируются приемники и передатчики сигналов идентификации абонента и каналов. Таким образом исключается взаимное влияние реальных сигналов идентификации и аналогичных сигналов, которые имитируются в диагностических микропрограммах.

Клавиша автономного режима «АВТОН». При нажатии этой клавиши блокируются запросы ЦП и разрешается выполнение пультовых операций и диагностических процедур.

13.2. АППАРАТУРНЫЙ КОНТРОЛЬ

При реализации аппаратного контроля в каналах использованы основные принципы, изложенные в п. 1.4. В каналах применены контроль по mod2 и контроль с использованием дублирующей аппаратуры. При передачах информации внутри каналов используется дополнение до четности, при передаче информации между каналами и ОП и передачах через интерфейс ввода-вывода используется дополнение до нечетности. Контрольным битом снабжается каждый байт информации. Контроль передач информации осуществляется при ее считывании на магистраль схемами контроля, входящими в состав РМ. Дублирование аппаратуры используется при контроле АЛУ. Схемы контроля функционируют непрерывно в процессе работы каналов. На пропускной способности и других параметрах каналов это не отражается.

На рис. 158 показана организация контроля N -го байта информации на магистрали МЗ. На входы А0 — А7 микросхем с магистрали М поступает информация $M [n/(n + 7)]$ из N -го байта, а на входы W0 и W1 — значение контрольного разряда КР этого байта. Если четность информации, принятой на магистраль, правильная, то на выходе F1 ИС формируется сигнал единичного

уровня ЧПМ [N] (четность правильная байта N соответствующей магистрали), который через БК УИ поступает в узел модификации адреса БУ ПМП для формирования адреса последующей микрокоманды. Если четность принятой информации неправильная, на выходе F2 ИС формируется сигнал ошибки. По микроприказу проверки ошибки магистрали ПОМ: = 1 он фиксируется в триггер ошибки магистрали ТОМ [N]. Триггеры ошибок байт магистралей входят в состав условных регистров РГП каждого из каналов (Р5 в МК и Р4 в СК). При индикации на пульте управления РГП5 или РГП4 состояния этих триггеров по микро-

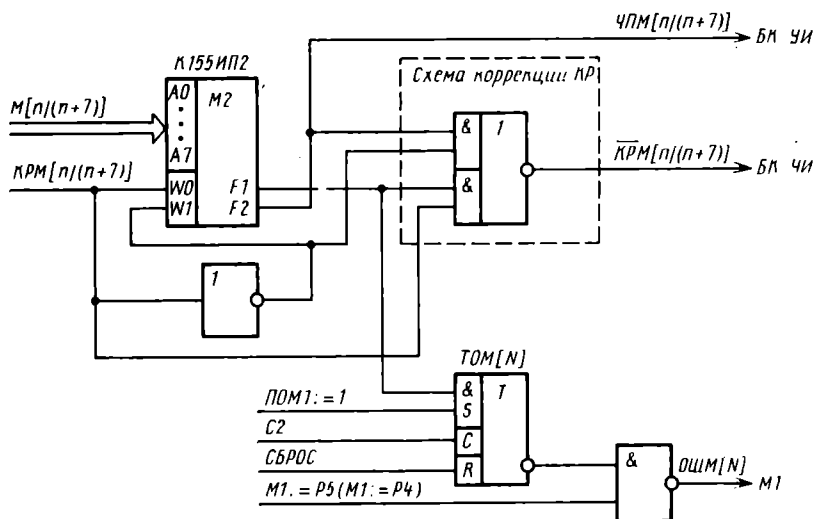


Рис. 158. Схема контроля байта информации на магистрали

приказу $M1: = P5$ ($M1: = P4$) информация считывается на магистраль M1 и поступает на схемы индикации. Организация контроля всех байтов магистралей всех каналов одинаковая.

При вводе данных в ОП, который происходит через магистраль M1 канала, ошибка при считывании на M1 не должна являться причиной, препятствующей записи информации в ОП. Поэтому в случае неверной четности выданной на магистраль информации происходит коррекция значения контрольного разряда байта. Его откорректированное значение КРМ $[n/(n + 7)]$ поступает в БК ЧИ и в дальнейшем сопровождает информацию. Коррекция значения контрольного разряда применяется только в РМ1.

На рис. 159 показана организация контроля АЛУ. Байты $A[n/(n + 7)]$ и $B[n/(n + 7)]$ операндов A и B поступают на входы основного АЛУ и дублирующего АЛУ*, а также на входы A0 — A7 микросхем M2-1 и M2-2. На управляющие входы W0 и W1 этих ИС поступают значения контрольных разрядов КР этих байт. При нарушении четности в байте хотя бы одного операнда ошибка фиксируется в триггере ТОВ АЛУ по управляющему сигналу ТОВАЛУ: = 1. Результат с выхода основного

АЛУ сворачивается по mod2, и полученное значение контрольного разряда вместе с байтом результата поступает на магистраль М3. Результат с выхода дублирующего АЛУ* поступает на другую схему свертки. На нее же подается значение контрольного разряда результата основного АЛУ. Если возникает ошибка, то

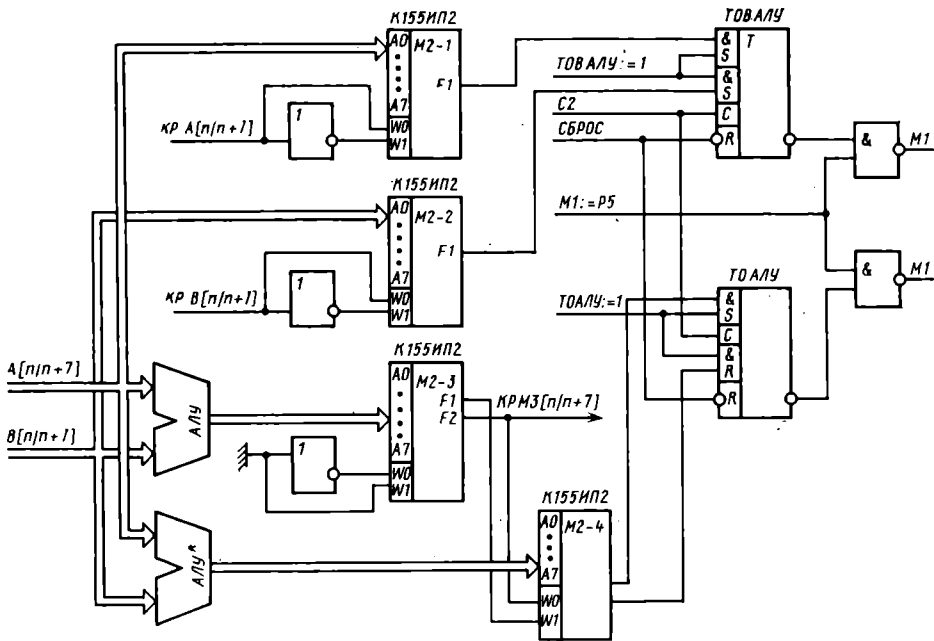


Рис. 159. Схема организации контроля АЛУ

она фиксируется в триггере ошибки АЛУ по управляющему сигналу $TOALU := 1$. Триггеры ошибок АЛУ также входят в состав условного регистра Р5 регистровой памяти каждого канала. При чтении Р5 для индикации на пульте управления состояния триггеров выдаются на магистраль М1 сигналом $M1 := P5$. Для контроля считанного слова микрокоманды в ОК из двух сигналов ошибки ОШ БОП4 и ОШ БОП5 (п. 1.7) формируется суммарный сигнал ошибки БОП.

13.3. ОРГАНИЗАЦИЯ ДИАГНОСТИКИ КАНАЛОВ

Диагностика каналов обеспечивается аппаратными, микропрограммными и программными средствами, которые классифицируют ошибки (сбой или отказ), локализируют место отказа и восстанавливают систему после ошибок в канале ввода-вывода.

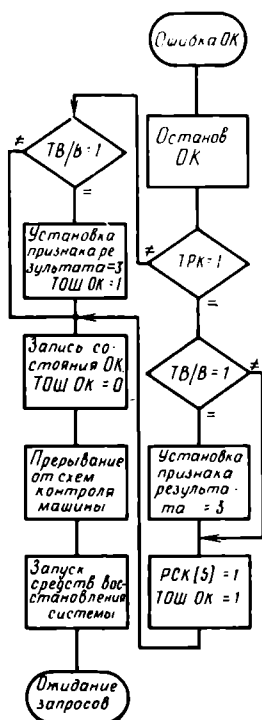
При обработке сигналов ошибок каналов происходит запись в фиксированную область ОП (прил. 10) состояния технических средств (регистровой памяти, управляющих триггеров), которое они имели в момент возникновения ошибки. Эта информация в дальнейшем используется программными средствами восстановления вычислительного процесса в системе, а также при составле-

нии системного журнала, отражающего статистику сбоев. Запись состояния МК и СК осуществляется микропрограммно.

Ошибка чтения управляющей информации из ПМП рассматривается как ошибка ОК и вызывает его останов. При этом невозможно произвести запись состояния ОК, используя ПМП каналов. Поэтому запись состояния ОК производится блоком диагностики ЦП (п. 10.1). После останова ОК (рис. 160) производится аппаратный анализ состояния триггера работы канала (ТРК). Если

до момента останова ОК находился в состоянии связи с одним из каналов ($ТРК = 1$) и выполняя при этом команду ввода-вывода ($ТВ/В = 1$), то схемно устанавливаются признак результата, равный 3, признак ошибки управления в том канале, с которым был связан ОК ($РСК[5] = 1$), а также устанавливается в единичное состояние триггер ошибки ТОШ ОК в ЦП. Общий канал остается в состоянии останова до тех пор, пока программные средства восстановления не предпримут попытку произвести его пуск. Если же ОК, связанный с каналом ввода-вывода, выполнял не команду ввода-вывода ($ТВ/В = 0$), то формирования признака результата не происходит. В случае если ОК в момент ошибки не был связан с каналом ввода-вывода, т. е. выполнял одну из своих подпрограмм ($ТРК = 0$), то установка признака результата происходит только тогда, когда эта подпрограмма относилась к обработке команды ввода-вывода.

Рис. 160. Схема алгоритма обработки ошибки общего канала



Процессор аппаратно анализирует состояние ТОШ ОК после окончания выполнения очередной команды и перед началом выполнения следующей.

Разрешение на этот анализ дается микроприказом АН38 (п. 10.1). Если $ТОШ ОК = 1$ и прерывания разрешены системой ($РССП[13] = 1$), то происходит прерывание программы, которое обрабатывается так же, как и прерывание от схем контроля ЦП. Сигналом $ТОШ ОК = 1$ индицируется работа блока диагностики ЦП, который производит запись в фиксированную область ОП состояния ОК в момент ошибки. При этом записывается следующая информация:

| Адрес ОП | Содержимое |
|----------|------------------|
| 1A8 | РА МК |
| 1AC | ТРК, ТБЛ НК, РНК |
| 1B0 | РАВ |
| 1B4 | РАВМ. |

По окончании записи состояния ОК блок диагностики ЦП завершает обработку прерывания путем замены ССП. В старом ССП записывается код прерывания от схем контроля ЦП. Тот факт, что это прерывание вызвано ошибкой ОК, отражается соответствующим кодом в ячейке ОП с адресом 1A0 (см. табл. 36). Новое ССП передает управление программным средствам восстановления ССН (п. 14.2). Программа восстановления с помощью команды ДИАГНОСТИКА делает попытку пустить ОК для перевода его в режим ожидания (на подпрограмму ожидания запросов). Такая попытка предпринимается N раз, где N — наперед заданное число. Если при n -й попытке ($1 \leq n \leq N$) это удастся, событие классифицируется как сбой, в противном случае — как отказ.

Общий алгоритм обработки ошибок МК и СК при выполнении операций ввода-вывода представлен на рис. 161. При выполнении каналом команды ввода-вывода на этапе, когда канал связан с ЦП, а признак результата еще не сформирован, могут возникнуть ошибки типа S2, S3, S5 и S6 (табл. 45). Установка признаков этих ошибок в байте состояния канала происходит либо при обнаружении аппаратурных S5 и S6, либо программных S2 и S3 ошибок (выход А). При анализе БСУ и признака цепочки команд ошибками являются несоответствие кода БСУ значениям 00, 08, 48, 0C или 4C (см. табл. 44), либо отсутствие признака ЦК при БСУ = 08, 48, 0C или 4C (выход В).

Ошибки, возникшие при выполнении операций ввода-вывода после того, как был сформирован признак результата и канал отключился от процессора, вызывают установку признака прерывания в канале (PCK[8] := 1, выход С). Канал переводится в режим ожидания на подпрограмму ожидания запросов. Если прерывания от соответствующего канала системой разрешены

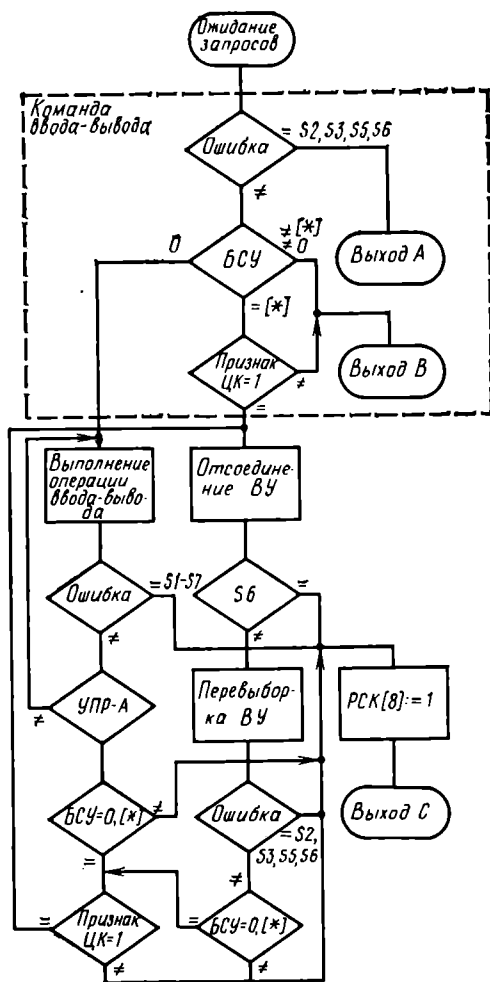


Рис. 161. Схема общего алгоритма обработки ошибок МК и СК: [*] — БСУ = 08 ∨ 48 ∨ 0C ∨ 4C

(прил. 7), то происходит обработка прерывания от каналов (вход С, рис. 162).

Обработка сигналов ошибок МК и СК при выполнении команды ввода-вывода, а также обработка прерываний в случае ошибок, возникающих при выполнении операций ввода-вывода, происходит по единому алгоритму, который представлен на рис. 162. Здесь входы А, В и С соответствуют выходам А, В и С на рис. 161. Этот алгоритм реализуется общим каналом микропрограммно.

В случае ошибок S2 и S3 происходят запись ССК, установка признака результата и выход канала из связи с процессором. В дальнейшем система, используя признак результата, записанный процессором в ССП, производит анализ состояния канала по содержимому ССК. При ошибках управления S5 и S6 происходит запись состояния технических средств канала в фиксированную область ОП. В этом случае в ОП записывается содержимое РГП канала слово за словом начиная с нулевого. Каждое слово РГП помещается в два соседних слова ОП. В первое из них

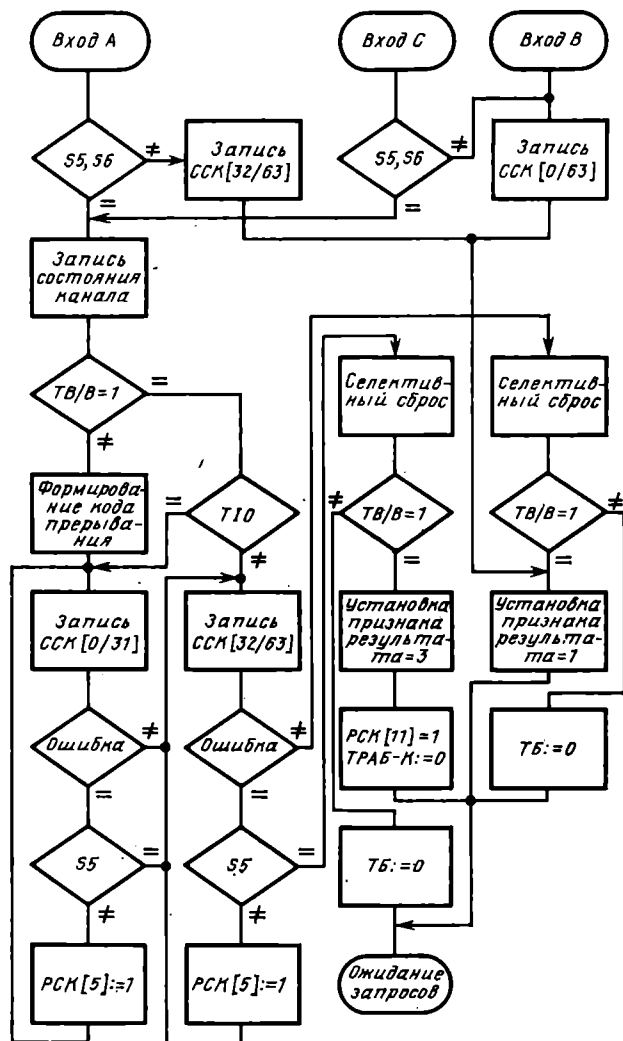


Рис. 162. Схема алгоритма обработки ошибок управления

помещается содержимое слова РГП, а во второе — значения контрольных разрядов этого слова. Таким образом, РГП МК занимает в ОП ячейки с адресами 1СС — 200, а РГП СК — 1СС — 218.

После записи состояния канала в фиксированную область ОП осуществляется анализ условий входа в процедуру обработки ошибки. Нулевое состояние ТВ/В, отражающее при наличии ошибки состояние прерывания в канале, вызывает формирование

кода прерывания в РБСП и запись полного слова состояния канала.

Если ошибка возникла в процессе выполнения команды ввода-вывода ($TB/V = 1$), то запись полного ССК происходит только при возникновении ошибки в процессе выполнения команды ОСТАНОВИТЬ ВВОД-ВЫВОД. В противном случае происходит запись только второго слова ССК.

Ошибка, возникающая в процессе записи ССК, рассматривается как ошибка в управлении каналом. Если до возникновения этой ошибки состояние РСК [5] было нулевым, то происходит фиксация этой ошибки в РСК [5] и возвращение на начало записи соответствующего слова. Если до возникновения этой ошибки состояние РСК [5] было единичным, то ошибка считается двухкратной,

При ошибке, возникающей в процессе записи первого слова, осуществляется запись второго слова. В этом случае наличие ошибки при первичной записи второго слова рассматривается как двойная ошибка в управлении каналом. Двойная ошибка приводит к установке РСК [11] в единичное состояние, т. е. происходит фиксация факта неработоспособности канала.

Безошибочная запись первой половины ССК и двухкратная ошибка в процессе записи второй половины также приводят к единичному состоянию РСК [11].

Двухкратная ошибка в процессе записи первой половины ССК не вызывает установки РСК [11] в единичное состояние, если ошибка не возникает во время записи второй половины.

Запись слова состояния канала имеет два окончания: безошибочное, когда отсутствует ошибка при записи второй половины ССК, и окончание по ошибке ($РСК [5] = 1$) при записи второй половины ССК. В обоих случаях происходят селективный сброс и анализ TB/V .

Нулевое состояние TB/V как в безошибочном окончании, так и в окончании по ошибке обеспечивает завершение каналом обработки прерывания. Отличие состоит в том, что в окончании по ошибке канал переводится в нерабочее состояние ($РСК [11] := 1$) и отключается от абонентов. Единичное состояние TB/V в окончании по ошибке вызывает формирование признака результата 3 перед переводом канала в нерабочее состояние. В безошибочном окончании единичное состояние TB/V обеспечивает формирование признака результата 1.

Процедура обработки ошибок управления каналом и ошибок управления интерфейсом завершается переводом ОК в состояние ожидания.

Перевод канала в нерабочее состояние означает, что нельзя гарантировать достоверности в ССК байта состояния устройства и байта состояния канала. Канал исключается из системы, ошибка классифицируется как отказ. Локализация места отказа осуществляется с помощью ТЛН в автономном режиме.

При безошибочном окончании инициируется работа программы ССН, которая делает N попыток восстановления вычислительного процесса, где N — некоторое наперед заданное число. Если при n -й попытке ($1 \leq n \leq N$) не возникает ошибок S5 и S6, то событие классифицируется как сбой. При $n > N$ фиксируется отказ и осуществляется его локализация с помощью ТЛН в автономном режиме.

13.4. ТЕСТЫ ЛОКАЛИЗАЦИИ НЕИСПРАВНОСТИ

Тесты локализации неисправности (ТЛН) являются микропрограммными. Они обеспечивают локализацию места отказа при проверке оборудования МК и той части СК, которая управляется микропрограммно. Тесты хранятся в ПМП, входящей в состав ОК. Выполнение их инициируется с пульта управления в автономном режиме работы каналов и требует исправности ПМП, ОК, а также ОП, которые составляют в данном случае так называемое диагностическое ядро. Проверка перечисленных узлов производится средствами пульта управления (п. 13.1).

Различают три уровня структурной организации ТЛН: пример, тест и процедура. Пример обеспечивает проверку одного или более однотипных регистров, групп или отдельных триггеров с помощью определенного тестового стимула. Критерием объединения примеров в тесты и тестов в процедуры является, во-первых, функциональная зависимость объектов тестирования друг от друга и, во-вторых, возможность реализации метода расширения диагностического ядра. Пример завершается успешно, если в ходе его выполнения микропрограммным анализом обнаружено наличие сигнала ошибки, сформированного аппаратурными средствами контроля. Ошибкой, завершающей пример, является также обнаруженное микропрограммным анализом несоответствие логического состояния объекта тестирования эталонному. Наличие ошибки вызывает останов каналов. При этом адрес микрокоманды (АМК) останова сохраняется в РА МК.

Сущность проверки сводится к записи в регистр констант определенного вида с последующим чтением содержимого регистра на магистраль и проверкой наличия ошибок. Для проверки самих схем контроля используются константы, имеющие неправильную четность одного из байт. Проверка другого вида заключается в записи в регистр констант с последующим анализом содержимого регистра. В основном применяются константы 0000 0000, 55555 55555, ААААА ААААА и 0101 0101, имеющие правильную четность, и константы 5555 5555 и АААА АААА, имеющие неправильную четность байт. Адреса ОП, используемые в ТЛН, содержатся в ПМП также в виде констант. Для проверки отдельных триггеров осуществляются их микропрограммная установка и сброс с последующим анализом состояния.

Любая из процедур P_i , входящая в состав ТЛН (рис. 163), идентифицируется адресом первой микрокоманды, который яв-

ляется адресом процедуры AP_i в ПМП. Адрес каждой очередной выполненной микрокоманды сохраняется в РАСТ в процессе выполнения любой микропрограммы. Чтобы сохранить в РАСТ адрес микрокоманды процедуры, производится блокировка записи содержимого РА МК в РАСТ по микроприказу БЛ РАСТ в первой микрокоманде процедуры. Здесь же в РАВМ записывается адрес

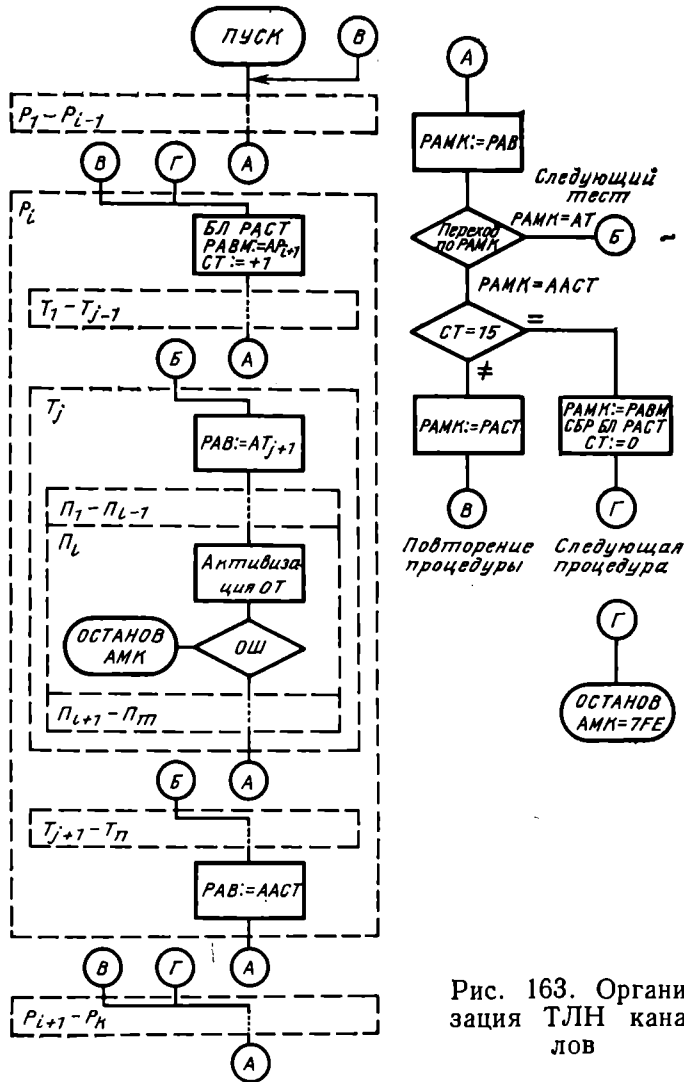


Рис. 163. Организация ТЛН каналов

следующей процедуры AP_{i+1} и увеличивается на единицу содержимое диагностического счетчика СТ, который был сброшен перед началом ТЛН. В конце любой процедуры P_i в РАВ записывается адрес анализа счетчика $РАВ := ААСТ$. После неуспешного окончания процедуры (выход А) происходит пересылка адреса из РАВ в РА МК и микропрограмма продолжает исполняться по текущему адресу, находящемуся в РА МК. После окончания процедуры это будет адрес микрокоманды анализа счетчика АА СТ. Если содержимое счетчика меньше 15 ($СТ < 15$), то в РА МК из РАСТ записывается адрес той же процедуры, и она повторяется.

Если $CT = 15$, то в РА МК из РА ВМ записывается адрес следующей процедуры, снимается блокировка РА СТ (СБР БЛ РА СТ), сбрасывается счетчик СТ и происходит переход к следующей процедуре.

Любой из тестов T_j идентифицируется адресом процедуры AP_i , в состав которой он входит и начальным адресом следующего теста T_{j+1} . В начале выполнения теста адрес T_{j+1} записывается в РА В. Если тест кончается неуспешно, т. е. в процессе его выполнения ошибки не обнаружено (выход А), то в РА МК записывается из РА В адрес T_{j+1} и происходит переход к следующему тесту.

Любой из примеров P_i идентифицируется адресом процедуры P_i , адресом T_{j+1} теста T_j , в состав которого входит пример, и адресом микрокоманды АМК, на которой происходит останов при обнаружении ошибки примером (АМК сохраняется в РА МК). Если примером ошибка не обнаружена, происходит переход к следующему примеру, адрес которого задается в конце текущего примера. В конце последнего теста в РА ВМ записывается адрес последней процедуры, которая завершает ТЛН и производит останов на адресе АМК = 7FF.

Таким образом, после останова по ошибке пример может быть идентифицирован содержимым РА СТ, РА В и РА МК. В свою очередь, пример однозначно определяет объект тестирования, т. е. неисправное оборудование.

Поиск места отказа осуществляется с помощью журнала неисправностей, состоящего из трех разделов:

- объект тестирования;
- результат первичного теста;
- активная цепь.

Первый раздел позволяет по содержимому РА, РА В и РА СТ определить объект тестирования и идентификатор раздела «Результат первичного теста».

Во втором разделе приведены следующие сведения:

- эталонное состояние магистралей;
- эталонное состояние регистровой памяти;
- совокупность конструктивных адресов, определенных в результате первичного тестирования;
- адреса микропрограммы для зацикливания теста, обнаружившего отказ;
- идентификатор раздела «Активная цепь».

Третий раздел определяет конструктивный адрес источника сигнала, идентификатор сигнала, номер канала, совокупность контрольных точек (активную цепь), используемых при осциллографировании.

13.5. ПРОГРАММНЫЕ ТЕСТЫ

Программные тесты загружаются в оперативную память с внешнего носителя. Они построены по методу расширения диагностического ядра и обеспечивают проверку оборудования селекторного

канала, которое не имеет микропрограммного управления. Это оборудование осуществляет передачу данных в селекторном канале и работает под управлением тактовой цепочки (ТЦ). Совокупность программных тестов составляет тест селекторных каналов.

Запуск программных тестов производится после неуспешного исхода микропрограммных ТЛН, которые обеспечивают диагностическое ядро в селекторном канале. В состав диагностического ядра при реализации программных тестов входят также оперативная память, центральный процессор, общий канал, блок пультовых режимов и диагностики, а также канал, обеспечивающий загрузку тестов и вывод результата тестирования.

Тактовая цепочка ТЦ представляет собой ряд последовательно включенных триггеров. Установка и сброс каждого из них синхронизируются одним из синхроимпульсов С1 или С2. Таким образом, один такт работы ТЦ равен одному машинному полутакту.

Реализация программного теста в селекторном канале заключается в выполнении определенных действий под управлением тактовой цепочки СК, завершающихся остановом на определенном такте ТЦ с последующей записью состояния СК в фиксированную область ОП. На основании анализа результата сравнения содержимого области записи состояния с его эталонным значением программа дает заключение о работоспособности проверяемого оборудования. Результат тестирования выводится на печать.

Для осуществления программного тестирования селекторного канала в узле диагностики каналов предусмотрена схема тактового режима СК, обеспечивающая пуск и останов тактовой цепочки после выполнения определенного количества тактов.

Тактовая цепочка СК имеет пять точек сопряжения с микропрограммным управлением общего канала, позволяющих осуществить запуск тактовой цепочки в любой из этих пяти точек путем формирования общим каналом микроприказов:

начать ввод-вывод;

продолжить;

модификация;

ИНФ-А;

разрешение передачи данных.

После выполнения определенного числа тактов ТЦ и останова вторичный пуск может быть произведен с любого такта.

Программные тесты используют команду ДИАГНОСТИКА, которая является привилегированной командой. Выполнение той или иной диагностической процедуры определяется управляющим словом этой команды — УСКД. После выборки команды процессор, на основании анализа значения восьмого разряда поля I2 команды, передает выполнение команды в каналы, если значение этого разряда равно нулю. При этом ЦП устанавливает в единичное состояние триггер диагностики каналов и записывает УСКД в РБСП. При единичном состоянии этого триггера микропрограмма ЦП зацикливается на его анализе, а ОК переписывает содержимое

РБСП в соответствующие регистры. Затем начинается выполнение команды ДИАГНОСТИКА в каналах. После ее завершения канал сбрасывает триггер диагностики каналов, а процессор по нулевому значению этого триггера, в свою очередь, завершает выполнение команды ДИАГНОСТИКА в ЦП и начинает выборку следующей команды.

В общем случае выполнение команды ДИАГНОСТИКА в канале приводит к запуску микропрограммы, начальный адрес которой задается в УСКД[1/11] (табл. 47). Таким образом, команда ДИАГНОСТИКА позволяет осуществить переход с системного уровня на микропрограммный.

Т а б л и ц а 47

Распределение полей УСКД

| Функция команды ДИАГНОСТИКА | Адрес БОП | Счетчик | К | Шина-А |
|-----------------------------------|-----------|---------|----|-----------------|
| | 1/11 | 17/20 | 23 | 24/31 |
| Пуск общего канала | 100 | 0 0 0 0 | 0 | 0 0 0 0 0 0 0 0 |
| Сброс каналов | 00F | 0 0 0 0 | 0 | 0 0 0 0 0 0 0 0 |
| Запись состояния канала | 0B1 | 0 0 0 0 | 0 | 0 0 0 0 0 0 0 0 |
| Установка тактового режима | 2B4 | 0 0 0 0 | 0 | 0 0 0 0 0 0 0 0 |
| Сброс тактового режима | 2B5 | 0 0 0 0 | 0 | 0 0 0 0 0 0 0 0 |
| Пуск СК по С1 | 2B3 | X X X X | 0 | 0 0 0 0 0 0 0 0 |
| Пуск СК по С2 | 2B2 | X X X X | 0 | 0 0 0 0 0 0 0 0 |
| Начать ввод-вывод | 2BC | X X X X | 0 | 0 0 0 0 0 0 0 0 |
| Продолжить | 2B0 | X X X X | 0 | 0 0 0 0 0 0 0 0 |
| Модификация | 2B6 | X X X X | 0 | 0 0 0 0 0 0 0 0 |
| Обмен | 2B8 | X X X X | X | X X X X X X X X |
| Сброс ИНФ-А | 2B7 | X X X X | 0 | 0 0 0 0 0 0 0 0 |
| Разрешение передачи данных | 61A | X X X X | 0 | 0 0 0 0 0 0 0 0 |

П р и м е ч а н и е. X — состояние разряда безразлично.

УСКД[17/20] задает количество тактов, которое должно быть выполнено тактовой цепочкой селекторного канала при запуске ее по команде ДИАГНОСТИКА. При этом пуск тактовой цепочки осуществляется диагностической микропрограммой, начальный адрес которой задается в УСКД[1/11], т. е. осуществляется переход с микропрограммного уровня на аппаратурный.

УСКД[24/31] содержит байт, имитирующий информацию на ШИН-А при блокировке приемников и передатчиков с помощью клавиши «ОТКЛ. ИНТЕРФ». Контрольный разряд этого байта задается УСКД[23].

При использовании команды ДИАГНОСТИКА в программных тестах количество тактов, которое должно быть выполнено ТЦ, записывается в диагностический счетчик СТ, а содержимое УСКД[1/11] — в РА МК. Пуск ТЦ производится в микропрограмме обработки УСКД по микроприказу ТПС1: = 1 или ТПС2: = 1 в зависимости от того, начиная с какого полутакта, С1 или С2, должна быть запущена ТЦ (ТПС1 и ТПС2 — триггеры пуска по С1 или С2 соответственно). Останов ТЦ происходит по результату анализа содержимого счетчика СТ.

Установка тактового режима перед проверкой СК и сброс его после проверки осуществляются также с помощью команды ДИАГНОСТИКА. Кроме того, как уже описывалось выше, команда ДИАГНОСТИКА используется программой восстановления ССН для пуска ОК при ошибках в нем, для выхода на микропрограмму сброса каналов, записи состояния канала в фиксированную область ОП, а также для выполнения других диагностических процедур.

ГЛАВА 14

ПРОГРАММНЫЕ СРЕДСТВА ВОССТАНОВЛЕНИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА

14.1. УРОВНИ ВОССТАНОВЛЕНИЯ

При работе ЭВМ ЕС-1033, как и любой другой достаточно сложной ЭВМ, неизбежно возникают ошибки обработки информации, приводящие к нарушению вычислительного процесса.

Ошибки бывают либо случайными, либо устойчивыми. Если ошибки случайные, их последствия могут быть нейтрализованы аппаратурными средствами восстановления путем повторения команды, на которой произошла ошибка. Если ошибки устойчивые, требуются дополнительные действия по поиску и устранению причин их возникновения. Ошибки разделяются на ошибки ввода-вывода и так называемые машинные ошибки.

Ошибкой ввода-вывода считается ошибка, обнаруженная каналом ввода-вывода, устройством управления или внешним устройством при выполнении программы канала. Ошибкой ввода-вывода считают также особую ситуацию, обнаруженную каналом ввода-вывода, внешним устройством или устройством управления, требующую программной обработки.

Машинной ошибкой принято называть ошибку, вызывающую прерывание от схем контроля машины. Для ЭВМ ЕС-1033 это ошибки центрального процессора, ошибки общего канала, опера-

тивной памяти, памяти ключей защиты и диагностического обслуживания.

Для повышения эффективности использования машины, ускорения поиска и устранения причин возникновения ошибок в состав операционной системы ЭВМ ЕС-1033 включены программные средства восстановления (ПСВ) вычислительного процесса, а также средства регистрации информации о состоянии аппаратуры ЭВМ и операционной системы в момент возникновения ошибки [4, 41].

В зависимости от характера ошибок ПСВ разделяются на средства восстановления вычислительного процесса после ошибок ввода-вывода и средства восстановления вычислительного процесса после машинных ошибок.

К первым относятся:

программа обработки ошибок каналов — ССН (CHANNEL CHECK HANDLER);

программа динамической реконфигурации периферийных устройств — DDR (DYNAMIC DEVICE RECONFIGURATION);

программа повторения операций ввода-вывода по альтернативному пути — АРР (ALTERNATE PATH RETRY);

программа регистрации состояния каналов и периферийных устройств в момент ошибки — ОВР (OUTBOARD RECORD);

программа обработки ошибок для периферийных устройств — ЕРР (ERROR RECOVERY PROCEDURE);

программа обработки недостающих прерываний ввода-вывода — МИС (MISSING INTERRUPT CHECK).

Ко вторым — программа обработки прерываний от схем контроля — МСН (MACHINE CHECK HANDLER) и программа регистрации состояния и восстановления — СЕР (SYSTEM ERROR RECOVERY).

К ПСВ также относятся программы редактирования и печати записей об ошибках — ЕРЕР (ERROR RECORD EDITING AND PRINTING PROGRAM) и СЕРЕР (SEPARATE ERROR RECORD EDITING AND PRINTING PROGRAM), а также программа IFCDIPOO, инициализирующая внешнюю память для набора данных SYS1. LOGREC, в котором происходит регистрация информации об ошибке.

Программы IFCDIPOO, ОВР, ЕРР и ЕРЕР включаются в генерируемый конкретный вариант ОС ЕС в обязательном порядке, остальные — по выбору пользователя.

Программные средства восстановления вычислительного процесса разделяются на моделезависимые и моделенезависимые. Моделенезависимые программы работают на любой из моделей ЕС ЭВМ, если они включены в операционную систему при генерации. Моделезависимые программы используют для своей работы диагностическую информацию, записанную в фиксированную область ОП — область записи состояния, и полностью зависят от возможностей модели, для которой они разработаны. Ниже коротко характеризуются основные моделенезависимые про-

граммы. Моделезависимые программы CCH, MCH, SER и EREP обсуждаются подробнее и описаны в отдельных разделах.

Программа IFCDIP00 предназначена для инициализации и распределения внешней памяти для набора данных SYS1. LOGREC, который является в ОС ЕС журналом регистрации записей об ошибках. Он создается на резидентном томе при генерации операционной системы и является обязательным. При необходимости изменения размера набора данных SYS1. LOGREC или его восстановления в результате повреждения также выполняется программа IFCDIP00. Она вызывается из системной библиотеки SYS1. LINKLIB с помощью операторов языка управления заданиями ОС ЕС.

Программа OBR является компонентом супервизора ввода-вывода ОС ЕС. Она собирает в наборе данных SYS1. LOGREC статистическую информацию об ошибках периферийных устройств и о состоянии операционной системы в момент фиксирования постоянной ошибки. Эта программа также включает записи о временной ошибке канала ввода-вывода, созданные программой CCH, в набор данных SYS1. LOGREC.

Программа DDR динамической реконфигурации устройств позволяет оператору ЭВМ осуществлять обмен томов на накопителе на магнитных дисках или накопителе на магнитной ленте, а также производить переключение запросов на ввод-вывод с одного устройства ввода-вывода на другое во время их использования операционной системой или пользователем ЭВМ с целью обхода постоянной ошибки ввода-вывода. Программа DDR определяет возможность обмена томов или переключения запросов, обеспечивает передачу информации между управляющими блоками ОС ЕС для устройств ввода-вывода, выдает указание оператору ЭВМ о перестановке нужного тома на новое устройство и возвращает управление через супервизор ввода-вывода программе ERP для повторения операции ввода-вывода.

Программа APR обеспечивает поиск альтернативного пути для доступа к заданному периферийному устройству при обнаружении программой CCH постоянной ошибки в канале ввода-вывода. Результат поиска передается программе ERP для повторения операции ввода-вывода. Для обеспечения эффективной работы программы APR необходимо, чтобы при генерации ОС ЕС к периферийным устройствам были указаны альтернативные пути доступа через другие каналы ввода-вывода.

Программа обработки недостающих прерываний ввода-вывода MIS обеспечивает проверку операции ввода-вывода на периферийном устройстве в пределах определенного интервала времени. Получая управление через определенный интервал времени, она выявляет факт отсутствия прерывания ввода-вывода и моделирует такое прерывание, завершая запрос на ввод-вывод.

Программа обработки сигналов ошибок ERP представляет собой набор стандартных процедур обработки ошибок периферий-

ных устройств. Для каждого устройства существует своя процедура ERP. Получив управление в случае ошибки периферийного устройства (или ошибки канала, если в системе есть программа CCH), ERP анализирует и классифицирует ошибку. В большинстве случаев она пытается повторить операцию ввода-вывода. Если повторение невозможно, оператору выдается сообщение о постоянной ошибке ввода-вывода. Всякий раз в набор данных SYS1. LOGREC заносится запись об ошибке с помощью программы OBR.

Целью работы программных средств восстановления является достижение одного из следующих четырех уровней восстановления вычислительного процесса в системе:

1. Функционального восстановления — сохранения работоспособности ОС ЕС и всех задач, выполнявшихся в системе.

2. Системного восстановления — сохранения работоспособности операционной системы за счет выборочного окончания поврежденных и продолжения выполнения неповрежденных задач.

3. Системно-обеспечиваемого рестарта — восстановления работоспособности ОС ЕС посредством ее начальной загрузки с возможностями системного рестарта, позволяющими сохранять имеющиеся системные очереди заданий и данных.

4. Ремонта — перевода центрального процессора в состояние ожидания, означающего необходимость вмешательства технического персонала.

Моделезависимые программы восстановления CCH, MCH и SER обеспечивают наиболее высокий уровень восстановления после возможных в конкретной ситуации ошибок технических средств ЭВМ ЕС-1033.

14.2. ВОССТАНОВЛЕНИЕ ПОСЛЕ ОШИБОК ВВОДА-ВЫВОДА

Супервизор ввода-вывода ОС ЕС обеспечивает управление вводом-выводом и стандартную обработку возникающих при этом ошибок. Ошибка в управлении каналом S5 и ошибка в управлении интерфейсом S6 являются критическими для супервизора ввода-вывода. Для обработки этих ошибок супервизор ввода-вывода передает управление программе обработки канальных ошибок CCH, если она включена в операционную систему.

Программа CCH получает управление и при обнаружении в слове состояния канала индикации об ошибке данных в канале S4. Основная цель этой программы — сформировать табличную восьмибайтную запись ERPIB (ERROR RECOVERY PROCEDURE INTERFACE BLOCK), информация из которой служит для последующего восстановления вычислительного процесса, и статистическую запись об ошибке IBR (INBOARD RECOVERY).

Программа CCH состоит из двух основных функциональных частей: ядра и моделезависимых модулей. Ядро выполняет общие для различных моделей ЕС ЭВМ функции:

определение типа канальной ошибки;

сбор информации о состоянии операционной системы в момент ошибки;
 построение статистической записи об ошибке;
 построение части записи ERP1B.

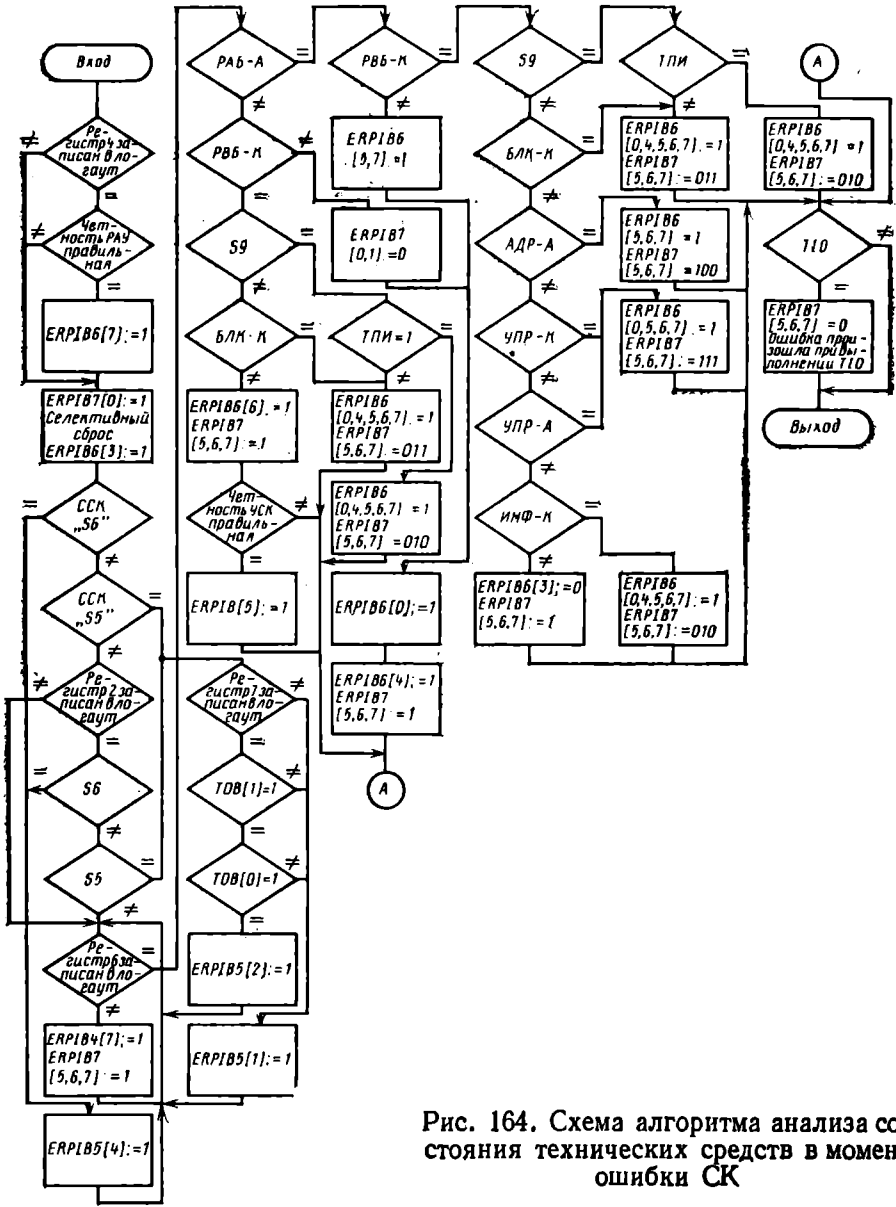


Рис. 164. Схема алгоритма анализа состояния технических средств в момент ошибки СК

Ядро ССН является резидентным и включается в ядро операционной системы при ее генерации с помощью параметра `OPTIONS = ССН` в макрокоманде `SUPRVSOR`.

Модельезависимая часть программы ССН для ЭВМ ЕС-1033 представлена в виде двух модулей: модуля анализа состояния технических средств в момент возникновения ошибки селекторного канала (IGFCCN63) и модуля анализа состояния технических

средств в момент возникновения ошибки мультиплексного канала (IGFCCN73).

Модули анализа на основе находящейся в ОП информации о состоянии машины определяют возможный источник ошибки, устанавливают индикаторы достоверности информации, форми-

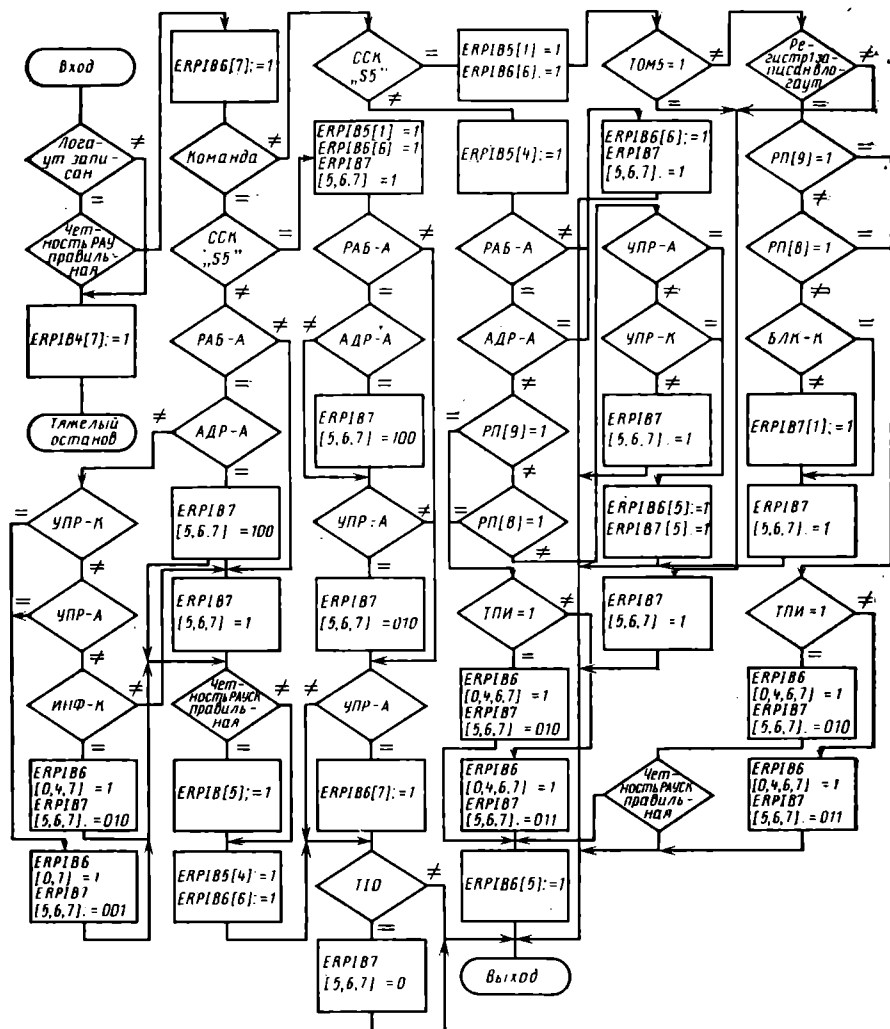


Рис. 165. Схема алгоритма анализа состояния технических средств в момент ошибки МК

руют код завершения, определяющий сигналы интерфейса, использованные для завершения операции ввода-вывода из-за канальной ошибки, и код последовательности, определяющий последовательность действий системы ввода-вывода в момент обнаружения ошибки. Результатом работы этих модулей является формирование 5, 6 и 7 байт таблицы ERPIB (табл. 48). Первые четыре байта ERPIB формируются ядром ССН.

Алгоритмы анализа состояния технических средств в момент ошибки, реализованные в модулях IGFCCN63 и IGFCCN73, приведены на рис. 164 и 165.

Формат записи ERPIB

| Байт | Значение бит | Описание поля |
|------|---------------------------|---|
| 0 | | Резерв |
| 1—3 | | Адрес UCSB для устройства, работающего в момент обнаружения ошибки в канале |
| 4 | | Программные флажки (указывают, когда было записано ССК) |
| | 1 | ССК записано после выполнения команды НАЧАТЬ ВВОД-ВЫВОД |
| | . 1 | ССК записано во время обработки прерывания ввода-вывода |
| | . . 1 | ССК записано после выполнения команды ПРОВЕРИТЬ ВВОД-ВЫВОД |
| | . . . 1 | ССК записано после выполнения команды ОСТАНОВИТЬ ВВОД-ВЫВОД |
| | X | резерв |
| | 1 | записана информация уточненного состояния |
| | 1 | поле счета в ССК правильное |
| | 1 | повторение невозможно |
| 5 | | Возможный источник ошибки: |
| | 1 | процессор |
| | . 1 | канал |
| | . . 1 | устройство управления памятью |
| | . . . 1 | оперативная память |
| | 1 | устройство управления ВУ |
| | X X X | резерв |
| 6 | | Индикаторы достоверности информации: |
| | 1 | адрес по интерфейсу ввода-вывода правильный |
| | . X X | резерв |
| | . . . 1 | код последовательности правильный |
| | 1 | состояние устройства правильное |
| | 1 | адрес команды правильный, т. е. в ССК указан правильный адрес команды |
| | 1 | адрес канала правильный |
| | 1 | адрес устройства правильный |
| 7 | | Коды завершения и коды последовательности (коды повторения): |
| | X X | код завершения (определяет сигналы интерфейса ввода-вывода, используемые для завершения операции из-за ошибки в канале) |
| | 0 0 | отключение от интерфейса |
| | 0 1 | последовательность сигналов ОСТАНОВ, ЗАПОМНИТЬ СОСТОЯНИЕ или НОРМАЛЬНОЕ ЗАВЕРШЕНИЕ ОПЕРАЦИИ |
| | 1 0 | селективный сброс |
| | 1 1 | системный сброс |
| | . . X X | резерв |

| Байт | Значение бит | Описание поля |
|------|---------------------|---|
| | 1 | сообщение об ошибке ввода-вывода, указывающее, что данное периферийное устройство выдало по интерфейсу сигнал об ошибке ввода-вывода, которая не позволяет устройству установить нормальную связь с каналом (получив этот сигнал, канал выполняет селективный сброс и устанавливает в ССК бит СБОЙ УПРАВЛЕНИЯ ИНТЕРФЕЙСА) |
| | XXX | код, определяющий последовательность действий системы ввода-вывода в момент обнаружения ошибки: |
| | 0 0 0 | ошибка произошла во время выполнения команды ПРОВЕРИТЬ ВВОД-ВЫВОД |
| | 0 0 1 | канал послал команду в периферийное устройство, но еще не принял состояния устройства |
| | 0 1 0 | команда принята периферийным устройством, но передача данных еще не началась |
| | 0 1 1 | по крайней мере, 1 байт данных передан через интерфейс |
| | 1 0 0 | команда не передана в периферийное устройство или устройство не приняло команду |
| | 1 0 1 | команда периферийным устройством принята, но передача данных непредсказуема |
| | 1 1 0 | резерв |
| | 1 1 1 | устанавливается в случае, если предыдущие последовательности неприменимы |

Во время инициализации ядра операционной системы модель-зависимые модули программы ССН выбираются из библиотеки SYS1.LINKLIB и включаются в резидентную часть операционной системы.

При ошибке данных в канале S4 запись ERPIB не строится, а создается только статистическая запись об ошибке IBR.

Завершив свои функции, программа ССН возвращает управление в супервизор ввода-вывода, который планирует выполнение программы избавления от ошибок ERP.

Программа ERP, получив управление в результате ошибки канала, анализирует запись ERPIB и принимает окончательное решение о возможности повторения операции ввода-вывода.

Если повторение операции ввода-вывода возможно, программа ERP возвращает управление супервизору ввода-вывода для организации повторения операции ввода-вывода. При успешном повторении достигается высший уровень восстановления — функциональный. При неуспешном повторении операции ввода-вывода обеспечивается системное восстановление.

В тех случаях, когда программа ССН определяет условия аварийного завершения работы вычислительной системы, она

передает управление той программе обработки прерываний от схем контроля, которая включена в конкретный вариант операционной системы (SER или MCH). Программа SER в этом случае сама строит запись о канальной ошибке и помещает ее в набор данных SYS1.LOGREC. Программа MCH помещает в набор данных SYS1.LOGREC запись, построенную программой CCH.

14.3. ВОССТАНОВЛЕНИЕ ПОСЛЕ МАШИННЫХ ОШИБОК

Программное восстановление вычислительного процесса после возникновения машинных ошибок в ЭВМ EC-1033 обеспечивается моделязависимыми программами SER и MCH.

Программы SER и MCH различаются возможностями в достижении высших уровней восстановления. Эти программы взаимно исключают друг друга. Возможность включения той или иной программы в процессе генерации конкретного варианта определяется значением параметра SER макрокоманды SUPRVSOR.

Программы SER и MCH используют диагностическую информацию из области записи состояния ЦП. Особое значение для анализа ситуации и обеспечения уровней восстановления имеет информация в той части области записи состояния ЦП, которую называют дополнительной (полуслово в адресом 1A0, см. табл. 35). По значению его разрядов различают следующие типы прерываний от схем контроля:

- 1) ошибка ОК;
- 2) сбой ЦП (команда повторялась аппаратурными средствами);
- 3) сбой ЦП (тесты локализации неисправности отказ не обнаружили);
- 4) сбой диагностического оборудования;
- 5) отказ регистра с плавающей запятой РПЗ;
- 6) сбой буферного регистра связи процессора и каналов РБСП (ТЛН отказ не обнаружили);
- 7) ошибка ОП или ПКЗ — порог повторения команды пройден;
- 8) ошибка ОП или памяти ключей защиты ПКЗ — порог повторения команды не пройден.

Программа SER получает управление в результате прерываний от схем контроля и по ошибкам канала S5 и S6 от программы CCH либо от супервизора ввода-вывода.

Программа SER обеспечивает функциональное восстановление при обработке прерывания типа 2.

При обработке остальных типов прерываний от схем контроля (кроме прерывания типа 1) обеспечивается системное восстановление, системно-обеспечиваемый рестарт или восстановление на уровне ремонта технических средств в зависимости от того, что оказалось поврежденным в результате ошибки: критическая задача, ядро ОС или текущая задача пользователя.

Если в результате ошибки оказались поврежденными критическая задача или ядро операционной системы, обеспечивается

третий уровень восстановления. Если повреждена текущая задача, может быть обеспечено системное восстановление.

При прерываниях типа I возможно функциональное восстановление, когда удается вывести ОК, не связанный в это время ни с одним из каналов ввода-вывода, из состояния ошибки.

Записи об ошибках программой SER строятся во всех случаях прерываний от схем контроля и получения управления по ошибкам канала, кроме ситуаций, приводящих к функциональному восстановлению.

Программа МСН получает управление по прерыванию от схем контроля, от программы ССН (критическая ситуация) или от супервизора ввода-вывода, если произошла ошибка в управлении каналом S5 или интерфейсом S6 и программа ССН в системе отсутствует.

Программа МСН имеет частично-резидентную структуру, т. е. одна часть МСН — ядро — постоянно находится в ОП во время работы системы. Другая часть, состоящая из отдельных модулей, называемых транзитными модулями программы МСН, вызывается по мере необходимости из системного набора данных SYS1.SVCLIB в собственную транзитную область размером 3К байта.

Основными функциями программы МСН являются:

проверка регистров РОН и РПЗ;

анализ информации, помещенной в область записи состояния ЦП с целью определения характера и местонахождения ошибки;

анализ возможности повторения или моделирования программным путем сбойной команды;

повторение или моделирование сбойной команды, а также моделирование прерывания;

проверка ОП и ПКЗ и восстановление в них поврежденной информации;

завершение поврежденных и продолжение неповрежденных задач;

построение записи об ошибке и занесение ее в набор данных SYS1.LOGREC;

выдача сообщений оператору ЭВМ.

При анализе ситуации программа МСН различает микропрограммы процессора: выборки команды, команды ВЫПОЛНИТЬ, команд, отличных от команды ВЫПОЛНИТЬ, микропрограммы ожидания, прерываний, таймера, а также микропрограмму пультовой процедуры ПОВТОРЕНИЕ РССП. Последнее связано с тем, что при останове процессора в режиме «Автомат» (например, при сравнении адресов команд или данных) после повторения РССП продолжается его автоматическая работа, т. е. эта процедура выполняется как команда. Кроме того, микропрограмма этой процедуры используется в автоматическом режиме при работе многомашиного комплекса.

Если ошибка произошла в микропрограмме обработки прерывания от схем контроля или обнаружено логическое несоответствие значений отдельных элементов информации в области записи состояния ЦП, то программа МСН планирует восстановление работоспособности системы на уровне ремонта технических средств. В остальных случаях уровень восстановления и действия программы МСН зависят от типа прерывания.

В результате обработки прерывания типа 2, которое означает успешное повторение сбойной команды аппаратурными средствами, как правило, обеспечивается первый уровень восстановления (функциональное восстановление). Если программа МСН определила, что старое ССП при прерывании от схем контроля неправильно или повторение недопустимо, то обеспечивается третий уровень восстановления (системно обеспечиваемый рестарт).

В результате обработки прерывания типа 5 обеспечивается второй уровень восстановления (системное восстановление) путем завершения прерванной задачи посредством аварийного окончания либо посредством ее перевода в постоянно недиспетчеризуемое состояние.

В результате обработки прерываний типов 3 и 4 возможны все четыре уровня восстановления. При этом функциональное восстановление достигается одним из следующих действий: обнаружением некорректного условия и моделированием соответствующего программного прерывания; моделированием сбойной команды ЦП; моделированием результата выполнения микропрограмм прерываний, ожидания, таймера, ПОВТОРЕНИЕ РССП при ошибках в них.

В результате обработки прерывания типа 6 возможны первый, третий и четвертый уровни восстановления. Функциональное восстановление достигается, если при ошибке РБСП в микропрограмме команд ввода-вывода программе МСН при проверке удалось однозначно определить код условия, являющегося результатом выполнения команды ввода-вывода. Функциональное восстановление также достигается, если при ошибке РБСП в микропрограмме прерываний программе МСН удастся однозначно определить адрес устройства ввода-вывода, прерывание от которого обрабатывалось в момент ошибки.

В результате обработки прерываний типов 7 и 8 возможны все четыре уровня восстановления. В этом случае функциональное восстановление достигается, если одновременно отсутствуют отказы в ОП и ПКЗ, восстановлена вся поврежденная в них информация либо невозстановленная информация сделана недоступной для пользования системой и это не повлияло ни на одну из задач, выполнявшихся в системе, и осуществлено успешное повторение сбойной команды ЦП либо смоделировано программное прерывание после обнаружения некорректного условия.

В результате обработки прерывания типа 1 возможности восстановления зависят от взаимодействия общего канала в момент

ошибки с каналами ввода-вывода. Функциональное восстановление достигается, если в результате выполнения команды ДИАГНОСТИКА общий канал вышел из состояния ошибки и в момент ошибки он не выполнял никаких действий по обеспечению ввода-вывода.

Во всех случаях обработки прерываний программа MCH строит и фиксирует запись об ошибке в SYS1. LOGREC.

Если в операционную систему программа CCH не включена, то по ошибке ввода-вывода управление получает программа MCH. В этом случае MCH формирует запись об ошибке, заносит ее в набор данных SYS1. LOGREC, выдает сообщение оператору и переводит систему в состояние ожидания с соответствующим кодом. Если программа MCH получила управление от программы CCH, то она помещает в SYS1. LOGREC запись, созданную программой CCH, выдает сообщение оператору и переводит систему в состояние ожидания с соответствующим кодом.

14.4. ПРОГРАММЫ РЕДАКТИРОВАНИЯ И ПЕЧАТИ ЗАПИСЕЙ ОБ ОШИБКАХ

Программы редактирования и печати записей об ошибках EREP и SEREP выдают информацию об ошибочной ситуации в отредактированном виде, удобном для анализа.

Эта информация адресована обслуживающему персоналу ЭВМ для проведения целенаправленной профилактики технических средств и повышения эффективности четвертого уровня восстановления — ремонта системы. Для ее использования необходимо знание оборудования ЭВМ ЕС-1033 и назначения содержимого области записи состояния.

Программа SEREP применяется в тех критических случаях, когда программа SER или MCH, если она включена в систему, заканчивает функционирование переводом процессора в состояние ожидания с соответствующим кодом ожидания в адресной части слова состояния программы.

Записи об ошибках, помещенные в набор данных SYS1. LOGREC, могут быть одного из следующих типов:

записями машинных ошибок, сформированными программами SER и MCH;

записями канальных ошибок, сформированными программой CCH;

записями о постоянных и случайных ошибках устройств ввода-вывода, сформированными программой OBR.

В набор данных заносятся записи еще некоторых типов ошибок, которые здесь не рассматриваются.

Программа EREP может редактировать, печатать, а также накапливать в специальном наборе данных, называемом «история», записи об ошибках и обобщать определенные типы записей. Текстовая часть информации об ошибке может выдаваться на

русском и английском языках. Возможность выдачи информации на том или ином языке задается при генерации конкретного варианта.

Зависимость программы EREP от модели определена особенностями записей о программных и канальных ошибках, создаваемых программами SER, MCH и CCH.

SEREP — автономная программа, которая использует сформированный ОС ЕС код типа ошибки (код отказа). Этот код записывается во второй байт зоны кода прерывания в новом ССП для прерываний от схем контроля (байт в ОП с адресом 73₁₆, прил. 10).

Коды отказов и типы распечаток в программе SEREP приведены ниже:

| <i>Код отказа</i> | <i>Содержание распечатки</i> |
|-------------------|---|
| 6F | Отредактированная информация о состоянии центрального процессора |
| 5F | Отредактированная информация о состоянии общего канала |
| 4F | Отредактированная информация о состоянии канала ввода-вывода, не связанного с общим каналом |
| 0F | Отредактированная информация о состоянии канала ввода-вывода, связанного с общим каналом |

Программа редактирует и выводит на печать информацию об ошибочной ситуации, запомненную в ОП в области записи состояния технических средств в момент ошибки. Программа SEREP находится в библиотеке SYS1.SAMPLIB с именем EHPP1033 (для печати текстовой информации на русском языке). Для работы с программой SEREP ее надо вывести на перфокарты, используя операторы управления заданием.

ГЛАВА 15

ТЕСТОВЫЕ ПРОГРАММНЫЕ СРЕДСТВА

15.1. СОСТАВ

В состав программного обеспечения ЭВМ ЕС-1033 входят развитые тестовые программные средства. К ним относятся комплект программ технического обслуживания (КПТО) и комплект автономных тестов устройств (КНТУ), предназначенные для проверки функционирования и локализации неисправностей устройств ЭВМ ЕС-1033 (рис. 166).

Автономная тестовая система КПТО имеет собственную управляющую программу и включает в себя тестовые программы (тест-секции, тесты), ориентированные на проверку функционирования отдельных устройств или групп устройств. Носителем КПТО является магнитная лента или пакет магнитных дисков.

Состав КПО в порядке расположения тестовых программ на магнитном носителе следующий:

- IPL-загрузчик;
- базовый тест;
- управляющая программа;
- диагностический тест селекторных каналов;
- программа измерения времени выполнения команд и производительности ЭВМ;
- сервисная программа;
- обслуживающая программа FOFF;
- обслуживающая программа FOFD;

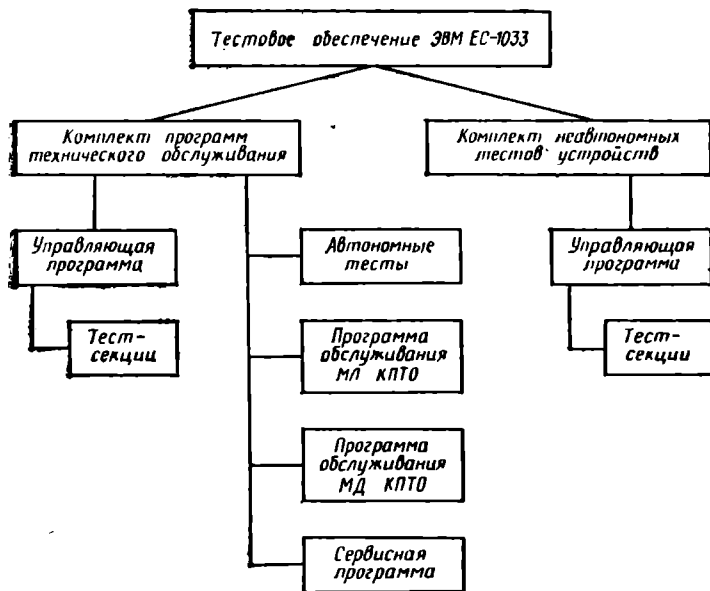


Рис. 166. Структура тестового обеспечения

- тест диагностического оборудования процессора;
- тесты процессора;
- тесты ОП;
- тесты каналов;
- тест прямого управления в режиме одной ЭВМ;
- тесты НМЛ и УУ НМЛ;
- тесты НМД и УУ НМД;
- тест устройства группового управления с выносными пуль-
тами-дисплеями;
- тест устройства ввода-вывода алфавитно-цифровой и графиче-
ской информации;
- тест печатающего устройства;
- тест перфокарточного устройства ввода;
- тест перфокарточного устройства вывода;
- тест пишущей машинки;
- тест перфоленточного устройства ввода;
- тест перфоленточного устройства вывода;
- листинги тест-секций.

Базовый тест, диагностический тест селекторных каналов, программа измерения времени выполнения команд и производительности ЭВМ и обслуживающие программы работают автономно, без участия управляющей программы и так же, как последняя, загружаются для выполнения программой IPL-загрузчиком.

Принцип работы тестов КПТО состоит в том, что в проверяемом устройстве выполняется команда или группа команд с заранее подготовленными исходными данными. Результаты выполнения сравниваются с заранее подготовленными эталонами. Если эти результаты не совпадают с эталонами, оператору выдается сообщение об ошибке. В ряде тестов применяется метод обратного подсчета. В этом случае исходные данные вырабатывает программный генератор псевдослучайных чисел.

По своему назначению тестовые программы КПТО разделяются на проверочные и диагностические. Проверочные тесты определяют правильность функционирования устройств ЭВМ и при обнаружении ошибки сообщают о ней, не локализуя место неисправности. Как правило, такие тесты являются моделезависимыми, и большинство из них входит в состав КПТО различных моделей ЕС ЭВМ. Диагностические тесты предназначены для того, чтобы отыскать место отказа в аппаратуре устройств ЭВМ. Эти тесты используют при своей работе специфичные для каждой модели ЕС ЭВМ диагностические аппаратурные средства, поэтому они являются моделезависимыми. Доступ к диагностическим аппаратурным средствам осуществляется с помощью команды ДИАГНОСТИКА.

В состав КНТУ входят программа выполнения неавтономных тестов устройств OLTER, являющаяся частью операционной системы ОС ЕС, и тесты проверки внешних устройств:

тест НМД;

тест НМЛ;

тест печатающего устройства;

тест перфоленточного устройства ввода;

тест перфокарточного устройства вывода;

тест перфокарточного устройства ввода.

Каждый тест состоит из нескольких программ (тест-секций). Тест-секции КНТУ для ЭВМ ЕС-1033 размещены на одном носителе данных вместе с конкретным вариантом ОС ЕС.

Состав тестового обеспечения ЭВМ ЕС-1033 периодически расширяется за счет включения тестов для новых устройств, а также за счет разработки тестов, повышающих эффективность тестового контроля процессора и каналов ввода-вывода. Эти тесты используют примеры операционной системы и задачи пользователей.

15.2. МОДЕЛЕЗАВИСИМЫЕ ТЕСТЫ

В составе КПТО имеются два моделезависимых диагностических теста: тест диагностического оборудования процессора и диагностический тест селекторных каналов. Эти тесты являются дополнением

развитой системы микропрограммной диагностики процессора и каналов ввода-вывода.

При выполнении процессором программы решаемой задачи средства микропрограммной диагностики включаются в работу при обнаружении ошибки в работе оборудования. Кроме того, они могут быть включены в работу с пульта управления машины или с помощью команды ДИАГНОСТИКА. Последняя используется диагностическими тестами КПО для запуска микропрограммных диагностических процедур с целью проверки работы оборудования и локализации места неисправности.

Тест диагностического оборудования процессора использует команду ДИАГНОСТИКА, выполняемую процессором (п. 10.6). Разряды УСКД [5/15] определяют код диагностического действия. Этот код является адресом микрокоманды, с которой начинается соответствующая диагностическая процедура.

Процесс проверки диагностического оборудования тестом осуществляется путем последовательного выполнения следующих примеров.

Примеры 1—15 организуют выполнение микропрограммных диагностических процедур-тестов локализации неисправности с целью проверки узлов РССП, РАП, РИП, масок, счетчиков циклов, счетчиков РОН, БАЛ, магистралей М1, М2, М3 (п. 10.3). Если при этом не будет обнаружено неисправностей в работе оборудования процессора, происходит прерывание от схем контроля. Программа проверяет состояние триггера неуспеха теста, записанное в области записи состояния процессора. Триггер должен находиться в единичном состоянии, в противном случае программа выдает сообщение о его неправильной установке. Если ТЛН обнаружат неисправность, происходит останов по отказу с одновременной индикацией на пульте управления соответствующей диагностической информации (п. 10.5). Используя эту информацию, с помощью специального справочника можно определить конструктивный адрес неисправности.

Примером 16 осуществляется проверка правильности функционирования аппарата записи состояния процессора в фиксированную область ОП (пп. 10.1, 10.3). Для этого РОН, РПЗ, РОП и некоторые другие регистры программно устанавливаются в определенное состояние, после чего выполняется микропрограмма записи состояния (п. 10.2). Содержимое области записи состояния в ОП сравнивается с эталоном, и в случае их несоответствия выдается сообщение об этом.

Примерами 17, 18, 20, 21 проверяется функционирование диагностического оборудования ЦП, реагирующего на ошибки процессора и оперативной памяти при различном состоянии триггера порога повторения команды (п. 10.1). Ошибки процессора и оперативной памяти имитируются с помощью команды ДИАГНОСТИКА. Для этого используется специальный диагностический режим работы магистрального коммутатора ЦП — «Блокировка

контрольного разряда», который принудительно устанавливает в единичное состояние контрольные разряды всех трех информационных магистралей. Таким образом, с помощью этого режима и подбора значения считанной на магистраль константы можно получить информацию с неправильным дополнением и записать ее в РОН, РПЗ и другие регистры ЦП, а также в ОП по адресу 1В8. После этого выполняются определенные команды системы и проверяется факт возникновения ошибки и правильность реакции на нее системы диагностики. В частности, проверяется реакция системы диагностики на ошибки ЦП и ОП, возникшие до и после порога повторения команды.

Примером 19 проверяется установка и сброс режима «Контрольный код» оперативной памяти (п. 1.10). При установке этого режима во время записи состояния ОП и ЦП в фиксированную область ОП информация с неправильной четностью, поступившая из ЦП в ОП, записывается в ОП с правильной четностью. Ошибка РИ ОП в этом случае не формируется.

Примерами 22, 23 проверяется выполнение установки и сброса режима «Блокировка» РМО (п. 10.1) при чтении слова ОП по адресу 1В8 и РОН, в которых находится информация с неправильной четностью. Если этот режим установлен, то ошибки в РМО не фиксируются и прерывание от схем контроля отсутствует.

Примерами 24, 25 проверяется функционирование диагностического оборудования при ошибках РССП и РПЗ. Информация с неправильной четностью загружается в эти регистры заранее командой ДИАГНОСТИКА.

Примером 26 производится проверка функционирования диагностического оборудования при ошибке общего канала (пп. 10.3, 13.3, 13.5). С этой целью организуется чтение БОП6 по адресу, в котором содержится информация с неправильной четностью. После прерывания от схем контроля программа анализирует содержимое области записи состояния общего канала и сравнивает его значение с эталоном.

Диагностический тест селекторных каналов предназначен для проверки и локализации неисправностей оборудования СК, которое не имеет микропрограммного управления. Это оборудование осуществляет передачу данных и выборку управляющих слов в СК и работает под управлением тактовой цепочки. Тестирование СК заключается в выполнении определенного количества тактов под управлением тактовой цепочки СК, завершающихся останом с последующей записью состояния, и сравнении содержимого области записи состояния СК с эталоном. При несравнении программа определяет неисправный узел (триггер, разряд регистра) и выводит на печать конструктивный адрес неисправности. Запуск тактовой цепочки СК осуществляется с помощью диагностических микропрограмм, имитирующих микроприказы общего канала НАЧАТЬ ВВОД-ВЫВОД, ПРОДОЛЖИТЬ, МОДИФИКАЦИЯ,

ИНФ-А, РПД и др. Для этого используется команда ДИАГНОСТИКА, выполняемая каналами (п. 10.6, 13.5).

Диагностический тест СК работает автономно от управляющей программы КПТО и состоит из собственной управляющей программы и примеров. Управляющая программа осуществляет запуск примеров и сравнивает содержимое области записи состояния СК с эталоном. В тесте имеется таблица конструктивных адресов неисправностей, каждая запись в которой поставлена в соответствие с определенным разрядом области записи состояния СК, отражающим состояние его узлов (триггеров, разрядов регистров). При несравнении, по номеру несовпавшего разряда, на печать выводится соответствующая запись из таблицы конструктивных адресов неисправностей.

Кроме того, управляющая программа может работать в режиме генерации. Генерация заключается в подготовке эталонов состояния СК в диагностических примерах и формировании теста с эталонами на магнитной ленте. Генерация необходима при аппаратурных изменениях в СК, которые влекут за собой изменения режима его работы и структуры области записи состояния СК.

Назначение примеров теста следующее:

- 1 — проверяет реакцию СК на сигнал СБРОС В КАНАЛЕ;
- 2—4 — проверяют работу тактовой цепочки модификации;
- 5—8 — проверяют работу схемы выборки УСК с кодом команды ЗАПИСЬ;
- 9—11 — проверяют схемы выборки по цепочке команд;
- 15—17 — проверяют реакцию СК на неверный формат АСК;
- 18—23 — проверяют реакцию СК на ошибки в УСК;
- 24—26 — проверяют схемы обмена данными в ОП;
- 27—28 — проверяют схему предварительной выборки УСК;
- 30 — проверяет выборку УСК с кодом команды ЧТЕНИЕ;
- 31, 34, 35 — проверяют схему обмена данными при операции типа ЗАПИСЬ;
- 36—38, 47—49 — проверяют установку ряда служебных триггеров СК;
- 39, 42—44 — проверяют работу схемы обмена данными при операции типа ЧТЕНИЕ;
- 50—53 — проверяют схему обмена данными при операции типа ОБРАТНОЕ ЧТЕНИЕ.

Номер проверяемого тестом селекторного канала задается оператором, который устанавливает его с помощью клавиш «АДРЕС КАНАЛА» (п. 3.6, 13.1). Структура теста предусматривает зацикливание примера, обнаружившего неисправность, с запретом выдачи диагностического сообщения.

15.3. МОДЕЛЕОРИЕНТИРОВАННЫЕ ТЕСТЫ

Основным принципом при создании моделиориентированных тестов явился учет особенностей аппаратуры и микропрограммной реализации команд в модели ЕС-1033. В числе тест-секций КПТО

имеется пять моделиориентированных. С их помощью проверяется правильность выполнения команд десятичной арифметики, арифметики с плавающей запятой, логических команд и команд перехода. Моделиориентированность этих тест-секций заключается в том, что исходные данные (операнды) подобраны так, что при выполнении проверяемой команды активизируются все микрокоманды, входящие в микропрограммы этих команд, тогда как немоделиориентированными тест-секциями КПО активизируются лишь 50—70% микрокоманд. Для подбора операндов строился граф микропрограммы команды и определялись условия, при которых происходили ветвления микропрограммы. Затем подготавливались первичные исходные операнды и производилась их обработка аналогично тому, как она происходит в микропрограмме команды. В точках ветвления микропрограммы определялась неактивизированная ранее ветвь микропрограммы и, если было необходимо, производилась корректировка операндов (или их размещение в ОП) для удовлетворения условий ветвления. На графе микропрограммы помечалась трасса прохождения примера.

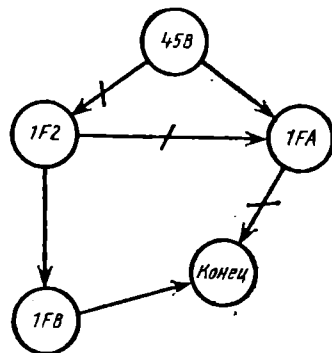


Рис. 167. Фрагмент графа микропрограммы команды СРАВНЕНИЕ ДЕСЯТИЧНОЕ (стадия завершения)

Простейший фрагмент графа микропрограммы представлен на рис. 167. В кружках отмечены адреса микрокоманд, на которые происходит ветвление. Ориентированные линии (ребра) — линейные участки микропрограммы. Представленный на рисунке участок микропрограммы активизируется в случае наличия разных знаков операндов. Очевидно, что для активизации всех микрокоманд участка необходимо подготовить три примера. Помеченные чертой ребра — активизированные примерами участки ранее использовавшихся тестов. В основном графы микропрограммы команд являются довольно сложными и процесс подготовки операндов трудоемок.

В тест-секции, проверяющей команды десятичной арифметики, предусматривается распечатка трассы прохождения примера по микропрограмме (перечень адресов микрокоманд в точках ветвления) в случае неверного выполнения проверяемой команды. Печать трассы происходит при установке оператором специального ключа тест-секции. Такое информационное сообщение об ошибке значительно сокращает процесс нахождения места неисправности.

Применение принципа охвата проверками всех графов микропрограмм позволяет проверить чтение содержимого ячеек БОП, проверить полностью блок формирования адресов БОП, правильность выполнения всех микроприказов проверяемой команды.

15.4. УПРАВЛЯЮЩАЯ И ОБСЛУЖИВАЮЩАЯ ПРОГРАММЫ

Управляющая программа (тест-монитор ТМ) предназначена для управления загрузкой и выполнением тест-секций. Основными ее функциями являются прием и выполнение директив оператора, загрузка тестов в ОП, печать сообщений, назначение тест-секциям тестируемых устройств и обработка прерываний. В соответствии с этим ТМ состоит из пяти логических компонент и программы инициализации, которая выполняется после загрузки ТМ в ОП. Последняя генерирует ТМ, ориентированный на конкретную конфигурацию системы, и при дальнейшей работе не запрашивается.

Программа FOFF предназначена для создания и обслуживания КПТО на магнитной ленте. Она позволяет генерировать МЛ с перфокарт, добавлять, замещать и исключать программы, вносить в них изменения, дублировать ленты, а также выводить программы на перфокарты и распечатывать их.

Обслуживающая программа и программа FOFF являются стандартными и входят в состав КПТО большинства моделей ЕС ЭВМ.

Программа FOFD предназначена для размещения КПТО ЕС-1033 на пакете магнитных дисков. Работа с дисковым вариантом КПТО позволяет значительно сократить время вызова тест-секции на выполнение (в среднем с 65,5 до 0,1 с). Программа состоит из подпрограммы определения и назначения устройств ввода-вывода и подпрограммы обработки директив оператора. Для работы программы необходимы следующие устройства:

пишущая машинка для ввода управляющей информации (директив) и вывода сообщений об ошибках;

НМЛ для ввода данных с тестовой магнитной ленты КПТО;

печатающее устройство для вывода информационных сообщений;

НМД ЕС-5056 или ЕС-5061 для создания тестового пакета магнитных дисков.

После загрузки программы начинает работать подпрограмма определения устройств, для которой оператор должен назначить устройства программы FOFD. Для назначения устройства необходимо ввести его тип и адрес. При этом используются следующие обозначения типов устройств:

У — для пишущей машинки;

Р — для печатающего устройства;

Л — НМЛ;

Д — НМД ЕС-5056;

Ф — НМД ЕС-5061.

После назначения устройств ввода-вывода оператор с пишущей машинки может задать директивы, обеспечивающие перепись КПТО с магнитной ленты на пакет магнитных дисков. Сначала оператор должен задать директиву FORMAT для инициализации пакета магнитных дисков. После окончания инициализации необходимо задать директиву GEN. После окончания переписи для

завершения формирования содержимого пакета магнитных дисков необходимо задать директиву TERM.

Включенная в состав КПТО ЕС-1033 сервисная программа позволяет пользователю получить распечатки исходных кодов (листинги) тест-секций, входящих в состав КПТО. Листинги находятся на МЛ КПТО, поставляемой пользователю.

После загрузки сервисной программы можно вывести листинг тест-секций на печатающее устройство указанием директивы LIST. При этом можно распечатывать листинги любой выбранной тест-секции, нескольких тест-секций или всех подряд.

Дополнительными функциями программы являются перфорация исходных модулей тест-секций;

добавление листингов на МЛ;

замена листингов на МЛ;

дублирование МЛ КПТО с листингами тест-секций.

15.5. КОМПЛЕКТ НЕАВТОНОМНЫХ ТЕСТОВ УСТРОЙСТВ

Комплект КНТУ предназначен для профилактической проверки устройств ввода-вывода под управлением операционной системы. Обеспечение неавтономной проверки было достигнуто в ОС ЕС, начиная с редакции 4.1. Тест-секции КНТУ для ЭВМ ЕС-1033 размещены на одном носителе данных вместе с конкретным вариантом ОС ЕС, и поэтому организация проверки устройств с помощью КНТУ на ЕС-1033 отлична от организации проверки устройств на других моделях ЕС ЭВМ.

Для организации проверки устройств необходимо выполнить следующие действия (соответствующие операторы заданий поставляются на перфокартах):

после загрузки операционной системы разметить рабочую МЛ с помощью утилиты IENINIT;

скопировать наборы данных КНТУ с системного тома на размеченную утилитой IEBGENER магнитную ленту. Полученную МЛ с тестами КНТУ можно использовать для создания библиотеки во вновь создаваемом конкретном варианте;

удалить скопированные наборы данных с системного тома утилитой IENPROGM;

создать библиотеку КНТУ (редактор — IEWL) на системном или рабочем томе;

включить с помощью утилиты IEBUPDTE специальную процедуру библиотеки КНТУ, облегчающую вызов управляющей программы OLTER. Вызов OLTER осуществляется вводом приказа S OLTER. PN где N — номер раздела памяти.

Каждое устройство перед проверкой необходимо перевести в автономное состояние. Для этого вводится приказ

V XXX, OFFLINE,

где XXX — адрес устройства.

По завершению проверки устройство переводится в состояние *on — line*.

Приказ на вызов и исполнение тест-секций включает в себя тип устройства, номера тест-секций и средства тестирования. Например, для проверки накопителей на магнитной ленте ЕС-5050 необходимо ввести приказ

R id, 'XXX/T5010A—H/PP(3)'

где XXX — адрес устройства.

Секции для НМЛ имеют номера от А до Н. Параметр средств тестирования РР (3) разрешает печатать на пульте оператора полные информационные сообщения тест-секций. Возможно применение такого средства, как приостанов процесса проверки при обнаружении ошибки. В этом случае после выдачи диагностического сообщения программа OLTEP переходит в состояние готовности повторить или продолжить выполнявшийся тест, или запустит другие тесты.

15.6. ИСПОЛЬЗОВАНИЕ ТЕСТОВЫХ ПРОГРАММНЫХ СРЕДСТВ ПРИ НАЛАДОЧНЫХ РАБОТАХ

На начальном этапе наладки машины могут успешно использоваться микропрограммные диагностические тесты (пп. 10.3, 10.5, 13.4), которые хранятся в управляющей памяти процессора и каналов ввода-вывода. Вызов теста, его фрагмента или последовательности тестов производится с пульта управления машины по начальному адресу теста или его фрагмента. Для наладки оператор может использовать различные режимы работы, предоставляемые пультами средствами процессора и каналов.

Проверка функционирования машины с помощью автономных тестов является следующим этапом наладки. В состав автономных тестов входит базовый тест, а также программа MIX измерения времени выполнения команд и измерения производительности ЭВМ.

Первым из программных тестов используется базовый тест. Он обеспечивает проверку ядра, включающего в себя часть аппаратуры процессора, каналов и оперативной памяти, которые используются при выполнении ограниченной части команд процессора (так называемых базовых команд). Проверяемый объем ядра позволяет в дальнейшем успешно работать управляющей программе КПО.

В состав программного обеспечения базовый тест включен дополнительно на перфокартах. Это связано с тем, что проверка функционирования мультиплексного канала при выполнении первоначальной загрузки представляет собой выполнение цепочки данных с чтением для перфокарточного устройства ввода. Процедура первоначальной загрузки не является сложной для ЭВМ ЕС-1033 и отлаживается в пультовом режиме. Таким образом,

базовый тест может быть загружен с перфокарточного устройства процедурой первоначальной загрузки мультиплексного канала. В процессе загрузки этого теста ни одна команда процессора не выполняется: работает так называемый канальный загрузчик.

Успешное выполнение базового теста позволяет в дальнейшем работать управляющей программе КПТО. При ее работе используется базовый набор команд процессора, чтение магнитных лент через селекторный канал и ряд команд устройства управления магнитной лентой.

После базового теста запускается диагностический тест селекторных каналов, описанный выше, а затем — тест-секции процессора и каналов, работающих под управлением управляющей программы КПТО.

Проверка функционирования ЭВМ ЕС-1033 на КПТО и КНТУ в процессоре проведения наладочных работ и приемо-сдаточных испытаний осуществляется согласно соответствующим инструкциям. Их разработка позволила создать технологический процесс наладки ЭВМ, а также обеспечить возможность использовать при этом операторов средней квалификации. Вместе с тем в КПТО и КНТУ предусмотрено достаточное количество управляющих команд, которые дают возможность квалифицированному персоналу осуществлять поиск и локализацию неисправностей в режиме диалога.

После завершения наладки машины ее производительность и время выполнения команд определяются в помощью программы MIX.

ГЛАВА 16

ПРОГРАММЫ, РАСШИРЯЮЩИЕ ОПЕРАЦИОННУЮ СИСТЕМУ

16.1. ПОДСИСТЕМА ПЛАНИРОВАНИЯ КРОС

Подсистема планирования КРОС (РОС — расширение ОС, К — индекс разработчика) совместно с ОС ЕС является программным обеспечением однопользовательской вычислительной системы. Она предназначена для использования на средних и больших ЭВМ, минимальный состав которых должен содержать оперативную память объемом не меньше, чем 256К байт, не менее двух накопителей на магнитных дисках, одно перфокарточное устройство ввода, одно печатающее устройство и пультовую пишущую машинку для связи с оператором. В состав подсистемы КРОС входит эксплуатационная документация и дистрибутивная лента,

которая содержит все исходные модули, а также задания, необходимые для генерации и включения подсистемы планирования КРОС в операционную систему.

Подсистема планирования КРОС по сравнению с ОС ЕС использует более оптимальные алгоритмы планирования. Она содержит в своем составе средства, которые обеспечивают заметное повышение производительности вычислительной системы, значительно облегчают работу оператора, расширяют возможности операционной системы и повышают надежность функционирования вычислительной системы в целом.

Повышение производительности вычислительной системы достигается благодаря использованию подсистемой планирования КРОС динамического приоритетного планирования. В операционной системе выбор тех или иных характеристик заданий в качестве определяющих при планировании, а также установку правила (алгоритма), согласно которому задания направляются в более приоритетный раздел, осуществляет человек — программист, руководитель вычислительного центра и т. д. Более того, в ОС ЕС приоритет заданий одного класса зачастую находится в тесной зависимости от того, в какой последовательности они введены в систему оператором, который в своих действиях не всегда руководствуется заботой об эффективном использовании вычислительной системы. Таким образом, функции планирования в ОС ЕС почти полностью возлагаются на человека, а сама операционная система в смысле планирования выполняет весьма примитивные функции. В условиях крупных вычислительных центров с большим количеством пользователей этот недостаток планирования в ОС ЕС создает большие трудности и не позволяет обеспечить высокий уровень производительности системы.

В отличие от ОС ЕС подсистема планирования КРОС при выборе заданий для выполнения сама учитывает такие характеристики задания, как время выполнения задания и количество строк информации, выдаваемой на печать. Эти характеристики указываются программистом либо назначаются подсистемой планирования КРОС по умолчанию. Зависимость между характеристиками заданий и порядком их выбора для выполнения определяется системным программистом. Задания, выбранные для выполнения, передаются программе системного ввода ОС ЕС подсистемой планирования КРОС, упорядоченные согласно этим характеристикам.

Характеристики задания могут существенно измениться за время его выполнения. Подсистема КРОС в течение некоторого, заранее заданного, промежутка времени собирает информацию о каждой выполняющейся задаче. На основе этой информации вычисляются характеристики использования каждой задачей времени центрального процессора. Преимущество отдается тем задачам, которые мало используют его время работы. Задачи, загружающие центральный процессор, служат фоном для тех

задач, которые выполняют в основном операции ввода-вывода. Таким образом повышается загрузка вычислительной системы. В очередь на вывод задача поступает с приоритетом, который назначает для нее КРОС в зависимости от реального числа строк, созданных задачами на стадии ее выполнения. Одна и та же задача на разных стадиях своего выполнения может иметь разный приоритет. Такой подход улучшает планирование, повышает производительность вычислительной системы.

На повышение производительности также положительно влияет использование подсистемой планирования КРОС цепочек команд в канальных программах, которые позволяют заметно увеличить скорость ввода-вывода и использование улучшенных алгоритмов распределения внешней памяти. В операционной системе для каждого набора данных входного потока создается последовательный набор на устройстве прямого доступа. Отдельные последовательные наборы создаются также для наборов данных, предназначенных для печати и перфорации на устройствах системного вывода. Чтобы разместить, заполнить и удалить эти наборы данных, требуются не только значительное время центрального процессора, но и нетривиальная работа и посредничество оператора. В отличие от ОС ЕС подсистема планирования КРОС буферизирует входные и выходные данные в одном наборе, расположенном на одном или нескольких томах прямого доступа. Память в этом наборе данных выделяется динамически по запросам, что приводит к уменьшению потребности во внешней памяти. Единицей распределения является группа дорожек, число которых определяется системным программистом. При распределении внешней памяти учитывается положение механизма доступа. Информация о свободных и занятых группах дорожек хранится в таблицах, находящихся в основной памяти, а не на устройствах прямого доступа.

Уменьшение нагрузки на оператора достигается благодаря тому, что подсистема планирования КРОС берет на себя часть его функций. В операционной системе каждому разделу основной памяти может быть приписано не более трех классов заданий. Поэтому при наличии во входном потоке заданий, у которых классы не совпадают с теми классами, которые приписаны разделам, оператору приходится пользоваться дополнительными командами для изменения классов разделов или классов заданий. В подсистеме планирования КРОС для описания разделов основной памяти вводится понятие логических разделов. Каждому такому разделу может быть приписано до восьми классов, с которыми задания могут выполняться в этом логическом разделе. Управлять логическими разделами значительно проще, чем физическими, так как при этом не нужно учитывать особенности команды оператора, используемой для определения характеристик физических разделов в ОС ЕС. Подсистема планирования КРОС автоматически запускает программы системного ввода и вывода, а также инициаторы, освобождая от выполнения этих функций оператора ЭВМ.

В операционной системе оператор имеет возможность управлять либо отдельными, либо всеми заданиями. В подсистеме планирования КРОС оператору, кроме этого, дополнительно предоставляется возможность управлять группами заданий. Значительным удобством является и то, что для управления как отдельными заданиями, так и группами оператор в КРОС использует номера заданий, присваиваемые подсистемой планирования КРОС, а не их имена. Следует отметить, что кроме имеющейся в ОС ЕС возможности задержать выборку отдельных или всех заданий оператору предоставляется возможность задерживать выборку заданий, введенных с определенного периферийного устройства.

Очень удобным средством, предоставляемым оператору, является возможность автоматического запуска перфокарточного устройства ввода. При наличии этой возможности программа, обеспечивающая ввод с перфокарточного устройства, не прекращает работу с этим устройством по условию КОНЕЦ НАБОРА ДАННЫХ. После приведения оператором перфокарточного устройства ввода в состояние готовности эта программа начинает ввод входного потока с этого устройства.

Оператору предоставляется ряд дополнительных команд для управления устройствами ввода-вывода. Например, в случае повреждения бумаги при печати набора данных оператор может возобновить печать с первой страницы набора данных или просто вернуться на несколько страниц назад, чтобы повторить печать потерянной информации.

Подсистема планирования КРОС предоставляет оператору большую свободу при вводе текста команды по сравнению с ОС ЕС. Пробелы можно вставлять в любом месте команды после специального символа, идентифицирующего команду оператора КРОС. Допускаются ошибки в ключевых параметрах команд оператора. Если ошибочный ключевой параметр содержит достаточно информации для идентификации нужного задания или средства КРОС, то он будет рассмотрен как правильный.

Расширение возможностей ОС ЕС заключается в том, что КРОС обеспечивает передачу заданий из любой выполняющейся в операционной системе задачи во входной поток заданий КРОС, изменение обычного порядка планирования заданий из входного потока и передачи этих заданий непосредственно обрабатывающей программе, минуя планировщик ОС ЕС; задержку заданий до установки всех необходимых для них томов прямого доступа и томов магнитной ленты, а также включение в распечатку задания части протокола оператора, относящегося к выполнению данного задания.

Повышение надежности функционирования вычислительной системы достигается тем, что подсистема планирования КРОС сохраняет все системные входные и выходные наборы данных и управляющие таблицы задания до его полного завершения. В случае повторного пуска подсистемы КРОС, необходимость

в котором возникает в случае прекращения функционирования вычислительной системы (например, из-за сбоев оборудования), задание повторяется либо со стадии выполнения без повторного ввода его в систему, либо с той страницы, во время печати которой оно было прервано.

В качестве иллюстрации эффективности средств, которыми располагает подсистема планирования КРОС, можно привести следующий пример.

В вычислительном центре, обслуживающем разработчиков программного обеспечения, выполнялся пакет заданий под совместным управлением ОС ЕС и подсистемы планирования КРОС. За каждый час объем системного ввода-вывода составлял в среднем 1300 записей из входного потока на магнитной ленте, 12000 строк на печатающем устройстве и 70 перфокарт на перфокарточном устройстве вывода. В результате выполнения пакета заданий было установлено, что пропускная способность вычислительной системы благодаря подсистеме планирования КРОС увеличивается примерно на 30%.

Более значительные результаты в повышении производительности вычислительной системы, очевидно, могут быть достигнуты в ВЦ, ориентированных на выполнение заданий с большим объемом системного ввода-вывода.

16.2. ОБРАБОТКА ЗАДАНИЙ ПОД УПРАВЛЕНИЕМ КРОС

Чтобы использовать подсистему планирования КРОС, необходимо провести специальную генерацию ОС ЕС и генерацию собственно подсистемы планирования КРОС, в результате чего конкретный вариант КРОС становится включенным в состав ОС ЕС.

Подсистема планирования КРОС запускается оператором с помощью каталогизированной процедуры и выполняется как обрабатывающая программа в разделе основной памяти. После запуска КРОС управление получает программа инициализации КРОС. Эта программа обращается к SVC-программе КРОС для перевода ее в состояние СУПЕРВИЗОР с нулевым ключом защиты, подготавливает управляющие блоки и таблицы, необходимые для работы КРОС; изменяет точки входа в некоторые программы ОС ЕС; запускает программу системного ввода, программу системного вывода и инициаторы.

После завершения процесса инициализации подсистема планирования КРОС приступает к обработке заданий. При этом каждое задание проходит четыре основные стадии: ввода, выполнения, вывода и завершения (рис. 168). На первой стадии осуществляется ввод входных потоков заданий с перфокарточных устройств ввода и накопителей на магнитной ленте, число которых может быть неограниченным. Операторы языка управления заданиями и входные данные запоминаются во внешней памяти для последую-

щей обработки. Строится резидентная очередь заданий. Более подробная информация о заданиях запоминается в виде таблиц во внешней памяти. Все эти действия в подсистеме планирования КРОС выполняет программа ввода. Она является реентерабельной, поэтому одна копия этой программы может обслужить неограниченное число входных потоков. Чтобы устройства ввода-вывода, предназначенные для ввода заданий в КРОС, не были распределены ОС ЕС другим заданиям, КРОС переводит их в автономное состояние. Такой способ резервирования устройств ввода-вывода

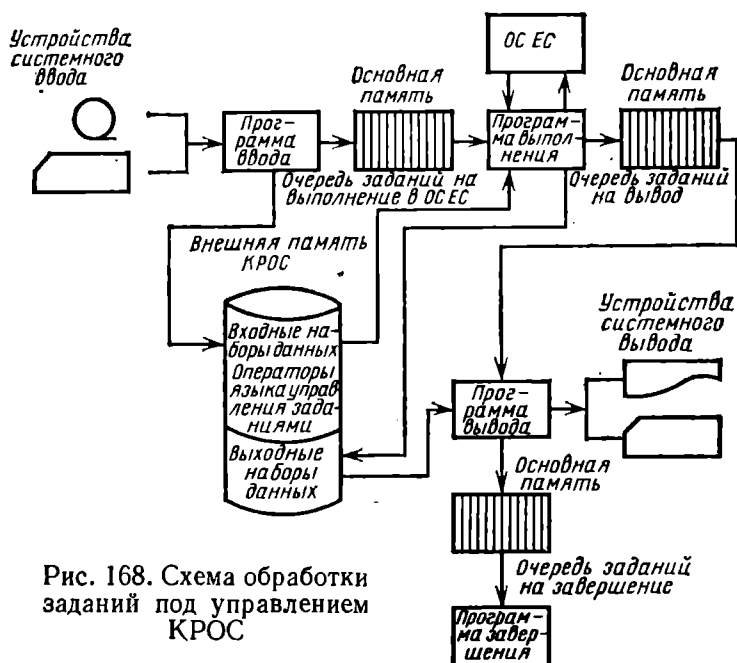


Рис. 168. Схема обработки заданий под управлением КРОС

(а не с помощью операторов DD в задании для вызова КРОС) позволяет, по мере необходимости, выделять эти устройства для других заданий.

На стадии выполнения задания выбираются из очереди, построенной на стадии ввода, и передаются для выполнения ОС ЕС. Порядок выбора заданий для выполнения определяется классом задания, а среди заданий одного класса — приоритетом. Приоритет задания указывается программистом явно или косвенно. Явное указание приоритета осуществляется включением в задание специального управляющего оператора. Косвенно приоритет планирования указывается программистом в поле учетной информации оператора JOB подпараметрами ожидаемого времени выполнения и ожидаемого числа строк печати.

Выбранное задание передается программе системного ввода ОС ЕС, запущенной программой инициализации КРОС. При запуске этой программы в качестве устройства системного ввода назначается псевдоустройство. При всех обращениях к этому

псевдоустройству подсистема планирования КРОС перехватывает эти обращения и передает задания, помещенные программой ввода во внешнюю память, программе ввода ОС ЕС. После обработки каждого оператора языка управления заданиями управление передается программе КРОС, которая в операторах DD заменяет параметры *, DATA и SYSOUT на параметр UNIT = XXX, где XXX — адрес псевдоустройства. Вследствие такой замены программа системного ввода будет обрабатывать только операторы языка управления заданиями, считая, что входные наборы данных расположены на псевдоустройствах. Благодаря этому сокращается время работы программы системного ввода. Следует отметить, что задание передается программе системного ввода только в том случае, если есть инициатор, способный инициировать данное задание. Это дает возможность существенно уменьшить размер набора данных SYS1.SYSJOBQE, используемого планировщиком заданий ОС ЕС для построения очереди заданий.

При выполнении задания подсистема планирования КРОС перехватывает все обращения к входным и выходным наборам данных и моделирует все запросы на операции обмена, обеспечивая ввод входных наборов данных из внешней памяти и вывод выходных наборов данных во внешнюю память. Заканчивается стадия выполнения извлечением блоков системных сообщений из набора данных SYS1.SYSJOBQE и добавлением их к выходным наборам данных во внешней памяти.

Всеми указанными выше действиями в подсистеме планирования КРОС управляет программа выполнения. Эта программа является реентерабельной, что позволяет ей одновременно управлять выполнением нескольких заданий. После завершения стадии выполнения задание помещается в очередь на вывод выходных наборов данных.

На стадии вывода выходные наборы данных, запомненные во внешней памяти на стадии выполнения, выводятся на печатающие и перфокарточные устройства, число которых может быть неограниченным. Порядок обработки задания на стадии вывода определяется его приоритетом в очереди заданий, ожидающих вывода выходных наборов данных. Этот приоритет либо сохраняется от предыдущей стадии, либо назначается вновь. Если приоритет задания был назначен с помощью специального управляющего оператора, то он сохраняется. Если такой управляющий оператор в задании отсутствовал, приоритет вычисляется на основе числа строк информации, созданных для печати заданием на стадии выполнения. Зависимость приоритета от числа строк информации, подготовленной для печати, задается во время генерации КРОС. На стадии вывода задание выполняется под управлением программы вывода КРОС. Она так же, как и программа выполнения, является реентерабельной, и поэтому одна ее копия может обслужить неограниченное число выходных потоков данных.

Последней при обработке задания является стадия завершения, когда соответствующая программа освобождает все ресурсы, выделенные заданию для выполнения в подсистеме планирования КРОС.

16.3. ПОДСИСТЕМА ПЛАНИРОВАНИЯ РОС

Подсистема планирования РОС относится к пакетам прикладных программ, расширяющим возможности ОС ЕС. Она позволяет объединить несколько ЭВМ Единой системы в многомашинную вычислительную систему для обработки общего потока заданий. Одна из ЭВМ этой системы, на которой функционирует подсистема планирования РОС, выполняет ввод заданий, управление их обработкой, печать и перфорацию выходных наборов данных. Эта ЭВМ называется «обслуживающей». Остальные ЭВМ вычислительной системы называются «основными» и предназначены только для выполнения заданий. Физическое объединение машин производится через адаптеры канал-канал.

Преимущества организации такой вычислительной системы заключаются в следующем.

Во-первых, обеспечиваются более высокая организация и оперативность обслуживания вычислительного процесса вследствие того, что оператор после объединения ЭВМ имеет дело только с одной машиной. Например, при работе с подсистемой планирования РОС отсутствует необходимость формирования отдельных пакетов заданий для каждой ЭВМ системы. Подсистема планирования РОС берет на себя функции диспетчера и выполняет эти функции с учетом реальной обстановки, сложившейся на каждой отдельной ЭВМ.

Во-вторых, улучшается использование оборудования вычислительной системы. Печатающие устройства, перфокарточные устройства ввода и вывода используются значительно интенсивнее благодаря наличию большого количества заданий на входе вычислительной системы и на стадиях печати и перфорации. Кроме того, центральные процессоры «основных» ЭВМ освобождены от выполнения таких функций, как печать и перфорация.

В-третьих, уменьшается количество взаимных помех заданий, возникающих в результате недостаточности ресурсов. Перед передачей задания на «основную» ЭВМ подсистема планирования РОС проверяет наличие на этой ЭВМ ресурсов, необходимых для его выполнения. Если ЭВМ не имеет достаточных ресурсов, то будет рассматриваться возможность выполнения задания на других ЭВМ.

В-четвертых, увеличивается важный для обрабатывающих программ ресурс — время использования центрального процессора. Выполнение обрабатывающих программ не прерывается программами операционной системы, выполняющими печать и перфорацию. Сокращается время работы программ операционной

системы, обеспечивающих выполнение заданий в системе. Например, планировщик ОС ЕС на «основной ЭВМ» работает в короткой очереди заданий. Обычно первое же задание в этой очереди запускается планировщиком ОС ЕС, так как доступность ресурсов, необходимых для выполнения задания, была проверена подсистемой планирования РОС до передачи задания на «основную» ЭВМ.

И, наконец, увеличивается количество накопителей на магнитных дисках Основных ЭВМ, которые могут использоваться обрабатываемыми программами. Операционная система, функционирующая без РОС, создает временные наборы данных на томах прямого доступа для размещения данных, поступающих из входных и выходных потоков системы. При использовании РОС буферизация входных и выходных наборов данных происходит на накопителях на магнитных дисках «обслуживающей» ЭВМ, которые образуют внешнюю память РОС. Накопители на магнитных дисках «основной» ЭВМ могут использоваться для установки томов с операционной системой и томов пользователей.

Подсистема планирования РОС обеспечивает функционирование вычислительной системы, в состав которой могут входить от одной до четырех вычислительных машин ЕС ЭВМ. В качестве «основных» ЭВМ рекомендуется использовать ЭВМ средней или большой мощности. Выбор ЭВМ для выполнения функций обслуживания зависит от количества «основных» ЭВМ, входящих в состав вычислительной системы. В вычислительной системе с одной или двумя «основными» ЭВМ в качестве «обслуживающей» можно использовать ЭВМ средней мощности. Для системы с тремя или четырьмя «основными» ЭВМ рекомендуются модели большой вычислительной мощности.

Размещение периферийного оборудования в вычислительной системе под управлением РОС асимметричное. Печатающие устройства, перфокарточные устройства ввода и вывода, пишущие машинки, алфавитно-цифровые дисплеи присоединяются к мультиплексному каналу «обслуживающей» ЭВМ. В состав периферийных устройств «основных» ЭВМ чаще всего входят НМД и НМЛ. Связь каждой «основной» ЭВМ с «обслуживающей» осуществляется через адаптер канал-канал. На рис. 169 приводится типичная конфигурация вычислительной системы, работающей с подсистемой планирования РОС.

В том случае, когда вычислительная система состоит из одной ЭВМ, подсистема планирования РОС обеспечивает поочередно выполнение машиной обслуживающих и основных функций. Считается, что в этом случае «основная» ЭВМ организована на «обслуживающей» и называется «местной основной». Связь между «местной основной» ЭВМ и «обслуживающей» осуществляется через программно-моделируемый адаптер канал-канал. Создание «местной основной» ЭВМ позволяет выполнять задания на «обслуживающей» ЭВМ, что может оказаться весьма желательным в случае, когда

ресурсы «обслуживающей» ЭВМ существенно превосходят ресурсы, необходимые для работы РОС.

Возможность создания «местной основной» ЭВМ позволяет, например, иметь две «основные» ЭВМ в вычислительной системе, состоящей из двух ЭВМ.

Любая из «основных» ЭВМ работает под управлением ОС ЕС. Допускается функционирование ОС ЕС в любом из трех режимов: РСР, МFT и MVT. Адаптер канал-канал на «основной» ЭВМ

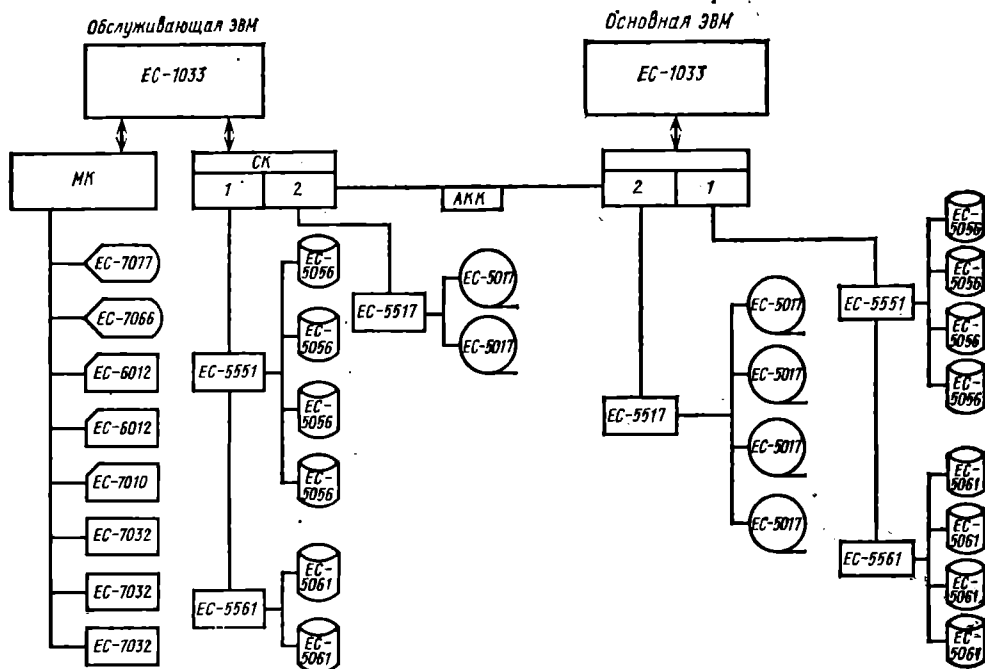


Рис. 169. Конфигурация оборудования РОС

используется в качестве консоли операционной системы, системного устройства ввода и системного устройства вывода. Адаптер канал-канал может также выделяться заданиям, выполняющимся на «основной» ЭВМ.

На «обслуживающей» ЭВМ также функционирует ОС ЕС в любом из трех режимов. Если в вычислительной системе организуется местная «основная» ЭВМ, то на «обслуживающей» машине должна функционировать ОС ЕС в режиме МFT или MVT.

Для работы подсистемы планирования РОС на «обслуживающей» ЭВМ требуется раздел основной памяти объемом не менее 110К байт.

Подсистема планирования РОС обеспечивает повышение производительности вычислительной системы, высокую степень автоматизации вычислительного процесса, возможность удобного размещения оборудования и сокращение времени восстановления вычислительного процесса после сбоев оборудования.

Производительность вычислительной системы повышается благодаря тому, что используются улучшенные каналные программы

ввода входных потоков заданий и вывода выходных наборов данных. Применяются улучшенные алгоритмы распределения памяти прямого доступа, используемой в качестве внешней памяти, которые учитывают положение механизма доступа, и рационально используется оборудование вычислительной системы.

Одной из важных характеристик подсистемы планирования РОС является высокая степень автоматизации вычислительного процесса. Она обеспечивается тем, что РОС выполняет часть функций оператора по управлению ОС ЕС. Подсистема планирования РОС, выполняя функции оператора, берет на себя управление загрузкой операционных систем, запуск программы системного ввода, инициаторов, программы системного вывода и управление НМЛ и НМД «основных» ЭВМ.

Для управления подсистемой планирования РОС обычно используется несколько консолей оператора, каждой из которых можно назначить некоторый класс сообщений. Эти свойства РОС позволяют удобно размещать оборудование системы. Например, перфокарточные устройства ввода, перфокарточные устройства вывода, печатающие устройства вместе с соответствующими консолями можно разместить рядом с пунктами приема заданий и выдачи результатов выполнения заданий НМЛ и НМД. Консоль, используемую для выдачи запросов на установку и снятие томов, можно разместить недалеко от места хранения томов. Центральные процессоры могут быть размещены в отдельном помещении, не перегруженном периферийным оборудованием.

При работе ЭВМ с использованием РОС можно добиться значительного сокращения времени восстановления вычислительного процесса после сбоев оборудования благодаря тому, что при сбое «основной» ЭВМ восстановление заданий, выполнявшихся на ней, производится без их повторного ввода в систему. При сбое «обслуживающей» ЭВМ процесс выполнения заданий повторяется в момент, отраженного в контрольной точке РОС.

16.4. ОБРАБОТКА ЗАДАНИЙ ПОД УПРАВЛЕНИЕМ РОС

Подсистема планирования РОС вводит пакеты заданий с перфокарточных устройств ввода или НМЛ «обслуживающей» ЭВМ. Ввод заданий выполняет программа обслуживания ввода. Эта программа управляет чтением перфокарт или образов перфокарт с устройства ввода и осуществляет буферное запоминание их во внешней памяти РОС. Программа обслуживания ввода является реентерабельной, поэтому одна копия этой программы может обслуживать неограниченное число входных потоков.

Для каждого задания во внешней памяти РОС создаются наборы данных для хранения операторов языка управления заданиями и данных, принятых из входного потока. Создается также таблица управления заданием, которая вводится в очередь заданий РОС с учетом приоритета задания.

Имеется стандартный маршрут обработки задания, при котором его выполнение разбивается на ряд этапов (рис. 170).

1. Первым этапом обработки задания является планирование его выполнения. Подсистема планирования РОС выбирает для выполнения задания ту «основную» ЭВМ, которая имеет необходимое количество периферийных устройств, требующихся для предварительной установки томов прямого доступа или томов магнитной ленты.

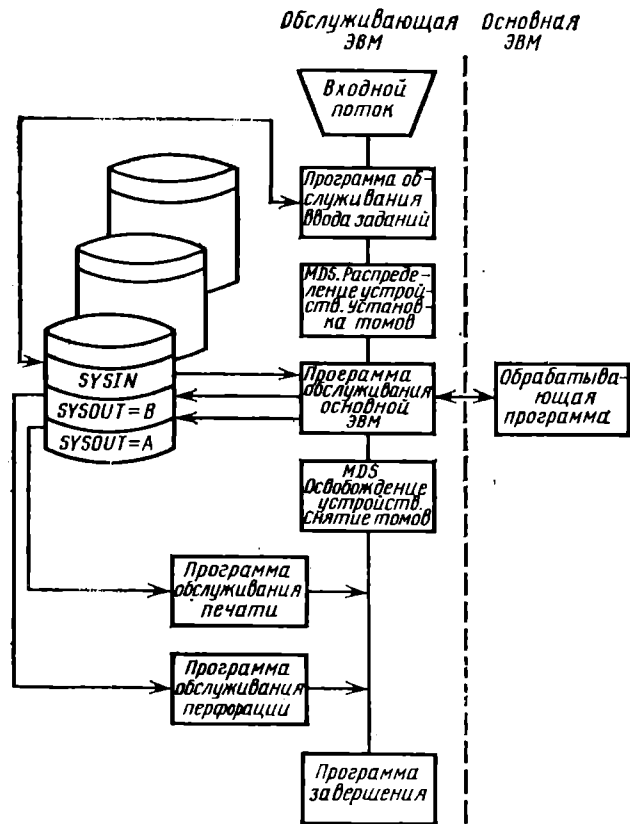


Рис. 170. Схема обработки заданий под управлением РОС

Предварительная установка томов, необходимых для выполнения задания, обеспечивается планировщиком периферийных устройств «основной» ЭВМ.

2. После завершения этапа планирования задание через адаптер канал-канал передается для выполнения программе системного ввода ОС ЕС на «основную» ЭВМ. Осуществляет это программа РОС, выполняющая обслуживание «основной» ЭВМ. После запуска задания программа обслуживания этой ЭВМ по запросам задания будет передавать через адаптер канал-канал входные наборы данных. Она же во время выполнения задания будет принимать

через адаптер и размещать во внешней памяти РОС данные, предназначенные для печати и перфорации.

3. Следующим этапом обработки является демонтаж томов магнитной ленты и томов прямого доступа, а также освобождение НМЛ и НМД, занятых заданием. Эти функции обеспечиваются планировщиком периферийных устройств «основной» ЭВМ.

4. После освобождения НМЛ и НМД программы обслуживания печати и перфорации выполняют на «обслуживающей» ЭВМ печать и перфорацию наборов данных задания в соответствии с инструкциями, указанными в управляющих картах подсистемы планирования РОС. Эти карты могут использоваться при оформлении задания. Они заменяют письменные инструкции программиста оператору. Например, программист с помощью соответствующей

управляющей карты может указать необходимость печати нескольких копий набора данных, форму вывода (вид бумаги) и т. д. При отсутствии таких карт печать и перфорация наборов данных выполняются в соответствии со стандартами вычислительной системы.

5. Последним этапом обработки задания является вывод его из системы, который выполняется на «обслуживающей» ЭВМ программой завершения.

Для определения уникального маршрута обработки, который может быть выбран с целью использования других возможностей РОС, задание дополняется управляющими картами подсистемы планирования РОС. Они определяют имена тех программ, выполнение которых необходимо для обработки задания.

Заслуживающими внимания возможностями подсистемы планирования РОС являются средства обработки внутренних заданий и обслуживающие программы.

Средства обработки внутренних заданий позволяют передавать на «обслуживающую» ЭВМ набор данных, расположенный либо на томе магнитной ленты, либо на томе прямого доступа. Этот набор данных по окончании передачи может быть обработан программой обслуживания ввода как обычный поток заданий, либо распечатан, либо отперфорирован. Причем эти действия могут быть выполнены параллельно с выполнением задания, передавшего этот набор.

Средства обработки внутренних заданий применяются заданиями, создающими поток заданий, и заданиями, требующими печати или перфорации результатов до завершения своего выполнения на «основной» ЭВМ.

Использование средств обработки внутренних заданий может потребоваться также при оснащении вычислительного центра диалоговыми системами телеобработки, передающими задания операционной системе, например, такими, как система диалогового удаленного ввода заданий в ОС ЕС — ДУВЗ. Эта система может функционировать на одной из «основных» ЭВМ. Наборы данных ДУВЗ могут передаваться средствами обработки внутренних заданий на «обслуживающую» ЭВМ для того, чтобы поместить задания, находящиеся в этих наборах данных, в общую очередь заданий. Выполнение заданий, созданных ДУВЗ, может осуществляться на любой «основной» ЭВМ системы.

С помощью обслуживающих программ подсистемы планирования РОС можно выполнить копирование или распечатку томов магнитной ленты, запись информации с перфокарт на тома магнитной ленты, дублирование перфокарт и т. д. Обычно эти работы выполняются с помощью автономных программ или требуют подготовки специальных заданий, использующих программы обслуживания, входящие в состав ОС ЕС. Во время выполнения этих довольно длительных работ автономными программами обработка заданий невозможна. Напротив, при использовании для этих же

работ обслуживающих программ РОС обработка заданий не прекращается. При этом «основные» ЭВМ не используются. Все действия выполняются на «обслуживающей» ЭВМ.

Описанная выше подсистема планирования РОС издания 1.0 может использоваться с ОС ЕС изданий 4.0 и 4.1. Дальнейшее развитие РОС связано, в первую очередь, с обеспечением совместного функционирования РОС с операционной системой 6.1. Кроме того, перспективным является включение в РОС средств обеспечения удаленного ввода заданий, управления работой сетей, состоящих из вычислительных систем, работающих под управлением РОС, и обеспечение возможности управления «основными» ЭВМ, работающими в режиме разделения времени.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1

Характеристики аппаратуры передачи данных ЕС ЭВМ

| Название и шифр | Скорость передачи Бод | Каналы связи | | | | | | Интерфейс | | | | |
|-----------------------|-----------------------------|-----------------|--------------------|-----------------|--------------------|--------------------------|--------------------------|-----------|----|----|---|---|
| | | тлр | | тлф | | широко- полос- ные | физи- ческая линия | С1 | С2 | С3 | | |
| | | выде- ленные | коммути- руемые | выде- ленные | коммути- руемые | | | | | | | |
| МОДЕМ 200 (ЕС-8002) | 200 | | | × | × | | | × | | | | |
| МОДЕМ 1200 (ЕС-8005) | 600, 1 200 | | | × | × | | | × | | | × | |
| МОДЕМ 1200 (ЕС-8006) | 600, 1 200 | | | × | × | | | × | | | × | |
| МОДЕМ 2400 (ЕС-8010) | 2 400 | | | × | × | | | × | | | × | |
| МОДЕМ 2400 (ЕС-8011) | 2 400 | | | × | × | | | × | | | × | |
| МОДЕМ 4800 (ЕС-8015) | 4 800 | | | × | × | | | × | | | × | |
| МОДЕМ 48000 (ЕС-8019) | 48 000 | | | | | | × | × | | | × | |
| УПС НУ (ЕС-8027) | 4 800 | | | | | | | × | | | × | |
| УПС НУ (ЕС-8028) | 9 600 | | | | | | | × | | | × | |
| УПС НУ (В) (ЕС-8029) | 96 000 | | | | | | | × | | | × | |
| УПС ТГ (ЕС-8030) | 200 | × | | | | | | × | | | × | |
| УПС ТГ (ЕС-8032) | 200 | × | | | | | | × | | | × | |
| УЗО-1200 (ЕС-8122) | 1 200 | | | | | | | | | | × | × |
| УЗО-4800 (ЕС-8135) | 4 800 | | | | | | | | | | × | × |
| УЗО-48000 (ЕС-8140) | 48 000 | | | | | | | | | | × | × |

Характеристики абонентских пунктов ЕС ЭВМ

| Название и шифр | Классификация | | | | | | | | | | Ввод | | | | | | Вывод | | | | | АПД | | | | |
|--|---------------|-------------------|-----------------|-------------|-------------------------------------|------------|------------|------------|-----------------------------|-----------------|--------|---------|------------|-------------------------|-----------------------------|-----------------|-------|------|------------------------|----------|------------|------------|------------|--------|--------|--|
| | Диалоговые | Пакетной передачи | Пропраммируемые | Сбор данных | Контроль и управление производством | Клавиатура | Перфокарты | Перфоленты | Карты с краевой перфорацией | Магнитная лента | Жетоны | Паспорт | Перфоленты | Последовательная печать | Карты с краевой перфорацией | Магнитная лента | Экран | АЦПУ | Метод защиты от ошибок | Модем 20 | Модем 1200 | Модем 2400 | Модем 4800 | УПС НУ | УПС ТТ | |
| АП-1 (ЕС-8501) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-2 (ЕС-8502) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-3 (ЕС-8503) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-4 (ЕС-8504) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-5 (ЕС-8505) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-6 (ЕС-8506) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-14 (ЕС-8514) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-31 (ЕС-8531) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-50 (ЕС-8550) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-61 (ЕС-8561) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-62 (ЕС-8562) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-63 (ЕС-8563) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-64 (ЕС-8564) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| АП-70 (ЕС-8570) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| ТА-5 элементный (ЕС-8592) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| ТА 7 элементный (ЕС-8593) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| Дисплей, система одиночная удаленная (ЕС-7925) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |
| Дисплей, система групповая удаленная (ЕС-7921) | × | | | | | × | | | | | | | | | | | | | × | × | × | × | × | × | × | |

ПРИЛОЖЕНИЕ 3

Характеристики мультиплексоров передачи данных ЕС ЭВМ

| Название и шифр | Максимальное число каналов связи | | Аппаратура, подключаемая на удаленном конце | | | | | | | | | | | | | | Работа в си- стеме | | Скорость, бит/с | Метод защиты от ошибок | | | | |
|--------------------|---|------------|---|------|------|------|------------|-------|-------|-------|-------|-------|-------|-------|-------|-----------------|--------------------------|-----|--------------------|------------------------|---------------|----------------|--------|---|
| | полу дуплексные | дуплексные | АП-1 | АП-2 | АП-3 | АП-4 | АП-5, АП-6 | АП-14 | АП-31 | АП-50 | АП-61 | АП-62 | АП-63 | АП-64 | АП-70 | ТА 5 элементный | ТА 7 элементный | МПД | | | двухпунктовый | многопунктовый | | |
| | | | 16 | 7 | 4 | 88 | 2 | 16 | 64 | 176 | 4 | 12 | 15/32 | × | × | × | × | × | | | × | × | × | × |
| МПД-1А (ЕС-8400) | | | × | | | | | | | | | | | | | | | | × | × | × | 50—4 800 | М, Ц | |
| МПД-1 (ЕС-8401) | | | × | | | | | | | | | | | | | | | | × | × | × | 50—2 400 | М | |
| МПД-2 (ЕС-8402) | | | × | | | | | | | | | | | | | | | | × | × | × | 50—4 800 | М, Ц | |
| МПД-3 (ЕС-8403) | | | × | × | | | | | | | | | | | | | | | × | × | × | 50—48 000 | М, Ц | |
| МПД-4 (ЕС-8404) | | | × | | | | | | | | | | | | | | | | × | × | × | 200—1 200 | М | |
| МПД-1 (ЕС-8410) | | | × | | | | | | | | | | | | | | | | × | × | × | 50—2 400 | М, Ц | |
| УМПД (ЕС-8421) | 20 | × | Первичная сторона | | | | | | | | | | | | | | | | | × | × | × | 50—200 | |
| | | | Вторичная сторона | | | | | | | | | | | | | | | | | | | | | |

Команды ЭВМ ЕС-1033 и время их выполнения

| Наименование | Мне-моника | Код опе-рации | Тип | Особые случаи (код прерывания) | Время выпол-нения, мкс |
|--|------------|---------------|-------|--------------------------------|------------------------|
| Команды обработки данных с фиксированной запятой | | | | | |
| ЗАГРУЗКА | LR | 18 | RR | — | 1,4 |
| ЗАГРУЗКА | L | 58 | RX | 4, 5, 6 | 2,7 |
| ЗАГРУЗКА ПОЛУСЛО- ВА | LN | 48 | RX | 4, 5, 6 | 3,3 |
| ЗАГРУЗКА И ПРО- ВЕРКА | LTR | 12 | RR, C | — | 1,4 |
| ЗАГРУЗКА ДОПОЛНЕ- НИЯ | LCR | 13 | RR, C | 8 | 1,55 |
| ЗАГРУЗКА ПОЛОЖИ- ТЕЛЬНАЯ | LPR | 10 | RR, C | 8 | 1,55 |
| ЗАГРУЗКА ОТРИЦА- ТЕЛЬНАЯ | LNR | 11 | RR, C | — | 1,55 |
| ЗАГРУЗКА ГРУППО- ВАЯ | LM | 98 | RS | 4, 5, 6 | 3,9— 20,4 |
| СЛОЖЕНИЕ | AR | 1A | RR, C | 8 | 1,4 |
| СЛОЖЕНИЕ | A | 5A | RX, C | 4, 5, 6, 8 | 2,7 |
| СЛОЖЕНИЕ ПОЛУ- СЛОВА | AN | 4A | RX, C | 4, 5, 6, 8 | 3,3 |
| СЛОЖЕНИЕ КОДОВ | ALR | 1E | RR, C | — | 1,4 |
| СЛОЖЕНИЕ КОДОВ | AL | 5E | RX, C | 4, 5, 6 | 2,7 |
| ВЫЧИТАНИЕ | SR | 1B | RR, C | 8 | 1,4 |
| ВЫЧИТАНИЕ | S | 5B | RX, C | 4, 5, 6, 8 | 2,7 |
| ВЫЧИТАНИЕ ПОЛУ- СЛОВА | SN | 4B | RX, C | 4, 5, 6, 8 | 3,3 |
| ВЫЧИТАНИЕ КОДОВ | SLR | 1F | RR, C | — | 1,4 |
| ВЫЧИТАНИЕ КОДОВ | SL | 5F | RX, C | 4, 5, 6 | 2,7 |
| СРАВНЕНИЕ | CR | 19 | RR, C | — | 1,4 |
| СРАВНЕНИЕ | C | 59 | RX, C | 4, 5, 6 | 2,7 |
| СРАВНЕНИЕ ПОЛУ- СЛОВА | CH | 49 | RX, C | 4, 5, 6 | 3,3 |
| УМНОЖЕНИЕ | MR | 1C | RR | 6 | 8,5 |
| УМНОЖЕНИЕ | M | 5C | RX | 4, 5, 6 | 9,7 |
| УМНОЖЕНИЕ ПОЛУ- СЛОВА | MN | 4C | RX | 4, 5, 6 | 8,7 |
| ДЕЛЕНИЕ | DR | 1D | RR | 6, 9 | 14,6 |
| ДЕЛЕНИЕ | D | 5D | RX | 4, 5, 6, 9 | 15,6 |
| ПРЕОБРАЗОВАНИЕ В ДВОИЧНУЮ | CVB | 4F | RX | 4, 5, 6, 7, 9 | 18,6 |
| ПРЕОБРАЗОВАНИЕ В ДЕСЯТИЧНУЮ | CVD | 4E | RX | 4, 5, 6 | 10,5 |
| ЗАПИСЬ В ПАМЯТЬ | ST | 50 | RX | 4, 5, 6 | 2,4 |
| ЗАПИСЬ В ПАМЯТЬ ПОЛУСЛОВА | STH | 40 | RX | 4, 5, 6 | 2,7 |
| ЗАПИСЬ В ПАМЯТЬ ГРУППОВАЯ | STM | 90 | RS | 4, 5, 6 | 30,6— 20,4 |
| СДВИГ ВЛЕВО | SLA | 8B | RS, C | 9 | 2,7 |
| СДВИГ ВПРАВО | SRA | 8A | RS, C | — | 2,7 |
| СДВИГ ВЛЕВО ДВОЙНОЙ | SLDA | 8F | RS, C | 6, 9 | 3,3 |

| Наименование | Мне-моника | Код операции | Тип | Особые случаи (код прерывания) | Время выполнения, мкс |
|--|------------|--------------|-------|--------------------------------|-----------------------|
| СДВИГ ВПРАВО ДВОЙНОЙ | SRDA | 8E | RS, G | 6 | 2,7 |
| Команды обработки данных с плавающей запятой | | | | | |
| ЗАГРУЗКА (ДЛИННАЯ) | LDR | 28 | RR | 6 | 1,7 |
| ЗАГРУЗКА (ДЛИННАЯ) | LD | 68 | RX | 4, 5, 6 | 4,2 |
| ЗАГРУЗКА (КОРОТКАЯ) | LER | 38 | RR | 6 | 1,4 |
| ЗАГРУЗКА (КОРОТКАЯ) | LE | 78 | RX | 4, 5, 6 | 3,0 |
| ЗАГРУЗКА И ПРОВЕРКА (ДЛИННАЯ) | LTDR | 22 | RR, G | 6 | 1,4 |
| ЗАГРУЗКА И ПРОВЕРКА (КОРОТКАЯ) | LTER | 32 | RR, G | 6 | 1,4 |
| ЗАГРУЗКА ДОПОЛНЕНИЯ (ДЛИННАЯ) | LCDR | 23 | RR, G | 6 | 2,0 |
| ЗАГРУЗКА ДОПОЛНЕНИЯ (КОРОТКАЯ) | LCER | 33 | RR, G | 6 | 1,4 |
| ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ (ДЛИННАЯ) | LPDR | 20 | RR, G | 6 | 2,0 |
| ЗАГРУЗКА ПОЛОЖИТЕЛЬНАЯ (КОРОТКАЯ) | LPER | 30 | RR, G | 6 | 1,4 |
| ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ (ДЛИННАЯ) | LNDR | 21 | RR, G | 6 | 2,0 |
| ЗАГРУЗКА ОТРИЦАТЕЛЬНАЯ (КОРОТКАЯ) | LNER | 31 | RR, G | 6 | 1,4 |
| СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ) | ADR | 2A | RR, G | 6, G, D, E | 4,2 |
| СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (ДЛИННОЕ) | AD | 6A | RX, G | 4, 5, 6, G, D, E | 6,0 |
| СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ) | AER | 3A | RR, G | 6, G, D, E | 3,8 |
| СЛОЖЕНИЕ С НОРМАЛИЗАЦИЕЙ (КОРОТКОЕ) | AE | 7A | RX, G | 4, 5, 6, G, D, E | 5,0 |
| СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ) | AWR | 2E | RR, G | 6, G, E | 4,0 |
| СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ) | AW | 6E | RX, G | 4, 5, 6, G, E | 5,9 |

| Наименование | Мне- моника | Код опе- рации | Тип | Особые случаи (код прерывания) | Время выпол- нения, мс |
|--|----------------|----------------------|-------|--------------------------------------|---------------------------------|
| СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ) | AUR | 3E | RR, G | 6, G, E | 3,5 |
| СЛОЖЕНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ) | AU | 7E | RX, G | 4, 5, 6, G, E | 4,5 |
| ВЫЧИТАНИЕ С НОР- МАЛИЗАЦИЕЙ (ДЛИННОЕ) | SDR | 2B | RR, C | 6, G, D, E | 6,3 |
| ВЫЧИТАНИЕ С НОР- МАЛИЗАЦИЕЙ (ДЛИННОЕ) | SD | 6B | RX, G | 4, 5, 6, G, D, E | 7,2 |
| ВЫЧИТАНИЕ С НОР- МАЛИЗАЦИЕЙ (КОРОТКОЕ) | SER | 3B | RR, G | 6, G, D, E | 3,8 |
| ВЫЧИТАНИЕ С НОР- МАЛИЗАЦИЕЙ (КОРОТКОЕ) | SE | 7B | RX, G | 4, 5, 6, G, D, E | 5,1 |
| ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ) | SWR | 2F | RR, G | 6, G, E | 4,4 |
| ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (ДЛИННОЕ) | SW | 6F | RX, G | 4, 5, 6, C, E | 5,9 |
| ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ) | SUR | 3F | RR, G | 6, G, E | 3,5 |
| ВЫЧИТАНИЕ БЕЗ НОРМАЛИЗАЦИИ (КОРОТКОЕ) | SU | 7F | RX, C | 4, 5, 6, C, E | 4,8 |
| СРАВНЕНИЕ (ДЛИН- НОЕ) | CDR | 29 | RR, G | 6 | 3,8 |
| СРАВНЕНИЕ (ДЛИН- НОЕ) | CD | 69 | RX, C | 4, 5, 6 | 5,3 |
| СРАВНЕНИЕ (КО- РОТКОЕ) | CER | 39 | RR, G | 6 | 2,6 |
| СРАВНЕНИЕ (КО- РОТКОЕ) | CE | 79 | RX, C | 4, 5, 6 | 3,9 |
| ПОПОЛАМ (ДЛИН- НОЕ) | HDR | 24 | RR | 6, D | 3,6 |
| ПОПОЛАМ (КОРОТ- КОЕ) | HER | 34 | RR | 6, D | 2,4 |
| УМНОЖЕНИЕ (ДЛИННОЕ) | MDR | 2G | RR | 6, G, D | 19,2 |

| Наименование | Мне- моника | Код опе- рации | Тип | Особые случаи (код прерывания) | Время выпол- нения, мс |
|-------------------------------------|----------------|----------------------|-------|--------------------------------------|---------------------------------|
| УМНОЖЕНИЕ (ДЛИННОЕ) | MD | 6C | RX | 4, 5, 6, C, D | 21,7 |
| УМНОЖЕНИЕ (КОРОТКОЕ) | MER | 3C | RR | 6, C, D | 9,5 |
| УМНОЖЕНИЕ (КОРОТКОЕ) | ME | 7C | RX | 4, 5, 6, C, D | 11,0 |
| ДЕЛЕНИЕ (ДЛИН- НОЕ) | DDR | 2D | RR | 6, C, D, F | 23,1 |
| ДЕЛЕНИЕ (ДЛИН- НОЕ) | DD | 6D | RX | 4, 5, 6, C, D, F | 24,0 |
| ДЕЛЕНИЕ (КОРОТ- КОЕ) | DER | 3D | RR | 6, C, D, F | 16,8 |
| ДЕЛЕНИЕ (КОРОТ- КОЕ) | DE | 7D | RX | 4, 5, 6, C, D, F | 17,7 |
| ЗАПИСЬ В ПАМЯТЬ (ДЛИННАЯ) | STD | 60 | RX | 4, 5, 6 | 3,6 |
| ЗАПИСЬ В ПАМЯТЬ (КОРОТКАЯ) | STE | 70 | RX | 4, 5, 6 | 2,7 |
| Команды обработки десятичных данных | | | | | |
| СЛОЖЕНИЕ ДЕСЯ- ТИЧНОЕ | AP | FA | SS, C | 4, 5, 7, A | 9—24 |
| ВЫЧИТАНИЕ ДЕСЯ- ТИЧНОЕ | SP | FB | SS, C | 4, 5, 7, A | 9—24 |
| СЛОЖЕНИЕ С ОЧИ- СТКОЙ | ZAP | F8 | SS, C | 4, 5, 7, A | 8—18 |
| СРАВНЕНИЕ ДЕСЯ- ТИЧНОЕ | CP | F9 | SS, C | 4, 5, 7 | 8—19 |
| УМНОЖЕНИЕ ДЕСЯ- ТИЧНОЕ | MP | FG | SS | 4, 5, 6, 7 | 18—46 |
| ДЕЛЕНИЕ ДЕСЯ- ТИЧНОЕ | DP | FD | SS | 4, 5, 6, 7, B | 19—89 |
| УПАКОВАТЬ | PACK | F2 | SS | 4, 5 | 10,2 |
| РАСПАКОВАТЬ | UNPK | F3 | SS | 4, 5 | 13,6 |
| ПЕРЕСЫЛКА СО СДВИГОМ | MVO | F1 | SS | 4, 5 | 9—24 |
| Команды логических операций | | | | | |
| ПЕРЕСЫЛКА НЕПОСР. | MVI | 92 | SI | 4, 5 | 2,1 |
| ПЕРЕСЫЛКА СИМВ. | MVC | D2 | SS | 4, 5 | 7—178 |
| ПЕРЕСЫЛКА ЦИФР | MVN | D1 | SS | 4, 5 | 7—234 |
| ПЕРЕСЫЛКА ЗОН | MVZ | D3 | SS | 4, 5 | 7—234 |

| Наименование | Мне- моника | Код опе- рации | Тип | Особые случай (код прерывания) | Время выпол- нения, мс |
|-------------------------------|----------------|----------------------|-------|--------------------------------------|---------------------------------|
| СРАВНЕНИЕ КОДОВ | CLR | 15 | RR, C | — | 1,4 |
| СРАВНЕНИЕ КОДОВ | CL | 55 | RX, C | 4, 5, 6 | 3,0 |
| СРАВНЕНИЕ КОДОВ | CLI | 95 | SI, G | 4, 5 | 3,0 |
| СРАВНЕНИЕ КОДОВ | CLC | D5 | SS, G | 4, 5 | 6,2—187 |
| И | NR | 14 | RR, C | — | 1,4 |
| И | N | 54 | RX, C | 4, 5, 6 | 3,0 |
| И | NI | 94 | SI, G | 4, 5 | 3,6 |
| И | NC | D4 | SS, G | 4, 5 | 6,9—234 |
| ИЛИ | OR | 16 | RR, C | — | 1,4 |
| ИЛИ | O | 56 | RX, C | 4, 5, 6 | 3,0 |
| ИЛИ | OI | 96 | SI, G | 4, 5 | 3,6 |
| ИЛИ | OC | D6 | SS, G | 4, 5 | 7—234 |
| ИСКЛЮЧАЮЩЕЕ ИЛИ | XR | 17 | RR, C | — | 1,4 |
| ИСКЛЮЧАЮЩЕЕ ИЛИ | X | 57 | RX, C | 4, 5, 6 | 3,0 |
| ИСКЛЮЧАЮЩЕЕ ИЛИ | XI | 97 | SI, G | 4, 5 | 3,6 |
| ИСКЛЮЧАЮЩЕЕ ИЛИ | XC | D7 | SS, G | 4, 5 | 7—234 |
| ПРОВЕРИТЬ ПО МАС- КЕ | TM | 91 | SI, G | 4, 5 | 3,6 |
| ПРОЧИТАТЬ СИМВОЛ | IC | 43 | RX | 4, 5 | 3,3 |
| ЗАПИСЬ В ПАМЯТЬ СИМВОЛА | STC | 42 | RX | 4, 5 | 2,7 |
| ЗАГРУЗКА АДРЕСА | LA | 41 | RX | — | 2,1 |
| ПЕРЕКОДИРОВАТЬ | TR | DC | SS | 4, 5 | 7—742 |
| ПЕРЕКОДИРОВАТЬ И ПРОВЕРИТЬ | TRT | DD | SS, C | 4, 5 | 7—690 |
| ОТРЕДАКТИРОВАТЬ | ED | DE | SS, C | 4, 5, 7 | 10,5— 9996 |
| ОТРЕДАКТИРОВАТЬ И ОТМЕТИТЬ | EDMK | DF | SS, C | 4, 5, 7 | 10,8— 998 |
| СДВИГ ВЛЕВО | SLL | 89 | RS | — | 2,7 |
| СДВИГ ВПРАВО | SRL | 88 | RS | — | 2,7 |
| СДВИГ ВЛЕВО ДВОЙ- НОЙ | SLDL | 8D | RS | 6 | 3,3 |
| СДВИГ ВПРАВО ДВОЙНОЙ | SRDL | 8C | RS | 6 | 3,75 |
| Команды переходов | | | | | |
| УСЛОВНЫЙ ПЕРЕ- ХОД | BCR | 07 | RR | — | 1,2 |

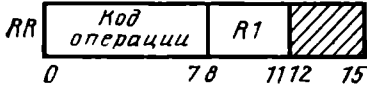
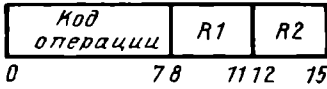
| Наименование | Мне- моника | Код опе- рации | Тип | Особые случаи (код прерывания) | Время выпол- нения, мс |
|---|----------------|----------------------|-------|--------------------------------------|---------------------------------|
| УСЛОВНЫЙ ПЕРЕ- ХОД | BC | 47 | RX | — | 1,2 |
| ПЕРЕХОД С ВОЗ- ВРАТОМ | BALR | 05 | RR | — | 1,05 |
| ПЕРЕХОД С ВОЗ- ВРАТОМ | BAL | 45 | RX | — | 1,7 |
| ПЕРЕХОД ПО СЧЕТЧИКУ | BCTR | 06 | RR | — | 1,2 |
| ПЕРЕХОД ПО СЧЕТЧИКУ | BCT | 46 | RX | — | 2,0 |
| ПЕРЕХОД, ЕСЛИ ИН- ДЕКС БОЛЬШЕ | BXH | 86 | RS | — | 2,4 |
| ПЕРЕХОД, ЕСЛИ ИНДЕКС МЕНЬШЕ ИЛИ РАВЕН | BXLE | 87 | RS | — | 2,4 |
| ВЫПОЛНИТЬ | EX | 44 | RX | 4, 5, 6, 3 | 4,2 |
| Команды переключения состояния | | | | | |
| ЗАГРУЗКА ССП | LPSW | 82 | SI, C | 2, 4, 5, 6 | 3,9 |
| УСТАНОВИТЬ МАС- КУ ПРОГРАММЫ | SPM | 04 | RR, C | — | 1,7 |
| УСТАНОВИТЬ МАС- КУ СИСТЕМЫ | SSM | 80 | SI | 2, 4, 5 | 3,6 |
| ОБРАЩЕНИЕ К СУ- ПЕРВИЗОРУ | SVC | 0A | RR | — | 1,7 |
| УСТАНОВИТЬ КЛЮЧ ПАМЯТИ | SSK | 08 | RR | 2, 5, 6 | 3,2 |
| ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ | ISK | 09 | RR | 2, 5, 6 | 3,2 |
| ПРЯМАЯ ЗАПИСЬ | WRD | 84 | SI | 2, 4, 5 | 2,7 |
| ПРЯМОЕ ЧТЕНИЕ | RRD | 85 | SI | 2, 4, 5 | 3,6 * |
| ПРОВЕРИТЬ И УС- ТАНОВИТЬ | TS | 93 | SI, C | 4, 5 | 2,7 |
| ДИАГНОСТИКА | — | 83 | SI | 2, 6 | 2,7 * |
| Команды ввода-вывода | | | | | |
| НАЧАТЬ ВВОД-ВЫ- ВОД | SIO | 9C | SI, C | 2 | 3,0 * |
| ОПРОСИТЬ ВВОД- ВЫВОД | TIO | 9D | SI, C | 2 | 3,0 * |
| ОСТАНОВИТЬ ВВОД- ВЫВОД | HIO | 9E | SI, C | 2 | 3,0 * |
| ОПРОСИТЬ КАНАЛ | TCH | 9F | SI, C | 2 | 3,0 * |

Примечания: 1. C — устанавливается признак результата.
2. Код прерывания определяет причину прерывания, прил. 3.
3. Время выполнения команд приведено среднестатистическое при выпол-
нении на линейных участках программы (с учетом подвыборки команд). Вероят-
ность начала команд с первого или третьего байта слова ОП одинакова.
4. Вероятность индексирования операций формата RX составляет 0,19.
5. В операциях с нормализацией предполагается нормализация на одну
шестнадцатиричную цифру.

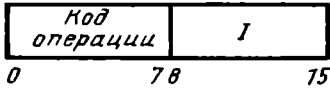
* При расчете времени операций учитывается только время работы обору-
дования процессора.

Команды

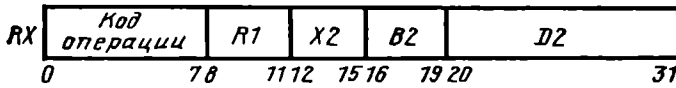
Все команды RR,
кроме SPM и SVC



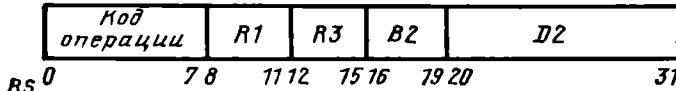
SPM



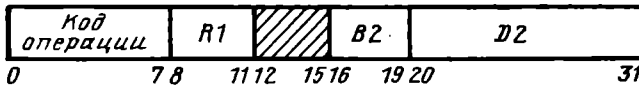
SVC



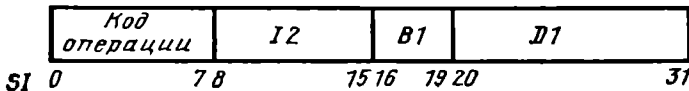
Все команды RX



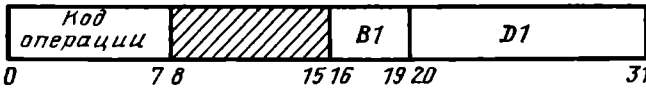
BXH, BXLE, STM
и LM



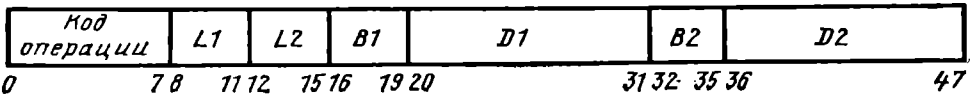
Все команды
сдвига



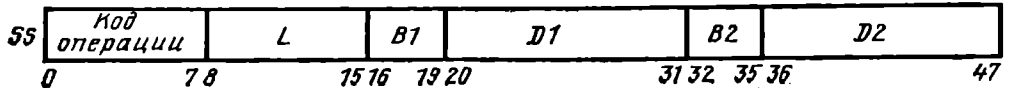
NI, CLI, OI, XI,
TM, MVI, RDD,
WRD.



SIO, HIO, TIO, TCH,
SSM, LPSW, TS



AP, SP, MP, DP,
CP, ZAP, MVD,
PACH, UNPK



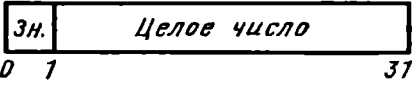
NC, CLC, DC, XC,
NVC, MVN, MVZ,
TR, TRT, EDMK,
ED

Форматы данных

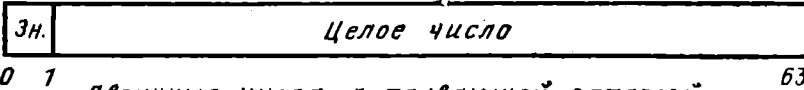
Двоичные числа с фиксированной запятой
Полуслово



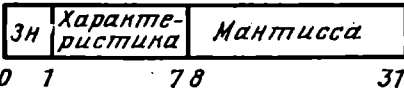
Полное слово



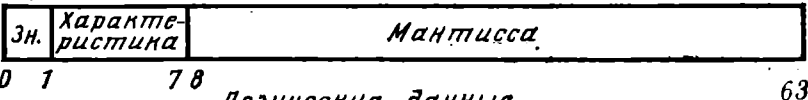
Двойное слово



Двоичные числа с плавающей запятой
Короткое



Длинное

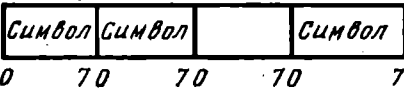


Логические данные

Полное слово

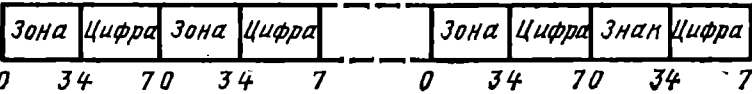


Поле переменной длины

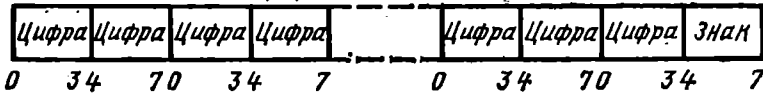


Десятичные данные

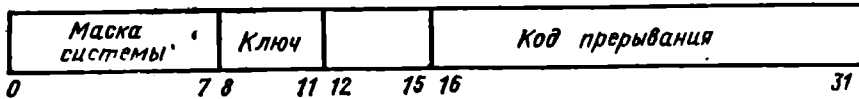
Формат с зоной



Упакованный формат



Слово состояния программы (ССП)



- Маска системных прерываний:
 - 0 мультиплексный канал
 - 1 селекторный канал 1
 - 2 селекторный канал 2
 - 3 селекторный канал 3
 - 4/6 не используются
 - 7 Прерывания от таймера, кнопки «ПРЕРЫВАНИЕ» на пульте управления или от внешнего источника
 - 8/11 Ключ защиты памяти
 - 12 Единичное и нулевое значение означают работу с КОИ-8 или ДКОИ соответственно
 - 13 Маска прерываний от схем контроля
 - 14 Единичное и нулевое значение означают состояния ОЖИДАНИЕ/СЧЕТ соответственно
 - 15 Единичное и нулевое значение означают состояния ЗАДАЧА/СУПЕРВИЗОР соответственно
 - 16/31 Код прерывания
 - 32/33 Код длины команды (КДК)
 - 34/35 Признак результата (ПР)
 - 36/39 Маска программных прерываний:
 - 36 переполнение в операции с фиксированной запятой
 - 37 переполнение в десятичной операции
 - 38 исчезновение порядка
 - 39 потеря значимости
 - 40/63 Адрес команды
- Единичный бит в маске разрешает прерывание, нулевой запрещает его.

Прерывания

| Источник прерывания | Код прерывания | Маска (разряд ССП) | Выполнение команды |
|---|---------------------|--------------------|------------------------------|
| От схем контроля (приоритет 1) | | | |
| Машинная ошибка | 0000 0000 0000 0000 | 13 | Прекращается |
| Программные (приоритет 2) | | | |
| Некорректность кода операции | 0000 0000 0000 0001 | — | Подавляется |
| Привилегированная операция | 0000 0000 0000 0010 | — | » |
| Некорректность команды ВЫПОЛНИТЬ | 0000 0000 0000 0011 | — | » |
| Нарушение защиты памяти | 0000 0000 0000 0100 | — | Подавляется или прекращается |
| Неправильная адресация | 0000 0000 0000 0101 | — | Подавляется |
| Неправильная спецификация | 0000 0000 0000 0110 | — | » |
| Неправильные данные | 0000 0000 0000 0111 | — | Прекращается |
| Переполнение в операции с фиксированной запятой | 0000 0000 0000 1000 | 36 | Завершается |
| Некорректность деления с фиксированной запятой | 0000 0000 0000 1001 | — | Подавляется или завершается |
| Переполнение в десятичной операции | 0000 0000 0000 1010 | 37 | Завершается |
| Некорректность десятичного деления | 0000 0000 0000 1011 | — | Подавляется |
| Переполнение порядка | 0000 0000 0000 1100 | — | Прекращается |
| Исчезновение порядка | 0000 0000 0000 1101 | 38 | Завершается |
| Потеря значимости | 0000 0000 0000 1110 | 39 | » |
| Некорректность деления в операции с плавающей запятой | 0000 0000 0000 1111 | — | Подавляется |
| Выполнение команды SSM в режиме МУЛЬТИСИСТЕМА | 0000 0000 0001 0010 | — | » |
| Обращение к супервизору (приоритет 2) | | | |
| Разряды команды SVC | 0000 0000 rrrr rrrr | — | Завершается |
| Внешние (приоритет 3) | | | |
| Таймер | 0000 0000 1bvv vvvv | 7 | » |
| Кнопка прерывания | 0000 0000 v1vv vvvv | 7 | » |
| Сигнал ВНЕШНЯЯ ОШИБКА (BC-2) | 0000 0000 vv1v vvvv | 7 | » |
| Сигнал ВЫЗОВ СИСТЕМЫ (BC-3) | 0000 0000 vvv1 vvvv | 7 | » |
| Ввод-вывод (приоритет 4) | | | |
| Мультиплексный канал | 0000 0000 aaaa aaaa | 0 | » |
| Селекторный канал 1 | 0000 0001 aaaa aaaa | 1 | » |
| Селекторный канал 2 | 0000 0010 aaaa aaaa | 2 | » |
| Селекторный канал 3 | 0000 0011 aaaa aaaa | 3 | » |

Примечание: а — биты адреса устройства; в — другие условия внешних прерываний; р — биты полей R1 и R2 команды ОБРАЩЕНИЕ К СУПЕРВИЗОРУ.

Признаки результата операций

| Операции | Значение признака | | | |
|---|-------------------|-------------------------|--------------------|-------------------------|
| | 0 | 1 | 2 | 3 |
| Арифметические с фиксированной запятой: | | | | |
| сложение и вычитание, загрузка дополнения, сдвиги влево | 0 | <0 | $\triangleright 0$ | Переполнение |
| загрузка и проверка, сдвиги вправо | 0 | <0 | $\triangleright 0$ | — |
| загрузка отрицательная | 0 | < 0 | — | — |
| загрузка положительная | 0 | — | >0 | — |
| сравнение | Равно | Меньше | Больше | — |
| сложение кодов | 0, нет переноса | $\neq 0$, нет переноса | 0, есть перенос | $\neq 0$, есть перенос |
| вычитание кодов | 0, нет переноса | $\neq 0$, нет переноса | 0, есть перенос | $\neq 0$, есть перенос |
| Арифметические с плавающей запятой: | | | | |
| сложение, вычитание | 0 | <0 | $\triangleright 0$ | Переполнение |
| загрузка и проверка, загрузка дополнения | 0 | <0 | $\triangleright 0$ | — |
| загрузка положительная | 0 | — | $\triangleright 0$ | — |
| загрузка отрицательная | 0 | <0 | — | — |
| сравнение | Равно | Меньше | Больше | — |

| Операции | Значение признака | | | |
|----------------------------|--------------------------|--------------------------|-------------------|--------------|
| | 0 | 1 | 2 | 3 |
| Арифметические десятичные: | | | | |
| сложение, вычитание | 0 | <0 | >0 | Переполнение |
| сравнение | 0 | Меньше | Больше | — |
| сложение с очисткой | Равно | <0 | >0 | Переполнение |
| Логические | | | | |
| И, ИЛИ, исключающее ИЛИ | 0 | ≠0 | — | — |
| редактирование | 0 | <0 | >0 | — |
| проверить по маске | Нуль | Нули и единицы | — | Единицы |
| перекодировать и проверить | Нуль | Не завершилась | Завершилась | — |
| Переключения состояния: | | | | |
| проверить и установить | Нуль | Единица | — | — |
| Ввода-вывода: | | | | |
| остановить ввод-вывод | Прерывание не обработано | ССК запомнено | Прекращение | Выключен |
| начать ввод-вывод | Успешный | ССК запомнено | Занят | Выключен |
| опросить канал | Доступны | Прерывание не обработано | Монопольный режим | Выключен |
| опросить ввод-вывод | Доступны | ССК запомнено | Занят | Выключен |

Значение признака результата может быть также изменено с помощью команд ЗАГРУЗКА ССП, УСТАНОВИТЬ МАСКУ СИСТЕМЫ, ДИАГНОСТИКА и при прерывании.

ПРОГРАММЫ

Постоянно распределенные ячейки памяти

| Десятичный адрес | Шестнадцатиричный адрес | Длина | Назначение | |
|------------------|-------------------------|---|---|----------------------------------|
| 0 | 00 | Два слова | ССП начальной загрузки | |
| 8 | 08 | | УСК1 начальной загрузки программы | |
| 16 | 10 | | УСК2 начальной загрузки программы | |
| 24 | 18 | | Старое ССП для внешних прерываний | |
| 32 | 20 | | Старое ССП для обращения к супервизору | |
| 40 | 28 | | Старое ССП для программных прерываний | |
| 48 | 30 | | Старое ССП для прерываний от схем контроля | |
| 56 | 38 | | Старое ССП для прерываний от ввода-вывода | |
| 64 | 40 | | Одно слово | Слово состояния канала (ССК) |
| 72 | 48 | | | Адресное слово канала (АСК) |
| 76 | 4C | | | Не используется |
| 80 | 50 | | | Таймер |
| 84 | 54 | | | Не используется |
| 88 | 58 | | | Новое ССП для внешних прерываний |
| 96 | 60 | Новое ССП для обращения к супервизору | | |
| 104 | 68 | Новое ССП для программных прерываний | | |
| 112 | 70 | Новое ССП для прерываний от схем контроля | | |
| 120 | 78 | Новое ССП для прерываний от ввода-вывода | | |
| 128—540 | 80—21C | 104 слова | Область записи состояния (область логаута): | |
| 128—132 | 80—84 | Два слова | оперативной памяти | |
| 136—408 | 88—198 | 69 слов | центрального процессора | |
| 412—416 | 19C—1A0 | Два слова | диагностических средств | |
| 420—436 | 1A4—1B4 | 5 слов | общего канала | |
| 440—456 | 1B8—1C8 | 5 слов | ячейки не используются | |
| 460—540 | 1CC—21C | 21 слово | селекторного и мультиплексного каналов | |

Список сокращений

| | |
|----------|---|
| АБОП | — адрес БОП |
| АДР | — адрес |
| АДР-А | — адрес абонента |
| АДР-К | — адрес канала |
| АДМК | — адрес ДМК |
| АКК | — адаптер канал-канал |
| АЛБ | — арифметико-логический блок |
| АЛУ | — арифметико-логическое устройство |
| АН НПУ | — анализ неуспешной передачи управления |
| АН СР | — анализ сравнением |
| АП | — абонентский пункт |
| АПД | — аппаратура передачи данных |
| АСК | — адресное слово канала |
| БД | — блок диагностики |
| БЗМ | — блок запоминающий магнитный |
| БК | — блок контроля |
| БКНВ | — блок конвертирования (преобразования чисел из одной системы счисления в другую) |
| БК УП | — блок коммутации управляющей информации |
| БК ЧИ | — блок коммутации числовой информации |
| БЛ ЗП ПЛ | — блокировка записи в РСДО в операциях с плавающей запятой |
| БЛ ЗПД | — блокировка записи в РСДО в операциях десятичной арифметики |
| БЛК-К | — блокировка канала |
| БЛ СБРМ | — блокировка сброса магистралей (в ЦП) |
| БЛ ОШ ОП | — блокировка ошибок ОП |
| БМК | — блок магистрального коммутатора |
| БМУ | — блок микропрограммного управления |
| БОД | — блок обмена данными |
| БОП | — блок односторонней памяти |
| БОПР | — блок обработки прерываний |
| БПРУ | — блок прямого управления |
| БПРД | — блок пультовых режимов и диагностики |
| БП УСК | — блок приема УСК |
| БРОП | — блок регистров операционных |
| БРП | — блок регистров программных |
| БРР | — блок режимов работы |
| БРСДВ | — блок регистров сдвиговых |
| БРУ | — блок регистров управляющих |
| БС | — блок синхронизации |
| БСМ | — блокировка сброса магистралей (в БД) |
| БСУ | — байт состояния устройства |
| БУ ВК | — блок управления вычислительным комплексом |
| БУ ОП | — блок управления оперативной памятью |
| БУ ПД | — блок управления передачей данных |
| БУ ПМП | — блок управления памятью микропрограмм |
| БФА БОП | — блок формирования адреса БОП |
| БФ МП | — блок формирования микроприказов |
| ВБАВ | — ветвление безусловное по адресу возврата |
| ВБР-К | — выборка канала |
| ВКЛ ПА | — включение переадресации операционных регистров по значению счетчика циклов |
| ВУ | — внешнее устройство |
| ВЫД ИНФ | — выдача информации из ОП |
| Д-ИНФ | — диагностическая информация |

| | |
|-----------------------|---|
| ДМК | — диагностическая микрокоманда |
| ДОСТ | — диагностический останов |
| ДП | — десятичное переполнение (в операциях десятичной арифметики) |
| ЗП | — запись |
| ЗПР | — запрос |
| ЗПРВ | — запрос на выборку команды |
| ЗПРК | — запрос канала |
| ЗПРП | — запрос процессора |
| ЗПРС | — запрос опережающий |
| ИЗМ-А | — измерение от абонента |
| ИЗМ-К | — измерение от канала |
| ИНФ-А | — информация от абонента |
| ИНФ-К | — информация от канала |
| ИС | — интегральная микросхема |
| КБ | — контрольный бит |
| КЛБ | — код левого байта слова 32-разрядного формата |
| КМ | — коммутатор магистральный |
| КМК | — коммутатор магистральный каналов |
| КНТУ | — комплект неавтономных тестов устройств |
| КОНСТ | — константа |
| КООП | — количество обращений в оперативную память |
| КОП | — код операции |
| КПБ | — код правого байта слова 32-разрядного формата |
| КПТО | — комплект программ технического обслуживания |
| КР | — контрольный разряд |
| КРОС | — программы расширения ОС (К — индекс разработчика) |
| КЧИ | — коммутатор числовой информации |
| ЛРИП | — логический обмен информацией между РИП и информационной магистралью М1 процессора |
| М1, М2, М3, М4, М5 | — информационные магистрали ЦП или каналов ввода-вывода |
| МАН | — магистраль анализов |
| МК | — мультиплексный канал |
| МЛ | — магнитная лента |
| МОК | — магистраль (информационная) общего канала |
| МПД | — мультиплексор передачи данных |
| МУ | — микропрограммное управление |
| МФЭ | — многофункциональный элемент |
| НК | — номер канала |
| НЛЖ ИНФ | — наложение информации |
| НМД | — накопитель на магнитных дисках |
| НМЛ | — накопитель на магнитной ленте |
| ОЗУ | — оперативное запоминающее устройство |
| ОК | — общий канал |
| ОП | — оперативная память |
| ОС | — операционная система |
| ОШ | — ошибка |
| ОШ ММХ УБ | — ошибка маски на магистрали X в байте Y |
| ОШ ЧМХ УБ | — ошибка числа на магистрали X в байте Y |
| ПА | — переадресация РОП (см. ВКЛ ПА) |
| ПАЛУ | — переносы АЛУ |
| ПЗУ | — постоянное запоминающее устройство |
| ПКЗ | — память ключей защиты |
| ПМК | — память мультиплексного канала |
| ПМП | — память микропрограмм |
| ПНЛ | — параметр нормализации в операциях с плавающей запятой |
| ППФЗ | — переполнение в операциях с фиксированной запятой |
| ПР | — признак результата |
| ПРА | — признак результата арифметических операций с фиксированной запятой |

| | |
|-------------|--|
| ПРАД | — признак результата в арифметических операциях с фиксированной запятой с операндами двойной длины |
| ПРАП | — признак результата арифметических операций с плавающей запятой |
| ПРАПД | — признак результата в арифметических операциях с плавающей запятой с операндами двойной длины |
| ПРВ | — признак ветвления (при выполнении анализов) |
| ПРИП | — обмен информацией между РИП и информационной магистралью М2 процессора с подменой полуслов |
| ПРЛ | — признак результата логических операций |
| ПРМА | — прием адреса |
| ПРСЛ | — признак сравнения логический |
| ПРСРА | — признак результата операций сравнения |
| ПСВ | — программные средства восстановления вычислительного процесса |
| ПСДВ | — параметр сдвига |
| ПСМ | — перенос сумматора |
| ПСО | — процессор статистической обработки |
| ПСЧ | — процессор случайных чисел |
| ПТД | — процессор телеобработки данных |
| ПУ | — пульт управления |
| РА | — регистр адреса |
| РАБ-А | — работа абонента |
| РАБ-К | — работа канала |
| РАВ | — регистр адреса возврата |
| РАВД | — регистр адреса возврата диагностический |
| РАВМ | — регистр адреса возврата мультиплексного канала |
| РАД | — регистр адреса данных |
| РАК | — регистр адреса канала |
| РАКБ | — регистр адреса команды буферный |
| РАМК | — регистр адреса текущей микрокоманды |
| РАОП | — регистр адреса оперативной памяти |
| РАП | — регистр адреса процессора |
| РАПК | — регистр адреса подканала |
| РАСТ | — регистр адреса старой (выполненной) микрокоманды |
| РАПУ | — регистр адреса пульта управления |
| РАУ | — регистр адреса устройства |
| РБ | — регистр буферный |
| РБСП | — регистр буферный связи с процессором |
| РВБ-А | — разрешение выборки абонента |
| РВБ-К | — разрешение выборки канала |
| РВИВЛ | — регистр выдачи информации на внешние линии |
| РВСИВЛ | — регистр выдачи синхроимпульсов на внешние линии |
| РГП | — регистровая память |
| РД | — регистр данных |
| РДШК | — регистр дешифрации команды |
| РИ | — регистр информации |
| РИП | — регистр информации процессора |
| РИ ОП | — регистр информации оперативной памяти |
| РИ ПУ | — регистр информации пульта управления |
| РКБ | — регистр количества байтов |
| РКД | — регистр коммутации данных |
| РКЗ | — регистр ключей защиты |
| РКОП | — регистр кода операции |
| РКС | — регистр константы сравнения |
| РМ | — регистр магистральный |
| РМК | — регистр микрокоманд |
| РМО | — регистр машинных ошибок |
| РМС | — регистр маски сравнения (загружается из поля РМС РМКЗ) |
| РМС1 (РМС2) | — регистр маски (загружается 32-разрядной константой из РМКЗ) |

| | |
|------------------|---|
| РНК | — регистр номера канала |
| РОН | — регистр общего назначения |
| РОНСТ1 (2) | — регистр общего назначения, адресуемый первым (вторым) счетчиком |
| РОП | — регистр операционный |
| РОС | — программы расширения ОС |
| РП | — регистр прерываний |
| РПБ | — регистр положения байта |
| РПЗ | — регистр с плавающей запятой |
| РПИВЛ | — регистр приема информации внешних линий |
| РПС | — регистр параметра сдвига |
| РСК | — регистр состояния канала |
| РПСИВЛ | — регистр приема синхроимпульсов внешних линий |
| РРД | — регистр результата деления |
| РСБФ | — регистр сравнения буферный |
| РСДВ | — регистр сдвига |
| РСДО | — регистр сдвига операционный |
| РССП | — регистр слова состояния программы |
| РСУ | — регистр состояния устройства |
| РФ | — регистр флажков (признаков) |
| РФП | — регистр фиксации прерываний |
| РФПД | — регистр фиксации прерываний дополнительный |
| РХК | — регистр хранения команды |
| С1, С2, С1', С2' | — основные синхросерии машины |
| СБРМ | — сброс магистралей |
| СДВ | — сдвиг |
| СК | — селекторный канал |
| СМ | — сумматор |
| СМС-К | — смена состояния канала |
| ССК | — слово состояния канала |
| ССП | — слово состояния программы |
| СТ | — счетчик |
| СТ1 РОН | — адресный регистр-счетчик РОН |
| (СТ2 РОН) | |
| СТР | — строб |
| СТЦ1 (СТЦ2) | — адресный регистр-счетчик РОП (счетчик циклов) |
| СУВВ | — специальное устройство ввода-вывода |
| ТБ | — триггер блокировки |
| ТБЛ НК | — триггер блокировки номера канала |
| ТВ/В | — триггер ввода-вывода |
| ТД | — триггер диагностики |
| ТЗ | — триггер загрузки |
| ТЗВ | — триггер запроса выборки |
| ТЗНАЧ | — триггер значимости |
| ТЗН1 | — триггер знака первый |
| ТЗН «+» | — триггер знака «+» |
| ТЗ ПКЗ | — триггер записи в ПКЗ |
| ТЗ ПРВ | — триггер запроса на выборку команды |
| ТЗ РХК | — триггер занятости РХК |
| ТИВ | — триггер интервалов времени |
| ТИПУ | — триггер инструкции ПРОВЕРИТЬ И УСТАНОВИТЬ |
| ТКД | — триггер команды ДИАГНОСТИКА |
| ТКЛ1 | — триггер символа класса I |
| ТЛН | — тест локализации неисправности |
| ТНЗ | — триггер начальной загрузки |
| ТНУТ | — триггер неуспеха теста |
| ТОВ | — триггер ошибки входа АЛУ (в каналах) |
| ТОМ | — триггер ошибки информационных магистралей канала |
| ТОР | — триггер операции редактирования |
| ТОЧП | — триггер очередности процессора |

| | |
|----------|--|
| ТОШ | — триггер ошибки |
| ТП ДМК | — триггер приема ДМК |
| ТПЗ | — триггер плавающей запятой |
| ТПИ | — триггер первой информации от абонента |
| ТП РМК | — триггер приема в РМК |
| ТПС | — триггер переноса сумматора |
| ТР | — триггер результата |
| ТРА | — табличный регистр анализа |
| ТРБ-А | — требование абонента |
| ТРАБК | — триггер работы канала |
| ТРАБП | — триггер работы процессора |
| ТРК | — триггер работы канала (в ОК) |
| ТРС | — триггер рабочего состояния процессора |
| ТФАК | — триггер фиксированного адреса канала |
| ТФАП | — триггер фиксированного адреса процессора |
| ТЭЗ | — типовой элемент замены |
| УВУ | — устройство управления ВУ |
| УПР-А | — управление от абонента |
| УПР-К | — управление от канала |
| УСК | — управляющее слово канала |
| УСКД | — управляющее слово команды ДИАГНОСТИКА |
| УУ | — устройство управления |
| УУ ПД | — узел управления передачей данных |
| ФА | — фиксированный адрес |
| ФЯ | — фиксированная ячейка ОП |
| ЦД | — цепочка данных |
| ЦК | — цепочка команд |
| ЦП | — центральный процессор |
| ЧТ | — чтение |
| ШИН-А | — информационные шины абонента |
| ШИН-К | — информационные шины канала |
| ШПУ | — шифратор пульта управления |
| ЭП | — элемент памяти |
| APR | — ALTERNATE PATH REPLY RETRY |
| CCH | — CANNEL CHECK HANDLER |
| DDR | — DYNAMIC DEVICE RECONFIGURATION |
| EREP | — ERROR RECORD EDITING AND PRINTING PROGRAM |
| ERP | — ERROR RECOVERY PROCEDURE |
| ERP1B | — ERROR RECOVERY PROCEDURE INTERFACE BLOCK |
| IBR | — INBOARD RECORD |
| IGFCCH63 | — имя модуля анализа состояния технических средств в момент ошибки селекторного канала |
| IGFCCH73 | — имя модуля анализа состояния технических средств в момент ошибки мультиплексного канала |
| MIC | — MISSING INTERRUPT CHECK |
| MCH | — MACHINE CHECK HANDLER |
| OBR | — OUTBOARD RECORD |
| SER | — SYSTEM ERROR RECOVERY |
| SEREP | — SEPARATE ERROR RECORD EDITING AND PRINTING PROGRAM |
| UCB | — UNIT CONTROL BLOCK |

1. Актуальные вопросы обеспечения надежности ЕС ЭВМ/В. В. Веригин, Б. И. Ермолаев, А. А. Кожарский, Б. П. Михайлин, И. Б. Михайлов, И. А. Попова, С. П. Соловьев, В. И. Яковлев. — Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 1, с. 112—122.
2. Аналоговые и цифровые интегральные схемы/Под ред. С. В. Якубовского. — М.: Сов. радио, 1979. 333 с.
3. Анисимов Б. В., Савельев А. Я. Основы конструирования и производства ЭВМ. — М.: Высшая школа, 1972. 276 с.
4. Ахмерова В. М., Касимова Д. А., Фадеев В. В. Программные средства восстановления вычислительного процесса в ОС ЕС после ошибок каналов ЭВМ ЕС-1033. — Вопросы судостроения. Сер. Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 54—65.
5. Баранов С. И. Синтез микропрограммных автоматов. — Л.: Энергия, 1979. 231 с.]
6. Байцер Б. Архитектура вычислительных комплексов. — М. Мир, 1974, т. 1. 498 с.; т. 2. 566 с.
7. Брик Е. А. Техника постоянных запоминающих устройств. — М.: Сов. радио. 1973. 240 с.
8. Брусиловский Е. Л., Кошман Е. А. Статистические методы расчета производительности вычислительных машин Единой системы. — Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 1, с. 122—141.
9. Букреев И. Н., Мансуров Б. М., Горячев В. И. Микроэлектронные схемы цифровых устройств. — М.: Сов. радио, 1975. 368 с.
10. Булей Г. Микропрограммирование. — М.: Мир., 1973. 126 с.
11. Валиев К. А., Кармазинский А. Н., Королев М. А. Цифровые интегральные схемы на МДП-транзисторах. — М.: Сов. радио, 1971. 384 с.
12. Волков А. Ф., Ведешников В. А., Зенкин В. Д. Автоматический поиск неисправностей в ЦВМ. — М.: Сов. радио, 1968. 151 с.
13. Выставкин Я. П. Сети обмена информацией между ЭВМ. — М.: Наука, 1975. 216 с.
14. Вычислительная система IBM-360. Принципы работы/Под ред. В. С. Штаркмана. — М.: Сов. радио, 1966. 440 с.
15. Данилович В. П. Программное обеспечение телеобработки данных в ЕС ЭВМ. — В кн.: Вычислительные средства в технике и системах связи. М.: Связь, 1978, вып. 3, с. 111—116.
16. Джермейн К. Программирование на IBM-360. — М.: Мир, 1974. 870 с.
17. Дроздов Е. А., Комарницкий В. А., Пятибратов А. П. Электронные вычислительные машины Единой системы. — М.: Машиностроение, 1976. 671 с.
18. Дроздов Е. А., Пятибратов А. П. Основы построения и функционирования вычислительных систем. — М.: Энергия, 1973. 368 с.
19. Единая система ЭВМ/Под общ. ред. А. М. Ларионова. — М.: Статистика, 1974. 164 с.
20. Иванов В. А., Палагин А. В., Писаревский А. В. Мини-ЭВМ: эволюция системы команд и их организация. — Управляющие системы и машины, 1974, № 1, с. 46—54.
21. Каган Б. М. Цифровые вычислительные машины и системы. — М.: Энергия, 1979. 528 с.
22. Каган Б. М., Мкртумян И. Б. Об одном подходе к проектированию систем автодиагностики ЦВМ. — Вопросы радиоэлектроники. Сер. ЭВТ, 1971, вып. 8, с. 42—51.
23. Каналы ввода-вывода ЭВМ ЕС-1022/Под ред. А. М. Ларионова. — М.: Статистика, 1976. 272 с.

24. Карцев М. А. Архитектура цифровых вычислительных машин. — М.: Наука, 1978. 293 с.
25. Катцан Г. Вычислительные машины системы 370. — М.: Мир, 1974. 508 с.
26. Клячко Э. И. Схемный и тестовый контроль автоматических цифровых вычислительных машин. — М.: Сов. радио, 1970. 192 с.
27. Козуб В. М. Система прерывания ЦВМ. — М.: Сов. радио, 1976. 200 с.
28. Коротышкин В. И. Аппаратно-микропрограммный способ преобразования кодов чисел в ЭВМ ЕС-1033. — Вопросы судостроения. Сер. ЭВМ, 1979, вып., 19, с. 27—33.
29. Лапин В. С., Семенихин В. С., Ларионов А. М. Средства и сети телеобработки данных в ЕС ЭВМ. — В кн.: Вычислительные средства в технике и системах связи. — М.: Связь, 1978, вып. 3, с. 79—95.
30. Ларионов А. М. Основные концепции развития ЕС ЭВМ. — В кн.: Вычислительная техника социалистических стран. — М.: Статистика, 1977, вып. 1, с. 41—51.
31. Логическая структура и принципы работы ЕС ЭВМ/А. М. Ларионов, В. К. Левин, С. П. Соловьев, А. П. Заморин. — Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 1, с. 3—18.
32. Малиновский Б. Н., Яковлев Ю. С., Маковенко Е. П. Анализ и синтез мини-ЭВМ. — Киев: Наукова думка, 1976. 178 с.
33. Наумов Б. Н. Создание СМ ЭВМ — новый этап развития средств вычислительной техники. — В кн.: Вычислительная техника социалистических стран. М.: Статистика, 1977, вып. 1, с. 84—97.
34. Нейман Л. И. Принцип организации микропрограммных тестов локализации неисправностей в каналах ЭВМ ЕС-1033. — Вопросы судостроения. Сер. Математические методы. Программирование. Эксплуатация ЭВМ, 1980, вып. 21, с. 55—59.
35. Пеледов Г. В., Райков Л. Д. Введение в ОС ЕС ЭВМ. — М.: Статистика, 1977. 119 с.
36. Пивовар Б. Система телеобработки, создаваемая в объединении МЕРА. — В кн.: Вычислительная техника социалистических стран. — М.: Статистика, 1978, вып. 4, с. 56—81.
37. Питерсон У., Уэлтон Э. Коды, исправляющие ошибки. — М.: Мир, 1976. — 594 с.
38. Поливода Е. О. Арифметический блок процессора ЭВМ ЕС-1033. Вопросы судостроения. Сер.: Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 17—27.
39. Принципы работы системы IBM-370/Под ред. Л. Д. Райкова. — М.: Мир, 1975. 576 с.
40. Принципы системной организации ЭВМ Единой системы/А. М. Ларионов, В. К. Левин, С. П. Соловьев, Е. Л. Брусиловский — Вопросы радиоэлектроники. Сер. ЭВТ, 1973, вып. 1, с. 55—65.
41. Иваницер Б. Д., Касимова Д. А., Крючкович Р. Я. Программные средства восстановления вычислительного процесса после машинных ошибок ЭВМ ЕС-1033. — Вопросы судостроения. Сер. Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 54—65.
42. Программные средства восстановления и диагностики в ОС ЕС для ЭВМ-1033/А. Н. Горбов, В. М. Ахмерова, Б. Д. Иваницер, Д. А. Касимова. — Тезисы докладов второй Всесоюзной конференции пользователей ЕС ЭВМ. Препринт НИИЭИР. М., 1979, с. 64—66.
43. Путинцев Н. Д. Аппаратный контроль управляющих цифровых вычислительных машин. — М.: Сов. радио, 1966. 420 с.
44. Радд У. Программирование на языке ассемблера и вычислительные системы IBM-360 и 370. — М.: Мир., 1979. 592 с.
45. Селлерс Р. Методы обнаружения ошибок в работе ЭЦВМ. — М.: Мир., 1972. 310 с.
46. Система документации Единой системы ЭВМ/Под общ. ред. А. М. Ларионова. — М.: Статистика, 1975. 327 с.

47. Система математического обеспечения ЕС ЭВМ/Под общ. ред. А. М. Ларионова. — М.: Статистика, 1974. 216 с.
48. Сорокин Г. П. Особенности микропрограммного управления блока диагностики ЭВМ ЕС-1033. — Вопросы судостроения. Сер. Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 39—44.
49. Сорокин Г. П. Особенности построения аппаратного контроля и диагностики процессора ЭВМ ЕС-1033. — Вопросы судостроения. Сер. Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 33—39.
50. Стэбли Л. Логическое программирование в «Системе 360». — М.: Мир, 1979. 752 с.
51. Уилкс М. Системы с разделением времени. — М.: Мир, 1972. 122 с.
52. Флорес А. Организация вычислительных машин. — М.: Мир, 1972. 428 с.
53. Функциональные возможности и опыт эксплуатации подсистемы планирования КРОС/А. И. Багавеева, Ю. С. Гаврилов, С. Н. Иванова, Т. В. Кондратьева, В. Б. Косалатов, Т. А. Марушева, Т. А. Петрова, Л. М. Урзова. — Вопросы судостроения. Сер.: Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 44—54.
54. Функциональные возможности многомашинной подсистемы ввода заданий для ЭВМ ЕС (Подсистема планирования РОС)/Е. Д. Беляева, Р. Г. Еналиев, В. Н. Марушев, В. Н. Мячин, Р. Г. Нургалеева, Ф. З. Рохлин, С. Н. Скворцова, Т. И. Фирсова, Л. М. Шайморданова, Ю. Р. Шарафи, Е. Ф. Шакирова. — Вопросы судостроения. Сер. Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 65—73.
55. Функциональная структура OS/360. — М.: Сов. радио, 1971. 88 с.
56. Хассон С. Микропрограммное управление. — М.: Мир, 1974; вып. 1. 240 с.; вып. 2. 477 с.
57. Цикритизис Д., Бернштейн Ф. Операционные системы. — М.: Мир, 1977. 336 с.
58. Чу Я. Организация ЭВМ и микропрограммирование. — М.: Мир, 1975. 592 с.
59. Шелихов А. А. Единая система ЭВМ. — М.: Энергия, 1975. 176 с.
60. Электронная вычислительная машина ЕС-1020/Под общ. ред. А. М. Ларионова. — М.: Статистика, 1975. 301 с.
61. Электронная вычислительная машина ЕС-1030/Под общ. ред. А. М. Ларионова. — М.: Статистика, 1977. 255 с.
62. Электронная вычислительная машина ЕС-1050/Под ред. А. М. Ларионова. — М.: Статистика, 1977. 256 с.
63. Ярмухаметов А. У. Микропрограммное управление в ЭВМ ЕС-1033. — Вопросы судостроения. Сер.: Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 3—11.
64. Ярмухаметов А. У. Организация местной и сверхоперативной памяти процессора ЕС-2433 и микропрограммное управление ими. — Вопросы судостроения. Сер. Математические методы. Программирование. Эксплуатация ЭВМ, 1979, вып. 19, с. 11—17.
65. State and Prospects of Development of EC-1033 Computer System. — World Science News, 1978, Vol. XV, No. 43, pp. 12—15.
66. State and Prospects of Further Development of Software Computer — Model EC-1033. — World Science News, Vol. XV, No. 43, pp. 17—18.

ОГЛАВЛЕНИЕ

| | |
|---|------------|
| Предисловие | 3 |
| Глава 1. Общие сведения об ЭВМ ЕС-1033 и ее структурные особенности | 7 |
| 1.1. Состав и характеристики | 7 |
| 1.2. Магистральная организация центрального процессора и каналов ввода-вывода | 12 |
| 1.3. Микропрограммное управление | 17 |
| 1.4. Средства контроля и диагностики | 19 |
| 1.5. Элементная база | 21 |
| 1.6. Система синхронизации | 30 |
| 1.7. Постоянная память микропрограмм | 32 |
| 1.8. Конструкция машины и система электропитания | 35 |
| 1.9. Оперативная память | 37 |
| 1.10. Программное обеспечение | 43 |
| Глава 2. Операционная часть центрального процессора | 45 |
| 2.1. Структура процессора | 45 |
| 2.2. Блок магистрального коммутатора | 53 |
| 2.3. Местная память | 60 |
| 2.4. Арифметико-логический блок | 68 |
| 2.5. Блок сдвига информации | 75 |
| 2.6. Блок преобразования чисел из одной системы счисления в другую | 86 |
| Глава 3. Управление в центральном процессоре | 92 |
| 3.1. Блок управляющих регистров | 92 |
| 3.2. Система прерываний и приоритетов | 95 |
| 3.3. Средства прямого управления | 97 |
| 3.4. Управление оперативной памятью | 101 |
| 3.5. Блок микропрограммного управления | 112 |
| 3.6. Пульт управления | 128 |
| Глава 4. Служебные и обслуживающие микропрограммы | 137 |
| 4.1. Микропрограммы пультовых процедур | 138 |
| 4.2. Обслуживание прерываний, корректировка таймера | 140 |
| 4.3. Выборка команд | 144 |
| Глава 5. Команды управления | 150 |
| 5.1. Переходы | 150 |
| 5.2. Переключения состояний | 155 |
| 5.3. Ввод-вывод | 160 |
| Глава 6. Команды обработки данных с фиксированной запятой | 160 |
| 6.1. Загрузка, сложение, сравнение, запись в память, сдвиг | 161 |
| 6.2. Умножение | 165 |
| 6.3. Деление | 170 |

| | |
|---|------------|
| Глава 7. Команды обработки данных с плавающей запятой | 172 |
| 7.1. Основные характеристики | 172 |
| 7.2. Загрузка, пополам, сложение, вычитание, сравнение | 173 |
| 7.3. Умножение | 178 |
| 7.4. Деление | 185 |
| Глава 8. Команды логической обработки | 194 |
| 8.1. Обработка операндов фиксированной длины | 194 |
| 8.2. Обработка операндов переменной длины | 198 |
| 8.3. Перекодирование | 203 |
| 8.4. Редактирование | 205 |
| 8.5. Сдвиги | 210 |
| Глава 9. Команды обработки десятичных данных | 210 |
| 9.1. Сложение, вычитание, сравнение | 211 |
| 9.2. Умножение | 219 |
| 9.3. Деление | 223 |
| 9.4. Упаковка, распаковка, пересылка | 227 |
| Глава 10. Средства контроля и диагностики центрального процессора | 234 |
| 10.1. Общие принципы организации контроля и диагностики | 234 |
| 10.2. Регистрация машинных ошибок | 237 |
| 10.3. Диагностические процедуры | 239 |
| 10.4. Микропрограммное управление в блоке диагностики | 243 |
| 10.5. Тесты локализации неисправности | 249 |
| 10.6. Команда ДИАГНОСТИКА | 254 |
| Глава 11. Организация ввода-вывода | 255 |
| 11.1. Общие сведения | 255 |
| 11.2. Интерфейс ввода-вывода | 258 |
| 11.3. Команды ввода-вывода | 259 |
| 11.4. Состояние системы ввода-вывода | 263 |
| 11.5. Выполнение операций ввода-вывода. Прерывания | 266 |
| Глава 12. Каналы ввода-вывода | 270 |
| 12.1. Организация каналов | 270 |
| 12.2. Общий канал | 271 |
| 12.3. Мультиплексный канал | 283 |
| 12.4. Селекторный канал | 294 |
| Глава 13. Средства контроля и диагностики каналов ввода-вывода | 305 |
| 13.1. Пульт управления | 305 |
| 13.2. Аппаратурный контроль | 309 |
| 13.3. Организация диагностики каналов | 311 |
| 13.4. Тесты локализации неисправности | 316 |
| 13.5. Программные тесты | 318 |
| Глава 14. Программные средства восстановления вычислительного процесса | 321 |
| 14.1. Уровни восстановления | 321 |
| 14.2. Восстановление после ошибок ввода-вывода | 324 |
| 14.3. Восстановление после машинных ошибок | 329 |
| 14.4. Программы редактирования и печати записей об ошибках | 332 |

Глава 15. Тестовые программные средства 333

- 15.1. Состав 333
- 15.2. Моделезависимые тесты 335
- 15.3. Моделеориентированные тесты 338
- 15.4. Управляющая и обслуживающая программы 340
- 15.5. Комплект неавтономных тестов устройств 341
- 15.6. Использование тестовых программных средств при наладочных работах 342

Глава 16. Программы, расширяющие операционную систему. 343

- 16.1. Подсистема планирования КРОС 343
- 16.2. Обработка заданий под управлением КРОС 347
- 16.3. Подсистема планирования РОС 350
- 16.4. Обработка заданий под управлением РОС 353

Приложения 357

Список литературы 378

ИБ № 3640

ЭЛЕКТРОННАЯ ВЫЧИСЛИТЕЛЬНАЯ МАШИНА ЕС-1033

**Герман Петрович СОРОКИН, Виктор Николаевич ИВАНОВ,
Азат Усманович ЯРМУХАМЕТОВ и др.**

Редактор Т. В. Абизова
Художественный редактор С. С. Водчиц
Технический редактор Е. П. Смирнова
Корректоры Л. В. Асташенок и В. Е. Блохина
Оформление художника Р. Р. Кожанова

Сдано в набор 15.12.81. Подписано в печать 26.11.82. Т-21412.
Формат 60×90^{1/16}. Бумага типографская № 2
Гарнитура литературная. Печать высокая. Усл. печ. л. 24;0.
Уч.-изд. л. 27,0. Тираж 20 000 экз. Заказ 786. Цена 1 р. 60 к.

Ордена Трудового Красного Знамени издательство «Машиностроение»
107076, Москва, Стромынский пер., д. 4

Ленинградская типография № 6 ордена Трудового Красного Знамени
Ленинградского объединения «Техническая книга» им. Евгении Соколовой
Союзполиграфпрома при Государственном комитете СССР
по делам издательств, полиграфии и книжной торговли.
193144, г. Ленинград, ул. Моисеенко, 10.