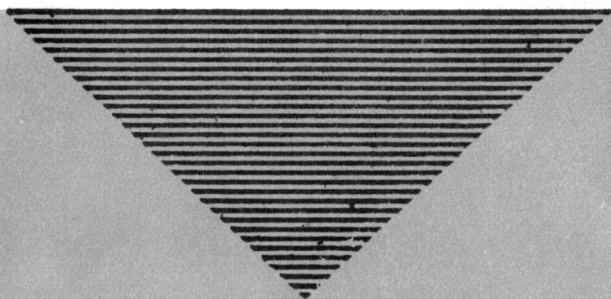


ПРОЦЕССОР



ЭВМ

ЕС-1020

ПРОЦЕССОР ЭВМ ЕС-1020

Под общей редакцией А. М. ЛАРИОНОВА

ХВ



МОСКВА «СТАТИСТИКА» 1975

*Р. М. Асцатуров, В. П. Качков, А. П. Кондратьев,
В. М. Ленкова, Н. А. Мальцев, Р. Б. Пашковская*

Процессор ЭВМ ЕС-1020

Редактор *Г. М. Клебче*
Техн. редактор *Р. Н. Феоктистова*, Корректор *О. Г. Шумская*
Худ. редактор *Т. В. Стихно*
Обложка художника *Л. С. Эрмана*

Сдано в набор 25/II 1975 г. Подписано к печати 30/VII 1975 г. Формат бумаги 60×90^{1/16}.
Бумага № 1. Объем: 10 печ. л. Уч.-изд. л. 10,88. Усл. п. л. 10. Тираж 52 000 экз. А 03525.
(Тематич. план 1975 г. № 77). Заказ № 2256. Цена 58 коп.

Издательство «Статистика», Москва, ул. Кирова, 39.

Областная типография управления издательств, полиграфии и книжной торговли
Ивановского облисполкома, г. Иваново-8, ул. Типографская, 6.

П84 **Процессор ЭВМ ЕС-1020.** Под общ. ред. А. М. Ларионова. М., «Статистика», 1975.

160 с. с ил.

На обороте тит. л. авт.: Р. М. Асцатуров, В. П. Качков, А. П. Кондратьев, В. М. Ленкова, Н. А. Мальцев, Р. Б. Пашковская.

В книге рассмотрены основные принципы работы процессора одной из моделей Единой системы ЭВМ—ЕС-1020.

Дано описание структуры, состава и функциональных схем отдельных устройств и блоков процессора, а также процедур обработки информации и организации работы процессора.

Книга предназначена для работников вычислительных центров, может быть также полезна студентам вузов и инженерам, занимающимся конструированием вычислительных машин.

П $\frac{30502-119}{008(01)-75}$ 77-75

6Ф7.3

Глава 1

ОСНОВНЫЕ СВЕДЕНИЯ О ПРОЦЕССОРЕ ЭВМ ЕС-1020

Широкое внедрение в народное хозяйство страны электронных вычислительных машин, и особенно машин третьего поколения, выдвигает задачу организации обслуживания и подготовки квалифицированных кадров.

Предлагаемая книга должна помочь в подготовке специалистов по эксплуатации машины ЕС-1020 и организации работ как на одной машине, так и в многопроцессорных системах. Она продолжает серию книг по Единой системе ЭВМ: «Единая система ЭВМ» и «Электронная вычислительная машина ЕС-1020» [4, 12].

Материал, вошедший в книгу «Процессор ЭВМ ЕС-1020», содержит общие сведения о процессоре ЭВМ ЕС-1020 (глава 1), описания отдельных блоков процессора (главы 2, 3, 4, 5, 6), системы прерываний (глава 7), принципов и системы контроля (глава 8), а также описание пульта оператора и сведения о мультисистемной работе (главы 9, 10).

1.1. Назначение и состав процессора

Процессор ЕС-2020 является центральной частью ЭВМ ЕС-1020 и предназначен для организации работы всей модели.

В процессоре применен микропрограммный принцип построения устройства управления, заключающийся в выработке управляющих сигналов путем последовательного считывания и расшифровки микрокоманд, расположенных в ячейках постоянной памяти (ПП).

Микрокоманда — это совокупность кодов совместимых микроопераций, управляющих процессором в течение одного машинного такта.

Микрооперация представляет собой элементарную операцию, которая может быть выполнена при подаче одного импульса на некоторую управляющую шину, например сброс регистра на ноль, передача информации из одного регистра в другой и т. д.

Совместимыми называются микрооперации, которые можно выполнять одновременно без искажения информации и которые могут быть закодированы в одной микрокоманде.

Последовательность микрокоманд, выполняющих одну процедуру, представляет собой микропрограмму, записанную в ПП.

Все процедуры как служебные (например, гашение системы), так и связанные с обработкой информации осуществляются с помощью микропрограмм, реализующих алгоритмы выполнения этих процедур.

При выполнении любой процедуры должен быть задан начальный адрес микропрограммы, реализующей данную процедуру, и далее сформированы адреса последующих микрокоманд.

По заданным адресам осуществляется чтение микрокоманды из постоянной памяти. Далее производится расшифровка микрокоманды и выработка сигналов, управляющих работой блоков процессора, которые реализуют выполнение заданной процедуры. При этом могут быть выполнены запись или чтение в память машины информации, подлежащей обработке, с предварительным определением адреса памяти, обработка информации в специальном блоке, установка ряда триггеров, управляющих режимами работы процессора, в нулевое или единичное состояние, а также определение адреса следующей микрокоманды. Все действия должны выполняться в строгой последовательности, т. е. микрооперации должны быть синхронизованы.

Таким образом, процессор должен содержать средства адресации постоянной и оперативной памяти, выборки и запоминания информации, арифметической и логической обработки данных и синхронизации работы отдельных блоков и узлов, а также установления связи между памятью и внешними устройствами (ВУ) (рис. 1.1). Связь между памятью и ВУ осуществляется через ка-

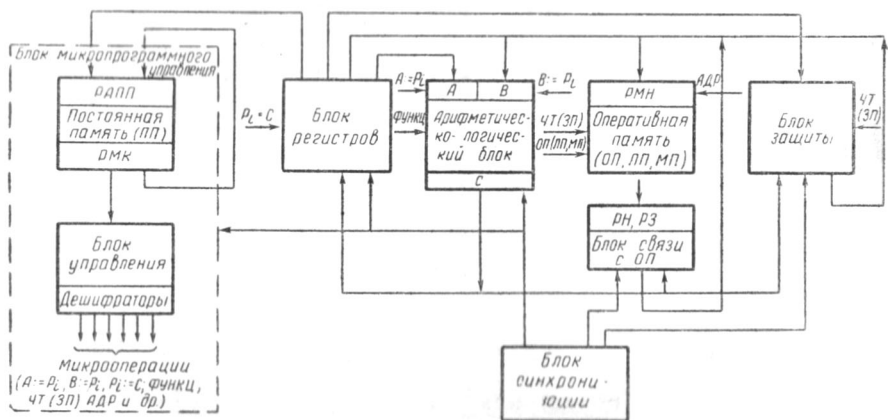


Рис. 1.1. Структура процессора ЕС-200

налы, интерфейс ввода-вывода и устройства управления ВУ (УВУ).

Процессор включает следующие блоки:

Блок синхронизации предназначен для выработки синхронимпульсов, обеспечивающих синхронную работу всех блоков процессора.

Блок микропрограммного управления обеспечивает выработку микроопераций, управляющих работой арифметическо-логического блока, выборкой и записью информации в оперативную память, выполнением процедур прерывания, обслуживания каналов и пульта. В состав этого блока входят постоянная память (ПП) с информационным (РМК) и адресным (РАПП) регистрами, предназначенная для хранения микропрограмм, и блок управления, вырабатывающий микрооперации и формирующий адреса следующей микрокоманды.

Блок регистров осуществляет хранение и выдачу различной информации, участвующей в процессе обработки программы. Он содержит адресные регистры, регистры общего пользования, служебные регистры, узел формирования адреса оперативной памяти, узлы формирования прерываний по защите и адресации, машинному сбою, а также узлы контроля.

Арифметическо-логический блок предназначен для побайтовой обработки информации, а также передачи данных из одного машинного регистра в другой. Блок состоит из ряда узлов, обеспечивающих выполнение заданного набора микроопераций.

Оперативная память служит для хранения последовательности команд, операндов и информации, необходимой для обеспечения работы системы. В состав оперативной памяти входят три логически самостоятельных типа памяти:

основная память (ОП) объемом 64К, 128К или 256К байт,

локальная память (ЛП) объемом 256 байт,

мультиплексная (МП) память объемом 768 или 1536 байт.

Все типы памяти имеют одинаковую разрядность. Информационным регистром для всех типов памяти является регистр НЗ, а при обращении ко всем типам памяти используется адресный регистр МН.

В состав локальной памяти входят 16 регистров общего назначения (длиной 4 байта), в которых хранятся операнды, адреса и индексы, и четыре регистра для операндов с плавающей запятой (длиной 8 байт), в которых хранятся числа с плавающей запятой.

Блок связи с оперативной памятью содержит информационные регистры (РН, РЗ), регистры-дублиеры (РН1, РН2), узлы занесения информации в РН, РЗ, узлы контроля и индикации.

Блок защиты защищает содержимое основной памяти от искажения или неправильного использования. Он состоит из запоминающего устройства ключей памяти, регистра блока защиты, узла сравнения, выдающего сигнал при несоответствии ключа памяти ключу защиты.

1.2. Выполнение команды

Выполнение команды, хранящейся в основной памяти, представляет собой замкнутый цикл, состоящий из двух фаз: фазы выборки команды и фазы выполнения.

Фаза выборки. Основное назначение этой фазы состоит в чтении байт команды из основной памяти, формировании адресов операндов, установке признаков выбранной команды и формировании начального адреса микропрограммы реализации фазы выполнения. В некоторых случаях в фазе выборки команды осуществляются чтение операндов и анализ правильности задания адресов.

Микропрограмма реализации фазы выборки команды имеет несколько ветвей, алгоритмы которых определяются типом команды.

Существует пять основных типов команд, обозначаемых форматными кодами, определяющими тип выполняемой операции: код *RR* обозначает операцию типа «регистр — регистр»; коды *RX*, *RS* — операции типа «регистр — память»; код *SI* — операцию, когда один операнд находится в памяти, а другой — в самой команде; код *SS* — операцию типа «память — память» (рис. 1.2).

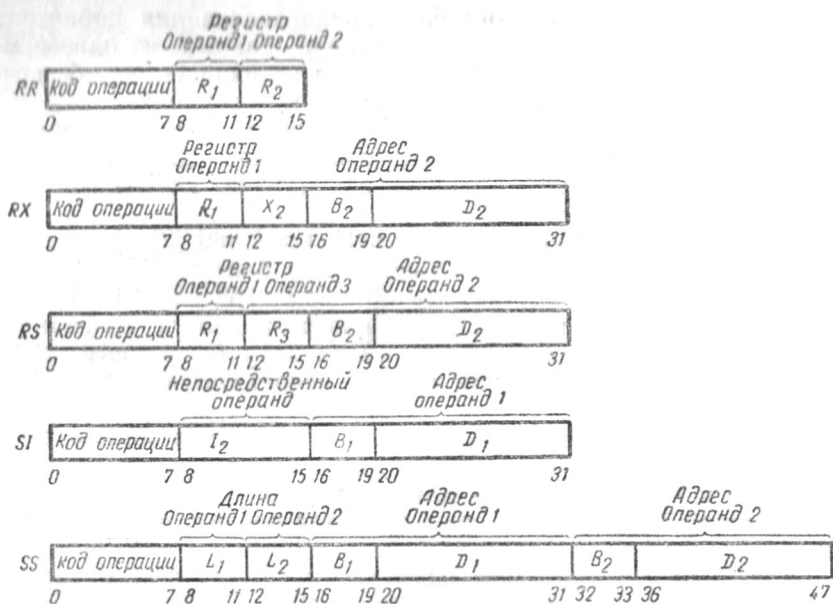


Рис. 1.2. Пять основных форматов команд

Для описания выполнения операции операнды нумеруются: первый операнд, второй операнд и третий операнд, в том случае, если есть команда ПЕРЕХОД ПО ИНДЕКСУ. Номера указывают

порядок, в котором операнды участвуют в операции. Операнд, на который ссылаются в форматном коде, обычно обозначается номером, следующим за буквенным обозначением поля, например R_1 , B_1 , L_2 , D_2 .

В каждом формате первое полуслово состоит из двух частей. Первый байт содержит код операции. Длина и формат команды определяются значениями первых двух бит кода операции:

Биты 0—1	Длина команды в полусловах	Форматный код
00	Одно	RR
01	Два	RX
10	Два	RX или SI
11	Три	SS

Второй байт используется либо в виде двух четырехбитовых полей, либо одного восьмибитового поля. Этот байт может содержать следующую информацию: четырехбитовый адрес регистра, хранящего операнд (R_1 , R_2 или R_3), или индекс (X_2), четырехбитовую маску (M_1), четырехбитовый код длины операнда (L_1 или L_2), восьмибитовый код длины операнда (L), восьмибитовый байт, представляющий собой сам операнд (I_2), т. е. непосредственно данные.

В некоторых командах четырехбитовое поле или весь второй байт в первой половине слова игнорируется.

Второе и третье полуслова содержат всегда четырехбитовый адрес регистра базы (B_1 или B_2) и следующее за ним 12-битовое смещение SM_1 или SM_2 .

Для легкого перемещения сегментов программ и обеспечения гибкого задания входных, выходных и рабочих областей основной памяти все команды, в которых производится обращение к основной памяти, могут использовать полные адреса. Полный адрес обращения к основной памяти формируется из базового адреса, индекса и смещения.

Базовый адрес B представляет собой 24-битовое число, находящееся в общем регистре, задаваемом с помощью поля B команды (поле B участвует в формировании всех адресов). Базовый адрес может быть использован как средство для статического перемещения программ и данных. Он служит также для указания положения массива и идентификации записей при их обработке. С помощью этого адреса можно адресовать всю основную память. Он также может быть использован для индексации.

Индекс X представляет собой 24-битовое число, находящееся в общем регистре, задаваемом в команде с помощью поля X . Индекс входит в состав адреса только для команд в формате RX . Он может быть использован для адресации отдельного элемента массива. Таким образом, при употреблении формата RX допускается двойная индексация.

Смещение SM представляет собой 12-битовое число, располагаемое в формате команды. Смещение участвует в формировании каждого адреса, позволяет выполнять относительную адресацию

до 4095 байт, следующих по порядку за байтом, заданным базовым адресом, или байтом, адрес которого получен индексацией базового адреса. При работе с массивами смещение может быть использовано для указания одного из нескольких компонентов элемента массива, при обработке записей — для идентификации компонента внутри записи.

При формировании адреса базовый адрес и индекс рассматриваются как 24-битовые положительные двоичные числа без знака. Смещение рассматривается как 12-битовое положительное двоичное число. Все три компонента складываются как 24-битовые двоичные числа без учета переполнения. Так как в формировании любого адреса участвует базовый адрес, сумма всегда имеет 24 бита. Биты адреса нумеруются числами 8—31 в соответствии с нумерацией бит базового адреса и индекса в регистре общего назначения.

В программе могут встречаться нулевые базовые адреса, индексы и смещения, которые указывают на отсутствие соответствующего компонента адреса. Нулевое содержимое поля базового адреса или поля индекса означает, что при формировании адреса должно быть использовано нулевое значение соответствующего компонента, а не содержимое регистра общего назначения с номером 0.

Установка признаков команды необходима в случаях, когда микропрограмма фазы выполнения реализует выполнение нескольких команд. Начальный адрес микропрограммы фазы выполнения формируется по коду выбранной команды.

Фаза выполнения различных команд реализуется соответствующей микропрограммой, причем часто одна микропрограмма реализует различные команды, например СЛОЖИТЬ и ВЫЧЕСТЬ, операции И, ИЛИ и т. п. Как правило, в одной микропрограмме могут быть реализованы алгоритмы выполнения различных команд с операндами одинакового типа. Это естественно, ибо алгоритмы обработки чисел, представленных в различном виде, иногда значительно отличаются друг от друга. Так, например, алгоритм сложения чисел с фиксированной запятой, представленных в дополнительном коде, отличается от сложения чисел с плавающей запятой, представленных в прямом коде.

В ЭВМ ЕС-1020 обработке подлежат:

числа с фиксированной запятой длиной 4 байта, представленные в дополнительном коде;

числа с плавающей запятой длиной 4 или 8 байт, мантисса которых представлена в прямом коде. Характеристика мантиссы получена путем сложения порядка со знаком и числа 64, а знак мантиссы хранится в нулевом (старшем) бите числа;

десятичные числа, представленные в коде 8421.

При выполнении арифметических операций и операций преобразования систем счисления используются «упакованные» числа (по две цифры в каждом байте). Знак числа занимает четыре бита

и хранится при этом в правой тетраде младшего байта. Число представляется в прямом коде и имеет переменную длину.

Логическая информация может иметь как фиксированную, так и переменную длину. Большая часть логической информации состоит из кодов букв и чисел, используемых в устройствах ввода-вывода.

Все особенности обработки перечисленных типов информации учтены при разработке алгоритмов процедур.

Выполнение микрокоманды. В постоянной памяти имеются микропрограммы фазы выборки команд всех форматов и микропрограммы выполнения отдельных или группы однотипных команд, а кроме того микропрограммы прерываний, обслуживания каналов ввода-вывода, выполнения пультовых и других процедур.

При выполнении микропрограммы микрокоманды считываются в РМК из постоянной памяти по адресу, находящемуся в регистре адреса постоянной памяти (РАПП). Адрес микрокоманды может быть занесен с пульта или получен автоматически при выполнении предыдущей микрокоманды. Группы бит РМК возбуждают дешифраторы, вырабатывающие набор микроопераций, которые управляют работой отдельных узлов машины.

При выполнении микрокоманды одновременно могут производиться следующие основные действия:

1) запись или чтение информации из оперативной памяти по адресу, расположенному в адресном регистре МН (РМН), который может получать информацию из блока регистров и каналов. Режим работы ОП, а также источник информации для РМН задается в микрокоманде (микрооперации ЧТ, ЗП и АДР на рис. 1.1).

В режиме записи по заданному адресу в оперативную память записываются два байта информации, расположенных в информационных регистрах памяти РН и РЗ. В режиме чтения из ячейки памяти, определенной содержимым РМН, в РН и РЗ считываются два байта информации;

2) обработка операндов в арифметическо-логическом блоке (БА). При этом на входы А и В БА из любого регистра процессора, из РМК, а также из каналов может быть подана информация, над которой производится арифметическая или логическая операция.

Информация с выхода С блока может быть подана в любой регистр процессора, а также в канал. Источники информации на входы А и В и приемник информации с выхода С, а также тип операции задаются в микрокоманде (микрооперации) $A: = P_i$; $B: = P_i$; $P_i = C$; ФУНКЦИЯ;

3) установка в нулевое или единичное состояние одного или нескольких триггеров процессора.

Кроме того, в каждой микрокоманде формируются адреса следующей микрокоманды. Микрооперации, задаваемые микрокомандой, выполняются в последовательности, определяемой тактовыми импульсами, вырабатываемыми блоком синхронизации.

1.3. Выполнение программы

Обычно работой ЦП управляют команды, выбираемые из последовательно расположенных ячеек основной памяти, определяемых значением текущего адреса команды. Для получения адреса следующей команды адрес выполненной команды увеличивается на число единиц, равное числу байт в этой команде. Затем выполняется очередная команда, и весь процесс повторяется с новым значением адреса команды.

Можно считать, что все полуслова, из которых состоит команда, выбираются из памяти после завершения выполнения предыдущей операции и до выполнения текущей, даже если физический размер слова в памяти и совмещение выполнения команды с обращением к памяти могут повлиять на выборку команды в действительности. Поэтому модифицировать команды в памяти можно непосредственно предшествующей командой.

Обычная последовательность операций может нарушаться при переходе, переключении состояния, прерывании или вмешательстве оператора.

Переходы. В тех случаях, когда обычная последовательность команд нарушается при обращениях к подпрограммам, разветвлении программ или циклическом выполнении некоторого участка программы используются команды передачи управления.

Связь с подпрограммами осуществляется так, что позволяет не только ввести адрес новой команды, но и сохранить адрес возврата и всю необходимую для возврата информацию.

Обычно ветвление обеспечивается командой **УСЛОВНЫЙ ПЕРЕХОД**, проверяющей двухбитовый признак результата (код условия), который отражает результат большинства арифметических и логических операций и операций ввода-вывода.

Для некоторых часто встречающихся комбинаций арифметических действий и проверок предусмотрены команды **ПЕРЕХОД ПО СЧЕТЧИКУ** и **ПЕРЕХОД ПО ИНДЕКСУ**. Такие переходы в определенных случаях значительно удобнее и производительнее.

Слово состояния программы. Информацию, необходимую для выполнения программы, содержит слово состояния программы (ССП), представляющее собой двойное слово. В ССП находятся адрес команды, признак результата и другая информация, назначение которой будет рассмотрено ниже. Общее назначение ССП — управление порядком выборки команд, а также фиксация состояния системы по отношению к выполняемой программе.

Прерывания. Система прерываний позволяет ЦП изменять свое состояние при появлении внешних по отношению к системе сигналов, а также сигналов, поступающих из устройств ввода-вывода или возникающих в самом ЦП. Возможны пять классов прерываний: прерывания от ввода-вывода, программные прерывания, прерывания при обращении к супервизору, внешние прерывания и прерывания от схем контроля машины.

Глава 2

БЛОК СИНХРОНИЗАЦИИ

2.1. Назначение и состав блока

ЭВМ ЕС-1020 основана на синхронном принципе работы, заключающемся в том, что выполнение всех действий (прием данных во входные регистры БА и результата операции БА в регистры процессора или каналов, переключение отдельных триггеров и т. д.) в вычислительном устройстве и каналах, а также запуск запоминающих устройств (оперативной памяти, постоянной памяти, памяти ключей) синхронизируется периодическими импульсами, называемыми тактовыми.

Блок синхронизации предназначен для выработки серии тактовых импульсов и распределения их по блокам процессора.

Периодичность тактовых импульсов — длительность такта T выбирается равной времени ожидания (полуцикла) ОП и составляет 1000 нс.

Блок синхронизации состоит из узла формирования основных и рабочих тактовых импульсов и узла распределения синхроимпульсов.

2.2. Узел формирования основных и рабочих тактовых импульсов

Для синхронизации действий, которые могут происходить в такте в различное время, в течение T вырабатываются четыре тактовых импульса.

Тактовые импульсы разных разновидностей вырабатываются из основных тактовых импульсов $C1, C2, C3, C4$, имеют длительность 200 нс и чередуются обычно через 250 нс, т. е. между соседними импульсами образуется «зазор» в 50 нс, который в некоторых случаях между импульсами $C1$ и $C2$ увеличивается до 100 нс (рис. 2.1).

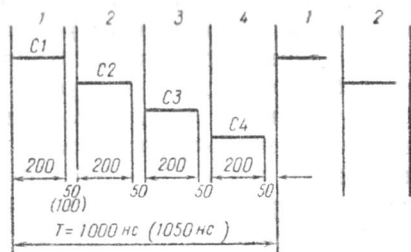


Рис. 2.1. Диаграмма выработки тактовых импульсов

Основные тактовые импульсы С1—С4 для синхронизации обычно не используются, а служат для выработки рабочих тактовых импульсов, непосредственно применяемых для синхронизации выполнения действий и запуска запоминающих устройств. Только в некоторых случаях основные тактовые импульсы используются как синхронизирующие.

При выработке рабочих тактовых импульсов можно выделить два режима работы процессора:

первый — работа ЦП или каналов происходит под управлением блока микропрограммного управления (при этом выполняются микропрограммы процессора или микропрограммы каналов);

второй — работа селекторных каналов (КС) происходит под управлением собственной аппаратуры (при этом осуществляется передача данных между ОП и КС, а выполнение микропрограмм приостанавливается).

В первом режиме работы вырабатываются рабочие тактовые импульсы, называемые импульсами такта ТИ (ТИ1, ТИ2, ТИ3, ТИ4). Эти импульсы используются для синхронизации всех действий (микроопераций), задаваемых микрокомандами, каждая из которых выполняется за один такт ТИ.

Во втором режиме, если хотя бы один из селекторных каналов выдает запрос на обмен данными с ОП, который осуществляется за два или три такта, выработка импульсов ТИ1—ТИ4 блокируется, и в течение этих двух или трех тактов вырабатываются рабочие импульсы такта СИ (СИ1, СИ2, СИ3, СИ4). По этим импульсам происходят выдача адреса и команд в ОП из КС, а также выполнение таких действий в каналах, как модификация регистров, адреса и счетчика байт, переключение управляющих триггеров и т. д.

Итак, при работе блока микропрограммного управления вырабатываются тактовые рабочие импульсы ТИ1—ТИ4, а при выполнении тактов обмена между ОП и КС — более приоритетные импульсы СИ1—СИ4.

Кроме этих двух серий рабочих тактовых импульсов (ТИ, СИ), могут вырабатываться еще импульсы холостого такта ХИ (ХИ1, ХИ2, ХИ3, ХИ4). Эти импульсы вырабатываются обычно в течение одного такта, если в предыдущем такте ТИ была обнаружена необычная или сбойная ситуация (например, произошло обращение к блоку ОП, отсутствующему в данной модели, появился запрос от каналов на управление от блока микропрограммного управления или схема контроля обнаружила сбой аппаратуры процессора). В такте ХИ микрокоманды не выполняются и он служит лишь для организации перехода к микропрограмме обработки необычной или сбойной ситуации (в такте ХИ формируется начальный адрес одной из этих микропрограмм и выбирается микрокоманда по сформированному адресу). При одновременном наличии нескольких причин может быть и больше одного такта ХИ

подряд. Такт XI выше по приоритету такта TI, но ниже СИ: при выработке импульсов серии СИ блокируется выработка импульсов TI и XI, а при выработке импульсов XI блокируются импульсы TI.

Параллельно с импульсами этих серий всегда вырабатываются также главные импульсы (ГИ1, ГИ2, ГИ3, ГИ4).

Основные тактовые импульсы С1, С2, С3, С4 присутствуют после включения машины всегда, а главные, как и импульсы тактов TI, СИ, XI, блокируются, если процессор при сбое аппаратуры переходит в так называемый тяжелый останов. Главные импульсы никаких особых функций не несут, они введены лишь для удобства работы (например, если какое-то действие выполняется как по TI2, так и по XI2 или СИ2, то вместо этих трех импульсов можно использовать ГИ2).

Рабочие тактовые импульсы, используемые для синхронизации выполнения действий и запуска запоминающих устройств (ТИ1—ТИ4, СИ1—СИ4, ХИ1—ХИ4, ГИ1—ГИ4), вырабатываются из основных импульсов С1—С4 по сигналам соответствующих тактов: TI, СИ, ХИ, ГИ.

Для выработки сигналов тактов TI, СИ, ХИ, ГИ используется регистр тактов (рис. 2.2), причем для каждого такта в регистре имеются два триггера, разрешающих соответственно выработку первых двух (ТИ1 и ТИ2, СИ1 и СИ2, ХИ1 и ХИ2, ГИ1 и ГИ2)

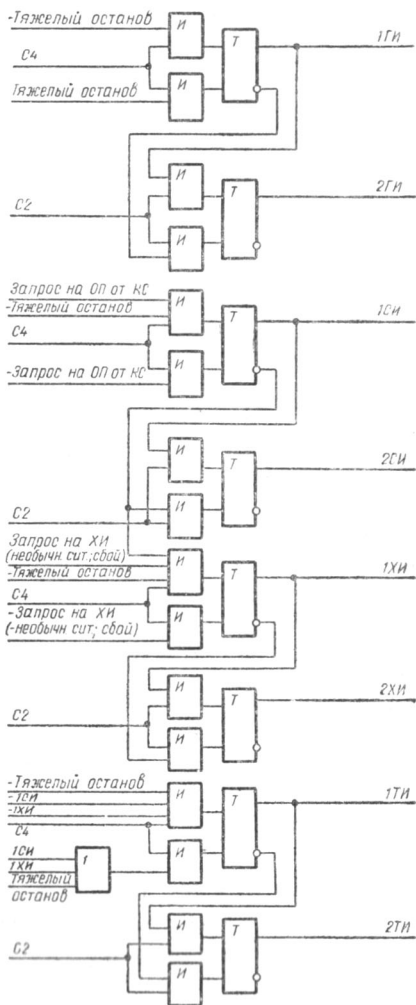


Рис. 2.2. Схема регистра тактов

и двух последних (ТИ3 и ТИ4, СИ3 и СИ4, ХИ3 и ХИ4, ГИ3 и ГИ4) импульсов такта.

Временная диаграмма выработки рабочих и главных тактовых импульсов приведена на рис. 2.3.

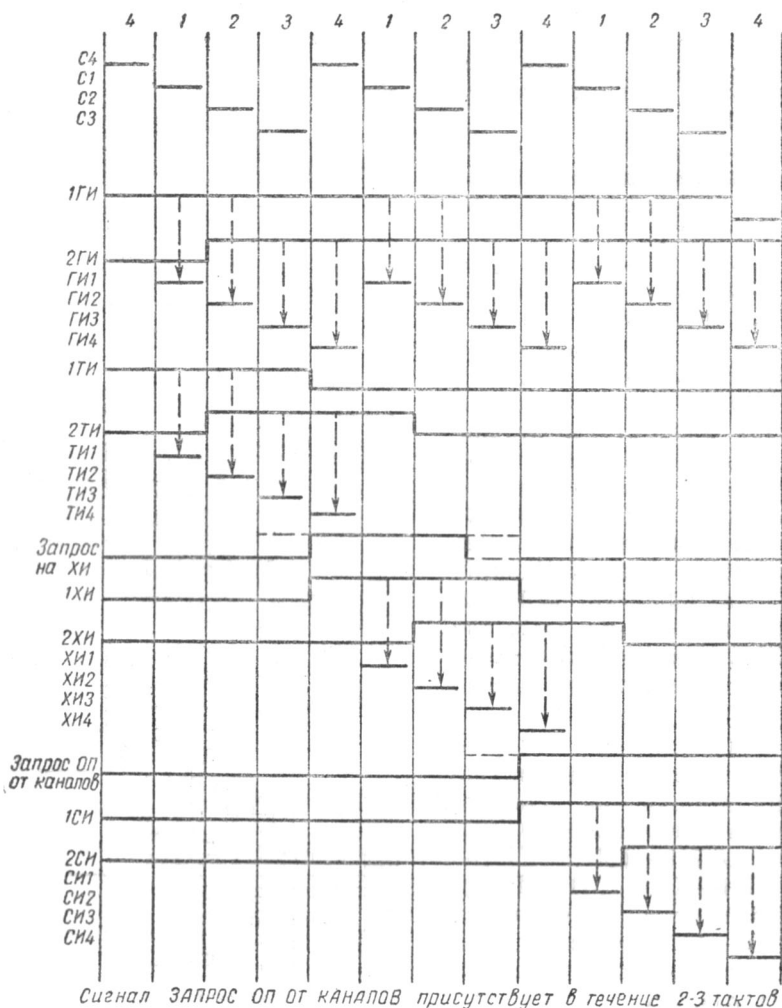


Рис. 2.3. Временная диаграмма выработки рабочих и главных тактовых импульсов

2.3. Узел распределения синхроимпульсов

Устройства, входящие в процессор (ЦП, каналы, ПП, ЗУКП), смонтированы в одной стойке, состоящей из трех рам: А, В, С. Аппаратура каналов занимает раму А, ПП и ЗУКП — раму В, а ЦП — раму С. Оперативная память занимает отдельную стойку. Системный генератор (генератор основных импульсов) расположен на раме С, там же находится и регистр тактов. Основные импульсы С1, С2, С3, С4 с помощью согласованного кабеля, а так-

товые потенциалы ТИ, СИ, ХИ, ГИ с помощью согласованной витой пары подаются на рамы А и В. Внутри рам эти импульсы и потенциалы тактов разводятся к платам со схемами, вырабатывающим необходимые рабочие тактовые импульсы. Платы со схемами называются ТЭЗ (типовой элемент замены). ТЭЗ синхронизации (ТС) расположены посередине каждого ряда логических ТЭЗ (ТЛ) в панелях. Если в ряду или во всей панели нет ТЭЗ, использующих синхроимпульсы, то ТС в этих панелях и рядах также отсутствуют.

В стойку ОП основные импульсы и потенциалы тактов не поступают, так как синхронизируются сами сигналы обращения к ОП (ЧТЕНИЕ, ЗАПИСЬ/РЕГЕНЕРАЦИЯ) и выдаваемый адрес.

Распределение синхроимпульсов на раме С показано на рис. 2.4. Подобным образом распределяются синхроимпульсы и на раме каналов А.

В раме В, которую занимает постоянная память и память ключей защиты, ТС не используются. Там, где необходимо (в постоянной памяти), рабочие тактовые импульсы вырабатываются непосредственно в ТЛ.

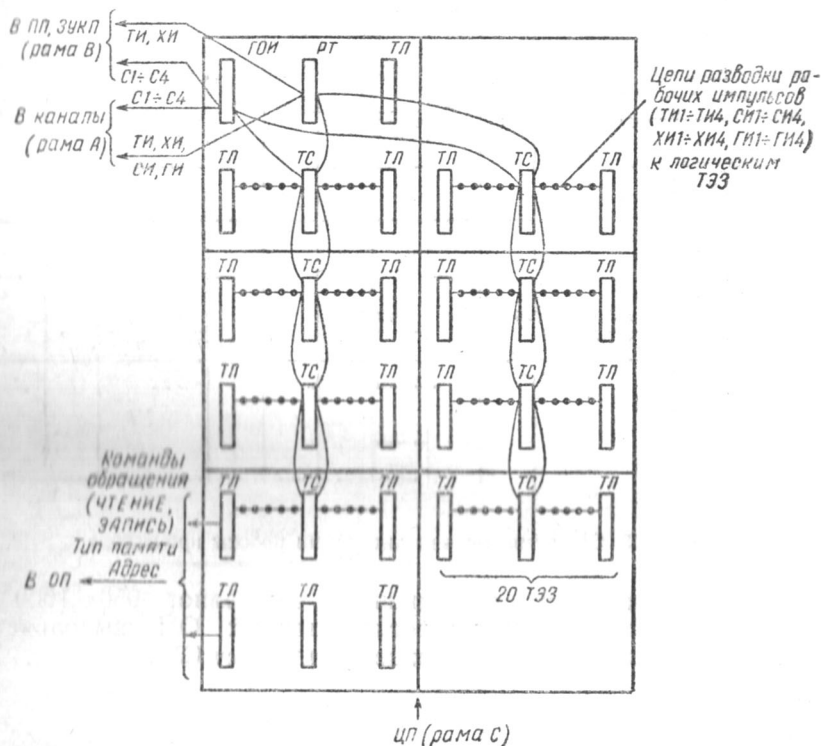


Рис. 2.4. Распределение тактовых импульсов на раме С: ГОИ — генератор основных импульсов, РТ — регистр тактовых импульсов, ТЛ и ТС — соответственно ТЭЗ с логическими схемами и ТЭЗ выработки рабочих синхроимпульсов

2.4. Общая временная диаграмма работы процессора

В течение одного такта ТИ выполняется одна микрокоманда. Причем выборка микрокоманды и ее выполнение сдвинуты на один такт (рис. 2.5).

Пуск ПП (осуществляется занесением адреса микрокоманды в регистр адреса ПП) происходит по ТИ2 или ХИ2. Информация из ПП, т. е. микрокоманда, появляется к концу этого такта.

Запуск арифметическо-логического блока производится путем занесения данных на входные регистры импульсом двойной длительности (ТИ1+ТИ2).

Результат операции принимается в один из регистров вычислительного устройства или каналов по ТИ4. Запуск оперативной памяти осуществляется путем подачи команды (сигналы ЧТЕНИЕ, ЗАПИСЬ/РЕГЕНЕРАЦИЯ) и адреса, которые синхронизируются импульсами ТИ1+ТИ2 (при обращении процессора или мультиплексного канала), или СИ1+СИ2 (при обращении селекторных

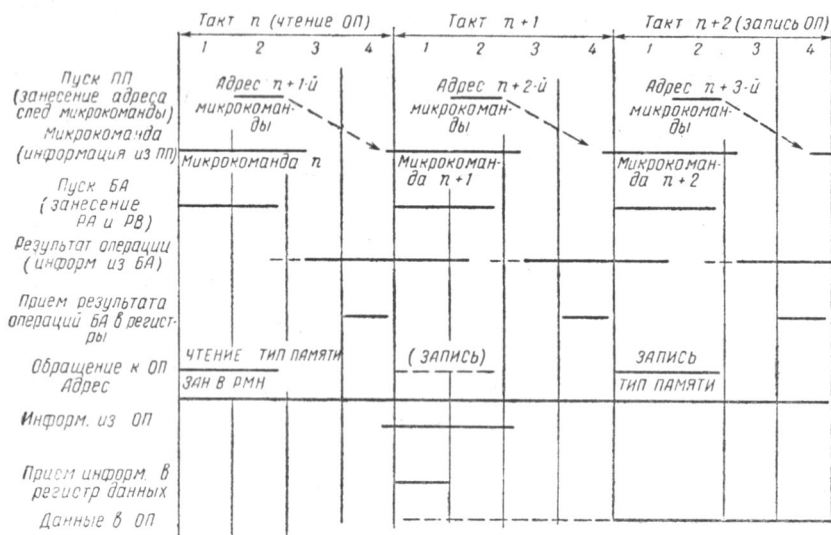


Рис. 2.5. Общая временная диаграмма работы процессора

каналов). Так как время ожидания ОП составляет 950—1000 нс, то прием считанных данных в регистр данных ОП выполняется уже в следующем такте (по основному импульсу С1).

БЛОК МИКРОПРОГРАММНОГО УПРАВЛЕНИЯ

3.1. Состав блока

В состав блока микропрограммного управления (рис. 3.1) входит постоянная память (ПП) со своими информационным (РМК) и адресным (РАПП) регистрами, узлы формирования адреса следующей микрокоманды (УФАМ, РВС, РВМ, УЗАНРАПП), дешифраторы адреса микрокоманды и полей микрокоманды.

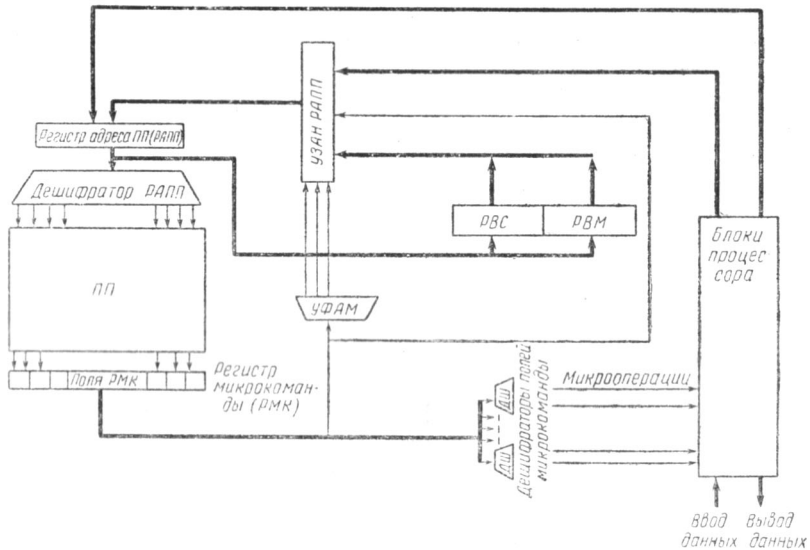


Рис. 3.1. Структура блока микропрограммного управления

Постоянная память (ПП) служит для хранения микропрограмм команд, процедур прерывания, мультиплексно-селекторной приостановки, микропрограммы ручных операций, диагностических

тестов и т. д. Объем памяти—8192 слова; длина слова—64 бит; время обращения (цикл)—1000 нс.

Постоянная память модели ЕС-1020 использует запоминающие элементы трансформаторного типа из материала с линейной характеристикой. Разъемные сердечники и печатные обмотки позволяют автоматизировать процесс записи информации в ПП.

Смена информации в ячейках ПП производится путем коммутационной замены носителей информации модулями по 512 слов или элементами модуля (картами) по четыре слова.

Микрокоманда, прочитанная из ПП по адресу в РАПП, запоминается в РМК. Микрокоманда разбита на поля определенной длины. Каждому полю соответствует дешифратор, который управляет работой определенных узлов процессора. Поля независимы и их дешифрация осуществляется независимо друг от друга.

Временная диаграмма работы блока микропрограммного управления представлена на рис. 3.2.

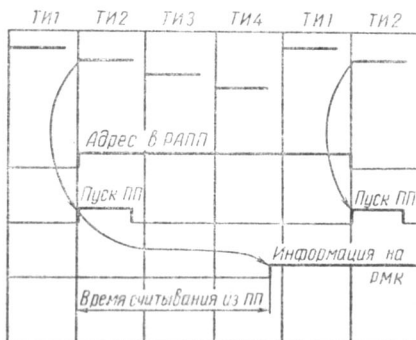


Рис. 3.2 Временная диаграмма работы блока микропрограммного управления

3.2. Структура микрокоманды

64-битовая микрокоманда (рис. 3.3) разбита на поля. Каждое поле определяет до 2^n микроопераций, где n — разрядность поля. За один машинный такт выполняется только одна микрооперация из данного поля.

Поля могут быть объединены в следующие группы:

- управление работой БА;
- управление работой ОП;
- занесение в триггеры и регистры;
- организация адреса микрокоманды;
- контроль.

Управление БА. Поля этой группы управляют работой арифметическо-логического блока:

Поле	Биты
А	5—8
ФУНКЦИЯ	9—12
В	13—17
ДЕФОРМАЦИЯ В	18—20

Четырехбитовое поле А определяет источник информации, подаваемой на входной регистр А (РА). Источником информации может быть один из регистров процессора или константа той же микрокоманды РМК (44—52).

Ниже приводятся коды поля А микрокоманды и соответствующие им регистры, являющиеся источником информации:

<i>Код поля</i>	<i>Источник информации</i>
0000	РА
0001	РН
0010	РЗ
0011	РЛ
0100	РД
0101	РТ
0110	РУ
0111	РР
1000	РИ
1001	Константа
1010	РМ
1011	РГ
1100	РП
1101	РБК
1110	РБЗ
1111	Нули

При значении кода 0000 информация на РА остается без изменений. При значении 1111 в этот регистр заносятся нули, а при значении 1001—содержимое бит 44—52 РМК (константа).

Четырехбитовое поле ФУНКЦИА определяет микрооперацию, выполняемую арифметическо-логическим блоком.

При всех значениях кодов поля ФУНКЦИА, кроме 0001, в БА выполняется одна из 15 микроопераций, заданная кодом этого поля. При значении кода 0001 выполняется заданная заранее микрооперация по косвенной функции.

Задание производится в одной из предшествующих микрокоманд, в которой код поля УСТАНОВ указывает на микрооперацию ЗАНЕСТИ КОСВЕННУЮ ФУНКЦИЮ (ЗКФ), а код заданной микрооперации БА записан в поле КМЛ. При выполнении этой микрокоманды содержимое поля КМЛ передается в регистр косвенной функции (РКФ).

Микрокоманда с кодом 0001 в поле ФУНКЦИА выполняет микрооперацию, код которой записан в РКФ БА.

<i>Код поля</i>	<i>Микрооперация БА</i>
0000	А∨В
0001	ВЫПОЛНИТЬ КОСВЕННУЮ ФУНКЦИЮ
0010	(А — В) ДЕС
0011	(А — В) ДВ
0100	А ТРАНЗИТ
0101	А&В
0110	(В — А) ДЕС
0111	(В — А) ДВ
1000	А∨В
1001	В ТРАНЗИТ
1010	А⊕В
1011	А&В
1100	СДВИГ В ВПРАВО
1101	СДВИГ В ВЛЕВО
1110	(А+В) ДЕС
1111	(А + В) ДВ

Пятибитовое поле В микрокоманды определяет источник информации, подаваемой на входной регистр В (РВ). Этим источником может быть либо один из регистров процессора, либо константа той же микрокоманды, либо байт состояния БА (ББА).

При значении кода 00000 информация на входном регистре РВ остается без изменения, при значении кода 11010 заносятся нули.

Код поля В	Источник информации	Код поля В	Источник информации
00000	РВ	01111	РБР
00001	РН	10000	РРЗ
00010	РЗ	10001	РР4
00011	РЛ	10010	РР5
00100	РД	10011	РР6
00101	РТ	10100	РР9
00110	РУ	10101	РРБ
00111	РР	10110	РРВ
01000	РИ	10111	РРГ
01001	Константа	11000	РРД
01010	РФ	11001	РРП
01011	РЕ	11010	Нули
01100	РО	11011	ББА
01101	РБД	11100	РРЕ
01110	РВС	11101	РРП

Трехбитовое поле ДЕФОРМАЦИЯ В определяет способы подачи информации из РВ в узлы обработки данных БА. Содержимое РВ остается без изменений.

Восьмибитовый байт информации в РВ разбивается на две тетрады (старшую и младшую).

При значении кода 111 производится предварительное задание режима ПЕРЕКОС КОСВЕННОЙ ФУНКЦИИ. В этом случае перекоп выполняется каждый раз, когда в поле ФУНКЦИЯ задана микрооперация ВЫПОЛНИТЬ КОСВЕННУЮ ФУНКЦИЮ.

Код поля	Деформация	Результат
000	ПРЯМО	Старшая и младшая тетрады передаются без изменений
001	НАКРЕСТ	Старшая тетрада передается на место младшей, младшая — на место старшей
010	МЛАДШИЕ ПРЯМО	На место старшей тетрады подаются нули, младшая передается без изменений
011	СТАРШИЕ ПРЯМО	На место младшей тетрады подаются нули, старшая передается без изменений
100	МЛАДШИЕ НАКРЕСТ	Младшая тетрада передается на место старшей, на место младшей передаются нули
101	СТАРШИЕ НАКРЕСТ	Старшая тетрада передается на место младшей, на место старшей передаются нули
110	ПЕРЕКОС	Младшая тетрада передается на место старшей, а старшая запоминается в буферном 4-битовом регистре. На место младшей передается прежнее содержание буферного регистра

Управление ОП. Поля этой группы управляют работой оперативной памяти:

<i>Поле</i>	<i>Биты</i>
АДРЕС	21—23
РЕЖИМ	24—25
ТИП	26—27

Трехбитовое поле АДРЕС определяет регистр, содержимое которого заносится в адресный регистр оперативной памяти.

При значении кода 000 адрес в адресный регистр не подается; при коде 111 в биты адресного регистра оперативной памяти подается константа.

<i>Код поля</i>	<i>Источник адреса</i>
001	РМФЕ
010	РГРИ
011	РПТУ
100	РР
101	РТ
110	РД
111	Константа

Содержимое РР, РТ, РД и константа передаются в младшие биты адресного регистра оперативной памяти МН, с помощью которых адресуются ячейки локальной памяти.

Двухбитовое поле РЕЖИМ определяет режим работы оперативной памяти:

<i>Код поля</i>	<i>Режим</i>
01	ЧТЕНИЕ (ЧТ)
10	ЗАПИСЬ, РЕГЕНЕРАЦИЯ (ЗП, РГ)
11	СТИРАНИЕ (СТ)

При коде 00 обращение к оперативной памяти не происходит.

Двухбитовое поле ТИП определяет тип памяти:

<i>Код поля</i>	<i>Тип</i>
00	З — память блока защиты
01	О — основная память
10	Л — локальная память
11	М — мультиплексная память

Оперативная память имеет разделенный цикл обращения, состоящий из двух тактов: ЧТЕНИЕ (СТИРАНИЕ) и ЗАПИСЬ (РЕГЕНЕРАЦИЯ).

В такте ЧТЕНИЕ происходит считывание информации из ОП по адресу в РМН, занесение ее в РНЗ и разрушение в ОП. Такт СТИРАНИЕ отличается от такта ЧТЕНИЕ тем, что информация не заносится в РНЗ. В такте ЗАПИСЬ (РЕГЕНЕРАЦИЯ) происходит запись содержимого РНЗ в оперативную память (запись новой информации или регенерация прочитанной ранее информации).

Между тактом ЧТЕНИЕ (СТИРАНИЕ) и тактом ЗАПИСЬ (РЕГЕНЕРАЦИЯ) содержимое РМН не изменяется. После каждой

микрооперации ЧТЕНИЕ (СТИРАНИЕ) следует микрооперация ЗАПИСЬ (РЕГЕНЕРАЦИЯ), т. е. не следуют друг за другом микрооперации ЧТ—ЧТ, ЧТ—СТ, СТ—ЧТ, ЗП—ЗП.

Занесение в триггеры и регистры. Эта группа содержит поле УСТАНОВ (биты 28—32) и поле С (биты 6—4).

Пятибитовое поле УСТАНОВ включает микрооперации, устанавливающие различные триггеры в единичное или нулевое состояние, а также специальные микрооперации.

В нулевое состояние триггеры устанавливаются при следующих кодах:

Код поля	Триггеры
00010	РБС (2)
00011	РБС (3)
00100	РБС (4)
00101	РБС (5)
01010	ППФ
01100	ПКФ
01110	РПФ
01111	РКФ
10000	ТБП
10010	ТВК
10100	ТАК
11011	ТПП (1ПЗУ)

В единичное состояние триггеры устанавливаются при кодах:

Код поля	Триггеры
00110	РБС (2)
00111	РБС (3)
01000	РБС (4)
01001	РБС (5)
01011	ППФ
01101	ПКФ
10001	ТБП
10011	ТВК
10101	ТАК
11100	ТПП (2ПЗУ)
11111	ТТО

При коде 00000 установка триггеров не происходит; если код 00001, то гасятся триггеры байта состояния арифметическо-логического блока (ББА).

Коды 11001 и 11010 определяют алгоритмы установки кода условия—КУ1 и КУ2 соответственно. Установка кода условия происходит в триггерах РБС (6—7) по следующему алгоритму:

	РБС (6)	РБС (7)	
КУ1	0	0	при $\Sigma = 0$
	0	1	при $\Sigma < 0$
	1	0	при $\Sigma > 0$
КУ2	1	1	при переполнении
	0	0	при $\Sigma = 0$ и отсутствии переноса
	0	1	при $\Sigma \neq 0$ и отсутствии переноса
	1	0	при $\Sigma = 0$ и наличии переноса
	1	1	при $\Sigma \neq 0$ и наличии переноса

Код 10110—ИГН (микрооперация ИГНОРИРОВАТЬ) используется для игнорирования значения межбайтового переноса.

По коду 10111—ЗКФ (микрооперация ЗАНЕСТИ КОСВЕННУЮ ФУНКЦИЮ) производится занесение кода микрооперации БА из поля КМЛ в РКФ БА и сброс предварительно заданного режима перекоса (если в поле ДЕФОРМАЦИЯ В не задана микрооперация ПЕРЕКОС КОСВЕННОЙ ФУНКЦИИ).

Код 11000—СПБ означает микрооперацию СБРОС БУФЕРА ПЕРЕКОСА; код 11101—ПСО—микрооперацию ПОТЕНЦИАЛ СОСТОЯНИЯ ОСТАНОВА, используемую при организации остановленного состояния.

Пятибитовое поле С определяет приемник информации с выхода С БА, которым является один из регистров процессора.

При значении кода 00000 информация с выхода С БА никуда не принимается.

Код поля	Приемник информации	Код поля	Приемник информации
00001	РН	01111	РБД
00010	РЗ	10000	РБС
00011	РЛ	10001	РБР
00100	РД	10010	РБК
00101	РТ	10011	РБЗ
00110	РУ	10100	РР1
00111	РР	10101	РР2
01000	РИ	10110	РР5
01001	РМ	10111	РР6
01010	РФ	11000	РР7
01011	РЕ	11001	РР8
01100	РО	11010	РР9
01101	РГ	11011	РРА
01110	РП		

Организация адреса микрокоманды. Поля этой группы используются для формирования адреса следующей микрокоманды.

Поле	Биты
УСЛ1	33—36
УСЛ0	37—41
М	42—43
КСТ/АПСТ	45—48
КМЛ	49—52
АПМЛ	53—58

Четырехбитовое поле УСЛ1 определяет установку РАПП (1). При коде 0000 в РАПП (1) заносится 0; при коде 0001 в РАПП (1)—1. При остальных кодах занесение в РАПП (1) выполняется по следующему правилу: заносится 1, если соответствующий коду поля триггер находится в определенном состоянии:

Код поля	Соответствующий триггер	Состояние триггера
0010	РБС (1)	«1»
0011	РБС (3)	«1»
0100	РБС (5)	«1»
0101	РБС (7)	«1»
0110	ТРПФ	«0»

Код поля Соответствующий триггер Состояние триггера

0111	ТРПФ	«1»
1000	ТРКФ	«0»
!001	ТРКФ	«1»
1010	ТППФ	«0»
1011	ТППФ	«1»
1101	ТПКФ	«1»
1110	ТЧЕТ	«1»

Если это правило не выполняется, то в РАПП (1) заносится 0. Пятибитовое поле УСЛО определяет установку РАПП (0). При коде 00000 в РАПП (0) заносится 0, при коде 00001—1. При остальных кодах занесение в РАПП (0) подчиняется вышеописанному правилу при следующем соответствии:

<i>Код поля</i>	<i>Соответствующий триггер</i>	<i>Состояние триггера</i>
00010	РБС (0)	«1»
00011	РБС (2)	«1»
00100	РБС (4)	«1»
00101	РБС (6)	«1»
00110	ТРПФ	«0»
00111	ТРПФ	«1»
01000	ТРКФ	«0»
01001	ТРКФ	«1»
01010	ТППФ	«0»
01011	ТППФ	«1»
01100	ТПКФ	«0»
01101	ТПКФ	«1»
01110	ТЗН	«1»
01111	ТПЕР	«1»
10000	ТНДД	«1»
10001	ТВВВ	«1»
10010	ТАК	«1»
10011	ТВК	«1»
10100	ТЦП	«1»
10101	ТБЗ	«1»
10111	ТРП	«1»

В РАПП (0) заносится 0, если это правило не выполняется. Двухбитовое поле М определяет способ формирования адреса следующей микрокоманды:

<i>Код поля</i>	<i>Способ</i>
00	Адрес перехода длинный (АПД)
01	Адрес перехода короткий (АПК)
10	АПД или сброс РАПП (АПД/В)
11	Функциональный переход, передача в РАПП из РВС, РВМ или РГРИ

Поля КСТ/АПСТ, КМЛ, АПМЛ. Содержимое этих полей непосредственно передается в РАПП и регистры А, В, МН, КФ (образуя адрес следующей микрокоманды и константу).

В одной микрокоманде адрес и константа образуются одним из следующих способов:

а) АПД—длинный двенадцатибитовый адрес перехода (поля КСТ/АПСТ, АПМЛ, УСЛ1, УСЛО) и короткая четырехбитовая константа (поле КМЛ);

б) АПК—короткий восьмибитовый адрес (поля АПМЛ, УСЛ1, УСЛО) и длинная восьмибитовая константа (поля КСТ/АПСТ и КМЛ).

Выбор способа определяется микрооперациями АПД и АПК из поля М.

Контроль. Поля этой группы являются контрольными битами:

Поле	Биты
КР1	44
КР2	62
КР3	63

Контрольные биты КР1, КР2 и КР3 являются контрольными битами соответственно константы РМК (45—52), РАПП и РМК.

Запас. Запасные биты 59—61 не используются.

3.3. Дешифраторы полей микрокоманды

Дешифраторы полей микрокоманды служат для преобразования микрокоманды в набор микроопераций.

Дешифратор поля С—двухступенчатый. Выработанные им сигналы используются в конце такта по синхроримпульсу ТИ4. Сохранение сигнала до конца такта обеспечивается триггерами на входе дешифратора.

Дешифратор поля А—двухступенчатый.

Дешифратор поля В—одноступенчатый. Коды 00001—01111, 11010—11011 вырабатывают сигналы, управляющие занесением содержимого регистров вычислительного устройства в РВ. Коды 10000—11001, 11100—11101 вырабатывают сигналы, управляющие занесением содержимого регистров каналов в РВ.

Информация из регистров каналов попадает в РВ через РВК (рис. 3.4). Занесением содержимого регистра каналов в РВК управляют сигналы с дешифратора. Для передачи содержимого РВК и РВ вырабатывается сигнал РВ:=РВК.

Дешифратор поля ПАМЯТЬ—одноступенчатый. Так как сигналы ЧТЕНИЕ и СТИРАНИЕ для ОП идентичны, то вырабатывается один сигнал, выдаваемый в ОП:

$$\text{ЧТСТ} = \text{ЧТ} \vee \text{СТ}$$

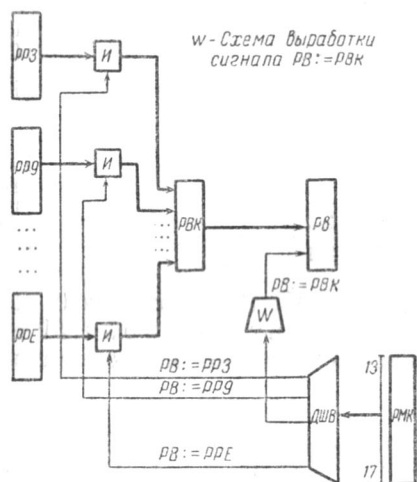


Рис. 3.4. Схема занесения информации в регистр В из регистров каналов

Дешифратор поля УСТАНОВ—двухступенчатый, аналогичный дешифратору поля С.

Дешифраторы полей УСЛ1 и УСЛ0—одноступенчатые.

3.4. Триггеры состояний

Поля микрокоманды УСТАНОВ, УСЛ1, УСЛ0 задают главным образом микрооперации установки и анализа триггеров, отражающих внутреннее состояние процессора:

РБС (0) предназначен для запоминания сбоя по адресации (сбой вырабатывается при обращении в основную память по адресу, превышающему имеющийся объем памяти);

РБС (1) служит для запоминания сбоя по защите (сбой вырабатывается при несоответствии ключа защиты программ ключу памяти);

РБС (2—5)—универсальные индикаторы, используемые для организации ветвления в микропрограммах;

РБС (6—7) служат для запоминания кода условия (КУ), устанавливаемого в конце выполнения команды. В процессе выполнения команды РБС (6—7) могут использоваться как универсальные указатели;

ТРПФ (триггер результата прямой функции) хранит информацию о результате микрооперации, выполняемой по прямой функции. Устанавливается в единичное состояние при каждой микрооперации, выполняемой с помощью прямой функции, если результирующий байт не равен нулю. Состояние ТРПФ не изменяется, если байт результата равен нулю;

ТРКФ (триггер результата косвенной функции) хранит информацию о результате микрооперации, выполняемой по косвенной функции. Устанавливается в единичное состояние по результату каждой микрооперации, выполняемой с помощью косвенной функции, если результирующий байт не равен нулю. Состояние ТРКФ не изменяется, если результирующий байт равен нулю. При анализе учитывается результат выполнения последней микрооперации, выполненной с помощью косвенной функции;

ТЗН (триггер знака) хранит информацию о состоянии старшего (нулевого) бита байта результата микрооперации, изменяет свое состояние в конце каждого такта работы БА, запоминая значение нулевого бита байта результата;

ТЧЕТ (триггер четности байта) хранит информацию о состоянии младшего (седьмого) бита байта результата, изменяет свое состояние в конце каждого такта работы БА, запоминая значение седьмого бита байта результата;

ТПЕР (триггер переполнения) хранит информацию о наличии или отсутствии переполнения восьмибитовой сетки БА, изменяет свое состояние в конце такта работы БА при выполнении микроопераций сложения, вычитания, сдвига влево;

ТНДД (триггер неверных десятичных данных) хранит инфор-

мацию о правильности задания десятичных данных. Устанавливается в нулевое состояние по микрооперации ГАШ, а в единичное состояние—по неверной десятичной цифре на любом входе БА при выполнении микроопераций (А+В)ДЕС или (А-В)ДЕС;

ТВВВ (триггер запроса ввода-вывода и внешних прерываний) устанавливается при появлении запроса на внешние прерывания или прерывания ввода-вывода, а также на корректировку таймера и состояние ОСТАНОВ;

ТБП (триггер блокировки прерывания) используется для блокировки прерывания по защите, адресации и машинному контролю во время выполнения микропрограммы, гашения системы (идентификатор ПЗП);

ТАК (триггер адреса команды) находится в единичном состоянии, если адрес команды располагается в локальной памяти, и в нулевом состоянии, если адрес команды—в РМФЕ;

ТВК (триггер выборки команды)—единичное состояние триггера отражает прохождение микропрограммы выборки команды (идентификатор ВЫБОР);

ТЦП (триггер центрального процессора)—единичное состояние триггера отражает работу каналов, нулевое—работу вычислительного устройства;

ТБЗ (признак блока защиты) находится в единичном состоянии, если в процессоре не установлен блок защиты памяти;

ТРП (триггер разрешения пакета)—единичное состояние этого триггера свидетельствует о том, что внешнее устройство запрашивает монопольный режим работы;

ТПП (триггер постоянной памяти)—фактически схема занесения в РАПП (12) устанавливается в единичное состояние при обращении к микрокоманде, адрес которой более 4096;

ТТО (триггер тяжелого останова)—единичное состояние его вызывает останов синхроимпульсов процессора—ТИ, ХИ, СИ.

3.5. Способы организации адреса микрокоманды

Как уже отмечалось, часть полей микрокоманды служит для задания следующего адреса и может:

указать на необходимость сбросить РАПП в нулевое состояние, т. е. перевести его в начальное состояние, соответствующее началу микропрограммы выборки команды;

передать содержимое РВС, РВМ в РАПП или занести другой фиксированный адрес;

указать на необходимость установки РАПП с учетом кода условия, отдельных триггеров или кода операции;

выбрать один из двух, четырех, тридцати двух адресов и т. д.

По способу организации адрес микрокоманды может быть нефиксированным и фиксированным.

Нефиксированный адрес организуется с помощью информации, содержащейся в полях М, УСЛО, УСЛ1, КСТ/АПСТ, АПМЛ предыдущей микрокоманды.

Поле М определяет четыре способа организации нефиксированного адреса микрокоманды;

АПД — адрес перехода длинный;

АПК — адрес перехода короткий;

АПД/В — АПД или сброс РАПП, т. е. переход к микропрограмме выборки команды;

Ф, РС, РМ, ГРИ — функциональный переход или переход по регистру возврата селекторного канала (РВС), или же переход по регистру возврата мультиплексного канала (РВМ), или занесение в РАПП из ГРИ.

При АПД и АПК с помощью полей УСЛО и УСЛ1, определяющих нулевое или единичное состояние двух младших бит РАПП, осуществляется ветвление в микропрограммах по четырем направлениям.

При Ф (функциональном переходе) с помощью УСЛО и содержимого старшей тетрады входа В БА, пересылаемых с выхода РВ БА в 1—4-й биты РАПП, осуществляется ветвление по 32 направлениям.

В тринадцатитривитовых регистрах ВС и ВМ запоминается адрес микрокоманды, перед которой происходят прерывание микропрограммы и переход на выполнение микропрограммы обслуживания селекторного или мультиплексного канала (микропрограммная приостановка).

Так как микропрограмма обслуживания селекторного канала имеет более высокий приоритет, то она может прерывать микропрограмму обслуживания мультиплексного канала. В этом случае в РВМ запоминается адрес микрокоманды рабочей микропрограммы, к которой необходимо перейти после выполнения микропрограммы обслуживания мультиплексного канала, а в РВС запоминается адрес микрокоманды микропрограммы обслуживания мультиплексного канала, к которой необходимо перейти после выполнения микропрограммы селекторного канала.

Блок-схема организации нефиксированного адреса микрокоманды представлена на рис. 3.5.

Ниже приведены перечень кодов поля М и соответствующие способы организации нефиксированного адреса микрокоманды.

Код 00. В тринадцатитривитовый регистр адреса постоянной памяти (РАПП) заносится адрес перехода длинный (АПД), который формируется из АПСТ (45—48-й биты РМК) и АПМЛ (53—58-й биты РМК). Содержимое 45—48-го бит передается в 11—8-й биты РАПП соответственно, а содержимое бит 53—58 — в 7—2-й биты РАПП соответственно.

Нулевой и первый биты РАПП формируются дешифраторами полей УСЛО и УСЛ1 соответственно, с помощью которых можно задать безусловное и условное состояние этих бит.

Безусловное состояние задается комбинациями 00000 и 00001 в поле УСЛО и комбинациями 0000 и 0001 в поле УСЛ1. Нуль в нулевой бит РАПП заносится при комбинации 00000 в поле УСЛО, а единица — при 00001. В первый бит РАПП заносится нуль или единица при тех же комбинациях, но по полю УСЛ1.

Условное состояние 0-го и 1-го бит РАПП задается всеми оставшимися комбинациями (см. рис. 3.3) в полях УСЛО и УСЛ1. Например, если в поле УСЛО комбинация 00111 (ТРПФ) и триггер ТРПФ находятся в единичном состоянии, следовательно, условие выполняется и в 0-й бит РАПП заносится единица. Нуль

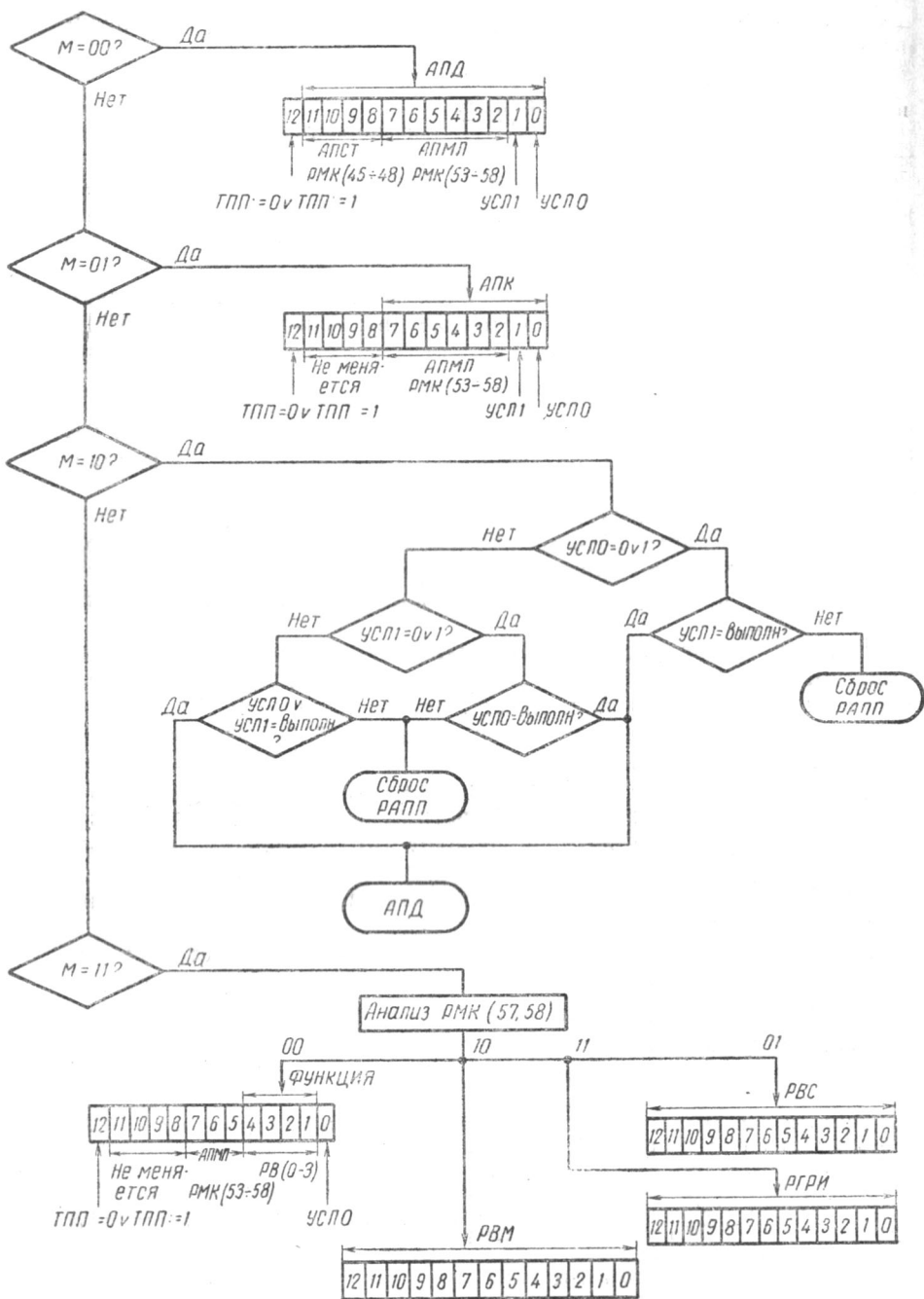


Рис. 3.5. Блок-схема организации нефиксированного адреса микрокоманды

в 0-й бит РАПП заносится, если триггер РПФ в нулевом состоянии, т. е. условие не выполняется. В 1-й бит РАПП заносится единица или ноль так же, как и в 0-й бит, в зависимости от выполнения условия в поле УСЛ1.

Двенадцатый бит РАПП устанавливается в нулевое или единичное состояние микрооперациями 1ПЗУ (ТПП:=0) или 2ПЗУ (ТПП:=1) поля УСТАНОВ. Микрооперацией ТПП:=0 12-й бит РАПП устанавливается в нулевое состояние, а микрооперацией ТПП:=1 — в единичное. Если 12-й бит РАПП находится в нулевом состоянии, то обращение происходит к 0—4095-й ячейкам постоянной памяти (ПП), если в единичном, то — к 4096—8192-й ячейкам ПП.

Код 01. В РАПП заносится адрес перехода короткий (АПК). При АПК содержимое 8—11-го бит РАПП не меняется, а во 2—7-й бит заносится АПМЛ — содержимое 53—58-го бит РМК. Биты 0, 1, 12 РАПП формируются так же, как и при АПД.

Код 10. Возможны два варианта организации нефиксированного адреса микрокоманды:

а) АПД — адрес формируется так же, как и в первом случае, т. е. при $M=00$;

б) В — сброс РАПП — занесение нулей во все биты регистра. Сброс РАПП позволяет перейти к микропрограмме выборки, расположенной в постоянной памяти с нулевого адреса.

Эти варианты зависят от выполнения условий, заданных в дешифраторах полей УСЛ0 и УСЛ1, следующим образом:

УСЛ0=0V1, а в поле УСЛ1 задано условие и оно выполняется, тогда в РАПП заносится АПД. Если условие не выполняется, то происходит сброс РАПП;

УСЛ1=0V1, в поле УСЛ0 задано условие и оно выполняется, в РАПП заносится АПД, в противном случае — сброс РАПП;

если заданы условия в поле УСЛ0 или УСЛ1 и одно из них выполняется — АПД; если же заданы условия в поле УСЛ0 и УСЛ1 и они не выполняются — сброс РАПП.

Код 11. Возможны четыре варианта организации адреса микрокоманды, отличающихся состоянием 57 и 58-го бит РМК:

а) Ф — функциональный переход. Этот вариант возможен при равенстве 57-го и 58-го бит РМК нулю. В этом случае содержимое 8—11-го бит РАПП не меняется, в 5, 6, 7-й биты передается содержимое 53—55-го бит РМК (старшие биты АПМЛ) соответственно, а в 4—1-й биты заносится ВСТ, где ВСТ — четыре старших бита (0—3) РВ БА. Содержимое 0-го бита РВ заносится в 4-й бит РАПП, содержимое 1-го бита РВ — в 3-й бит РАПП и т. д. 0-й бит РАПП формируется дешифратором поля УСЛ0, аналогично как при АПД и АПК;

б) РВС — если 57-й бит равен 0, а 58-й бит равен 1, то в РАПП заносится информация из РВС;

в) РВМ — если 57-й бит равен 1, а 58-й бит равен 0, то в РАПП заносится информация из РВМ;

г) РАПП1=ГРИ — если 57-й и 58-й биты равны 1, то в РАПП заносится информация из регистров Р (биты 3—7 в 12—8-й биты РАПП соответственно), И (биты 0—7 в 7—0-й биты РАПП соответственно).

Занесение нефиксированного адреса в РАПП происходит по тринадцати шинам. Наличие или отсутствие высокого уровня на шине соответствует занесению в определенный бит РАПП единицы или нуля.

Занесение нефиксированного адреса происходит по синхронному импульсу ТИ2.

Фиксированные адреса — это адреса, формирование которых происходит без помощи полей КСТ/АПСТ, АПМЛ, УСЛ0, УСЛ1. Занесение фиксированного адреса в РАПП осуществляется по синхронному импульсу ХИ2.

Блок-схема занесения в РАПП фиксированного адреса представлена на рис. 3.6.

Фиксированных адресов, заносимых в РАПП, шесть:

0001 — начальный адрес микропрограммы гашения системы. В РАПП адрес заносится при нажатии на пульт управления кнопок

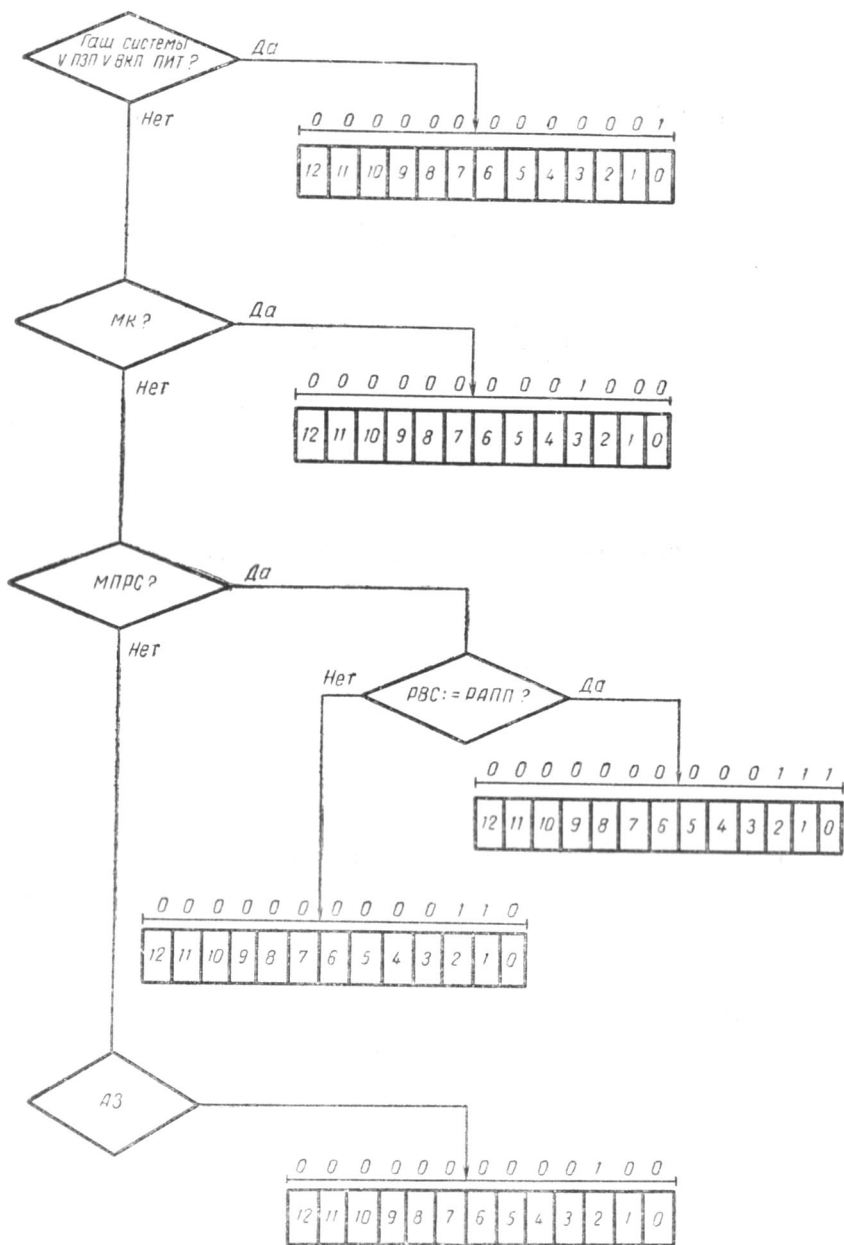


Рис. 3.6. Блок-схема организации фиксированного адреса микрокоманды

«Питание вкл.», «Гашение» или «Загрузка» и сигналом аппаратного гашения (АГ);

0008 — начальный адрес микропрограммы обработки прерываний по контролю машины (МАШК);

0007 — начальный адрес микропрограммы обслуживания селекторного канала (ОБССК);

0006 — начальный адрес микропрограммы обслуживания мультиплексного канала (ОБСМК);

0004 — начальный адрес микропрограммы обработки сбоев по защите и адресации (АЗ);

000С — этот адрес используется в микропрограмме сканирования ПП (вырабатывается, когда есть микрооперация РАПП: = ГРИ и переключатель «Тип памяти» на пульте управления в положении ПП).

Между запросами на занесение фиксированного адреса в РАПП установлен следующий приоритет:

1) адрес микропрограммы гашения системы;

2) адрес микропрограммы МАШК или сканирования (при появлении запроса на адрес 000С микропрограммы сканирования отсутствуют запросы на занесение адресов микропрограмм МАШК, ОБССК или ОБСМК, АЗ);

3) адрес микропрограммы ОБССК или ОБСМК (выбор определяется сигналом РВС: = РАПП);

4) адрес микропрограммы АЗ.

3.6. Разработка микропрограмм

Использование принципа микропрограммного управления процессом требует создания специальной системы микропрограммирования, обеспечивающей высокую механизацию процесса написания микропрограмм, их проверку и выдачу документации для создания физических носителей микрокоманд в виде карт ПП.

Принятый в машине ЕС-1020 вариант микропрограммного управления, основанный на совмещении нескольких микроопераций в одной микрокоманде, большой объем микропрограмм, достаточно сложное взаимодействие микропрограмм, делают невозможным программирование в двоичных кодах ПП. Поэтому был разработан специальный язык символического микропрограммирования (ЯМП), который относится к машинно-ориентированным языкам автопрограммирования уровня «один к одному».

Программа, написанная на ЯМП, называется символической микропрограммой (СИМП). На рис. 3.7 показано два вида записи одной и той же микрокоманды: на ЯМП и в двоичных кодах ПП (на машинном языке).

Запись И=И+К читается процессором следующим образом: на вход РА БА подается содержимое регистра И (код микрооперации 1000), на вход РВ БА — константа (код микрооперации 01001). Значение константы записано в поле КОНСТ и равно 0010, т. е. 2.

НОМЕР	АДР	МЕТКА	ФУНКЦИЯ	ПАМЯТЬ	УСТАНОВ	УСЛ1	УСЛО	КОНСТ	ПЕРЕХОД	ЗАМЕЧАНИЯ
080	63В		И-И+К	ГРИ ЧТО	ОППФ		1602	?	44	644

а)

ИДЕНТИФИКАТОР МДМ

АДР	С	А	ФУНКЦИЯ	В	ДЕФ	АДРР	Р	Т	УСТАНОВ	УСЛ1	УСЛО	М	1К/АП	К	АП	?	?	
63В	01000	1000	1111	01001	000	010	01	01	01010	0000	00011	00	0	0110	0010	010001	1	1

б)

Рис. 3.7. Запись микрокоманды на языке символического микропрограммирования (а) и на машинном языке (б)

В арифметическо-логическом блоке производится двоичное сложение (код микрооперации 1111) содержимых RA и RB. Результат сложения подается в регистр И (код микрооперации 01000). Для микрооперации сложения необходимо иметь значение входящего переноса. Задание в поле УСТАНОВ кода 01010 (ОППФ) вызывает формирование нулевого значения входящего переноса, независимо от состояния переноса, образованного в результате предыдущего действия.

Запись ГРИ ЧТО рассматривается следующим образом: содержимое регистра ГРИ подается в адресный регистр памяти МН по микрооперации, выработанной в дешифраторе поля АДРЕС (код микрооперации 010). В дешифраторе поля РЕЖИМ вырабатывается сигнал чтения (код микрооперации 01), а в дешифраторе поля ТИП — сигнал, определяющий обращение к основной памяти (код микрооперации 01).

Параллельно с обработкой информации происходит формирование адреса следующей микрокоманды, в котором участвуют дешифраторы полей УСЛ1, УСЛО, М и 45—48-й, 53—58-й биты РМК.

Поле М указывает способ формирования адреса следующей микрокоманды. При записи микрокоманды на символическом языке он явно не задан. Адрес перехода (смотри поле ПЕРЕХОД) в микрокоманде на ЯМП задан микропрограммистом (44), но с учетом поля УСЛО он может быть или 44, или 45. В данной микрокоманде используется короткая константа, равная 2, поэтому можно задать адрес перехода длинный, равный 644.

В 11—8-й биты РАПП заносится содержимое поля КСТ/АПСТ (45—48-й биты соответственно).

В 7—2-й биты РАПП заносится содержимое поля АПМЛ (53—58-й биты РМК соответственно).

Информация из бит 45—48, 53—58 РМК передается в РАПП (2—11):

РАПП	12	11	10	9	8	7	6	5	4	3	2	1	0
РМК	0	1	1	0	0	1	0	0	0	1			

РАПП (12) определяется состоянием триггера ТПП (триггер постоянной памяти). Нулевое состояние этого триггера определяет младшую часть ПП (0—4095), единичное — старшую (4096—8192). Этот триггер управляется полем УСТАНОВ, т. е. устанавливается на нуль или единицу. Содержимое его переносится в РАПП при каждом формировании адреса (см. рис. 4.5).

В рассматриваемом примере ТПП находится в нулевом состоянии. Младшие две цифры РАПП (1,0) определяются полями УСЛ1 и УСЛ0. Так как в дешифраторе поля УСЛ1 код равен 0000, то в РАПП (1) заносится 0. Поле УСЛ0 задает анализ триггера РБС (2) (код 00011), и если этот триггер находится в единичном состоянии, то в РАПП (0) заносится 1, т. е. адрес следующей микрокоманды 0645.

Глава 4

БЛОК РЕГИСТРОВ

4.1. Назначение и состав блока

Блок регистров (БР) предназначен для хранения и выдачи различной информации, участвующей в процессе обработки программы. Он содержит адресные регистры (МФЕ, ГРИ, ПТУ), регистры общего пользования (Л, Д), служебные регистры (БК, БД, БС, БР, О, БЗ), узел формирования адреса оперативной памяти (УФАОП), узлы формирования прерываний по защите и неверной адресации, машинному контролю (УФПАЗ, УФКТРМ), узлы выработки контрольных бит служебных регистров (УВКРБК, УВКРБД, УВКРБС, УВКРБР, УВКРО).

Во все регистры информация заносится с девятибитового (восемь информационных и один контрольный бит) выхода С БА.

Все адресные регистры содержат 19 информационных (0—18) и два контрольных бита, т. е. два байта с соответствующим контрольным битом каждый и три бита расширения.

Биты расширения и каждый байт рассматриваются как самостоятельные части адресного регистра и поэтому считаются отдельными регистрами. Например, регистр МФЕ состоит из регистров М (биты расширения), Ф (старший байт), Е (младший байт).

В РМ, РГ, РП (7, 6 и 5-й, уплотненный, биты) информация заносится с выхода С БА. Пятый бит (уплотненный) формируется в БА и указывает на наличие единицы в 0—5-м битах выхода С.

Два бита расширения используются при формировании адреса оперативной памяти. Для подачи на РА БА используются три бита расширения.

Разрядность регистров общего пользования равна одному байту плюс один контрольный бит. Контроль адресных, служебных и регистров общего назначения осуществляется при подаче их во входной регистр БА.

Разрядность служебных регистров равна одному байту. Краткую информацию о регистрах содержит табл. 4.1.

Наименование регистра	Разрядность	Пересдача
Адресные		
РМ	8+1К	РА БА, РМН
РФ	8+1К	РВ БА, РМН
РЕ	8+1К	РВ БА, РМН
РГ	3+1К	РВ БА, РМН
РР	8+1К	РВ БА, РМН
РИ	8+1К	РВ БА, РМН
РП	3+1К	РА БА, РМН
РТ	8+1К	РА БА, РМН
РУ	8+1К	РА и РВ БА, РМН
Общие		
РЛ	8+1К	РА и РВ БА
РД (адресный)	8+1К	РА и РВ БА, РМН
Служебные		
РВК	8	РА БА
РВД	8	РВ БА
РВС	8	РВ БА
РБР	8	РВ БА
РВЗ	8+1К	РА БА
РО	8	РВ БА

4.2. Адресные регистры

Адресные регистры МФЕ, ГРИ, ПТУ предназначены для хранения адресов команд и операндов, а также любой другой информации.

Изменение информации в адресных регистрах осуществляется микропрограммно, т.е. передачей выхода С БА на вход каждой восьмибитовой части адресного регистра по микрооперациям поля С микрокоманды.

Содержимое каждой части адресного регистра принимается на РА или РВ БА по микрооперациям полей А и В микрокоманды.

Регистр МФЕ (РМФЕ) служит для хранения адреса текущей команды. Если этот регистр используется для хранения операндов, то адрес текущей команды записывается в локальную память: РМ — по адресу 8D, РФ — по адресу 8E, РЕ — по адресу 8F и устанавливает триггер адреса команды (ТАК) в единичное состояние. При возвращении адреса текущей команды в РМФЕ выдается сигнал $ТАК = 0$.

При передаче содержимого РМ на РА БА используются три бита расширения с выработкой контрольного бита схемой свертки ЛСВ2 (рис. 4.1).

Занесение содержимого выхода С БА в РМФЕ осуществляется по сигналам $РМ := С$, $РФ := С$, $РЕ := С$, вырабатываемым дешифратором поля С микрокоманды.

Содержимое РМФЕ, за исключением РМ (5) (18 информационных и три контрольных бита), заносится в РМН по сигналу

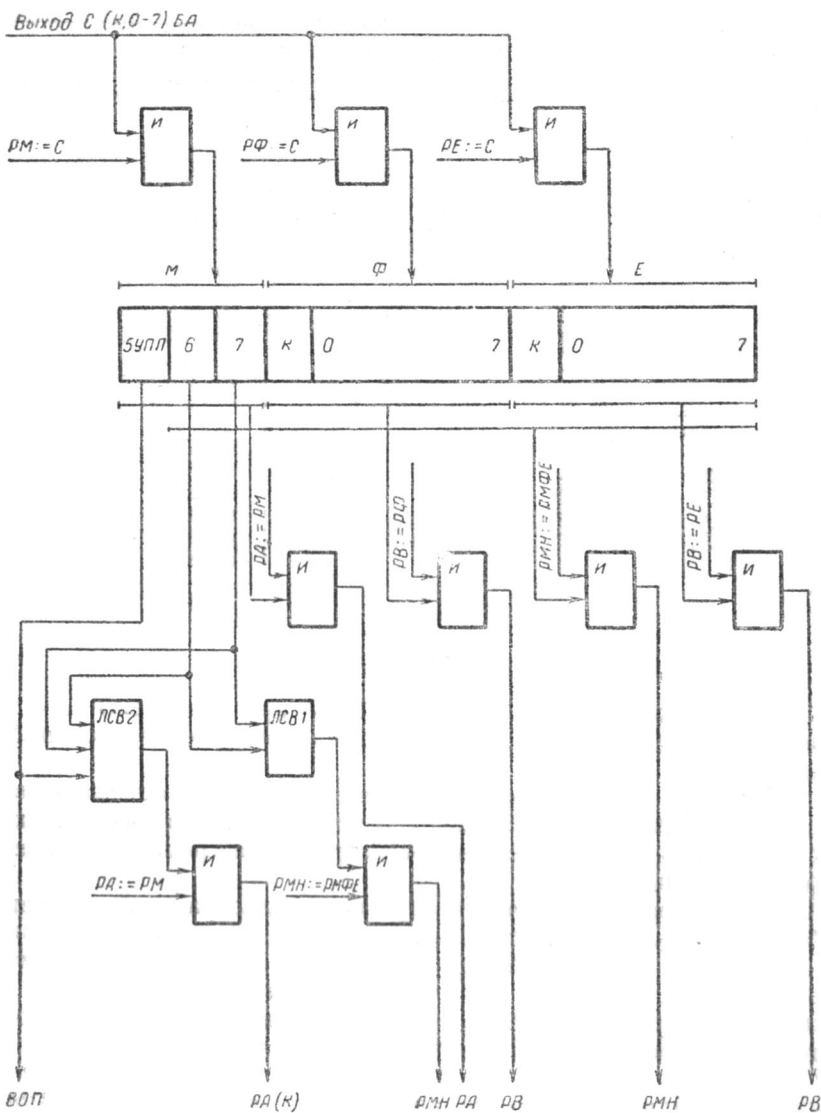


Рис. 4.1. Схема регистра МФЕ

РМН := РМФЕ, вырабатываемому дешифратором поля микрокоманды АДРЕС.

Наличие 1 в 18-м бите РМ(5) указывает на адрес, превышающий максимальный объем памяти (256К байт) и ведет к сбою по адресации.

По сигналу $PA:=PM$, поступающему с дешифратора поля А микрокоманды, идет передача информации из PM (три информационных (5—7) и один контрольный бит, выработанный схемой свертки ЛСВ2) на PA БА. Содержимое информационных бит передается в младшие биты PA ; остальные биты заполняются нулями.

На PA БА принимается информация из регистров PF , PE по сигналам $PB:=PF$, $PB:=PE$, вырабатываемым дешифратором поля В микрокоманды.

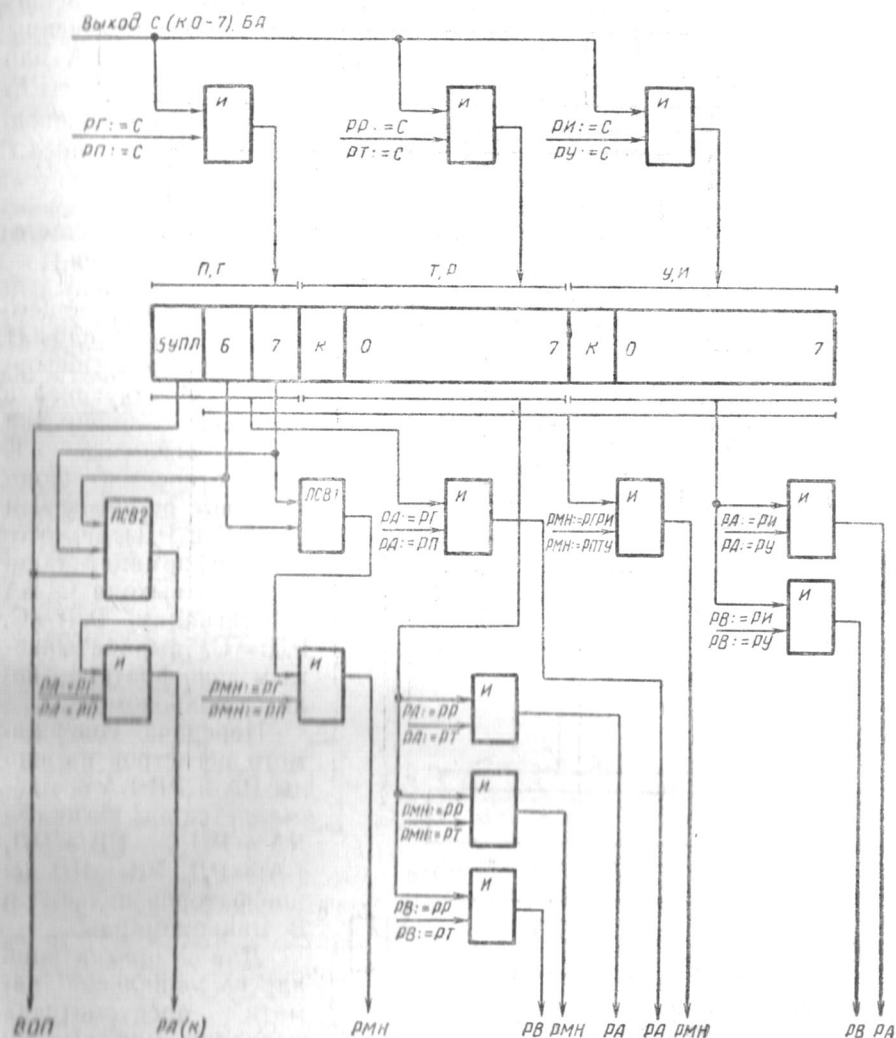


Рис. 4.2. Схема регистров ГРИ и ПТУ

Регистры ГРИ и ПТУ (РГРИ и РПТУ) служат для хранения адресов операндов. Информация заносится с выхода С БА (восемь информационных и один контрольный бит) в РР, РИ по сигналам $РР:=С$, $РИ:=С$, вырабатываемым дешифратором поля С микрокоманды. Передача в РГ осуществляется так же, как и в РМ. Содержимое РГРИ передается в РМН по сигналу $РМН:=РГРИ$, вырабатываемому дешифратором поля АДРЕС микрокоманды.

Для задания двух адресов локальной памяти в одном регистре содержимое РР передается в младшие биты (0—7) РМН по сигналу $РМН:=РР$, а старшие биты РМН остаются без изменения.

Передача информации из отдельных частей РГРИ на РА или РВ БА осуществляется по сигналам $РА:=РГ$, $РА:=РР$, $РВ:=РР$, $РВ:=РИ$, $РА:=РИ$.

Схемы регистров ГРИ и ПТУ показаны на рис. 4.2.

4.3. Регистры общего пользования

Регистры общего пользования служат для хранения информации, участвующей в процессе обработки микропрограммы. К ним относятся однобайтовые регистры Л и Д (РЛ и РД), в которые информация заносится с выхода С БА по сигналам $РЛ:=С$, $РД:=С$, вырабатываемым дешифратором поля С микрокоманды.

Передача содержимого регистров на входы РА и РВ БА осуществляется по сигналам $РА:=РЛ$, $РВ:=РЛ$, $РА:=РД$, $РВ:=РД$ дешифраторов полей А и В микрокоманды.

Для организации адреса локальной памяти предусмотрена передача информации из РД в младшие биты РМН по сигналу

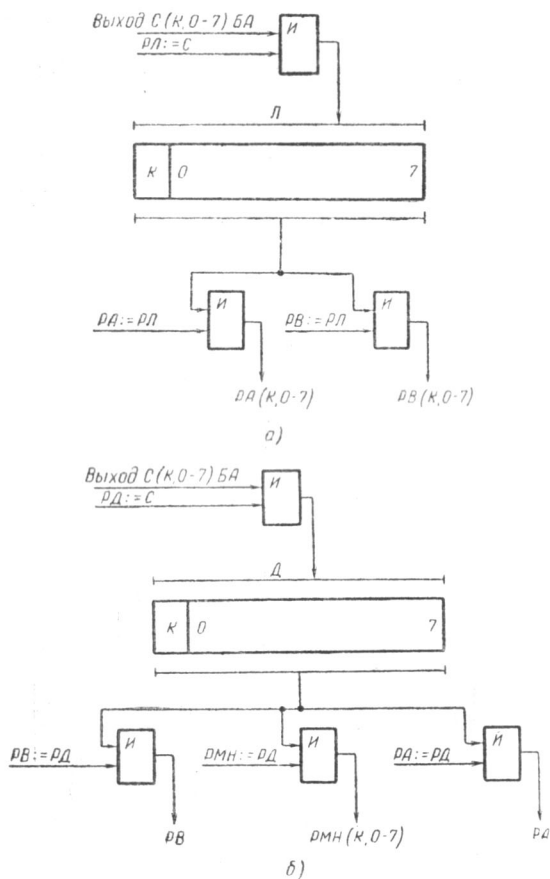


Рис. 4.3. Схема регистров Л (а) и Д (б)

РМН: = РД, вырабатываемому дешифратором поля АДРЕС микрокоманды.

На рис. 4.3 изображены схемы регистров Л и Д. Контроль обоих регистров осуществляется через БА.

4.4. Формирование адреса оперативной памяти

Для задания адресов ячеек оперативной памяти используется расположенный в стойке оперативной памяти регистр МН, содержащий 18 информационных (0—17) и три контрольных бита.

С помощью РМН можно адресоваться к любому типу памяти (основной, локальной, мультиплексной и памяти ключей защиты). В зависимости от объема основной памяти в формировании адреса участвует различное число бит этого регистра. При объеме основной памяти 64К байт используется 16 бит, при объеме 128К байт — 17 бит, при объеме 256К байт — 18 бит.

При обращении к мультиплексной памяти, содержащей 768 и 1536 байт, используются соответственно 10 младших бит (0—9) и 11 бит (0—10) РМН.

Для обращения к локальной памяти, содержащей 256 байт, достаточно восьми младших бит РМН. Для адресации к памяти блока защиты (128 слов) используются семь старших бит (11—17) регистра МН.

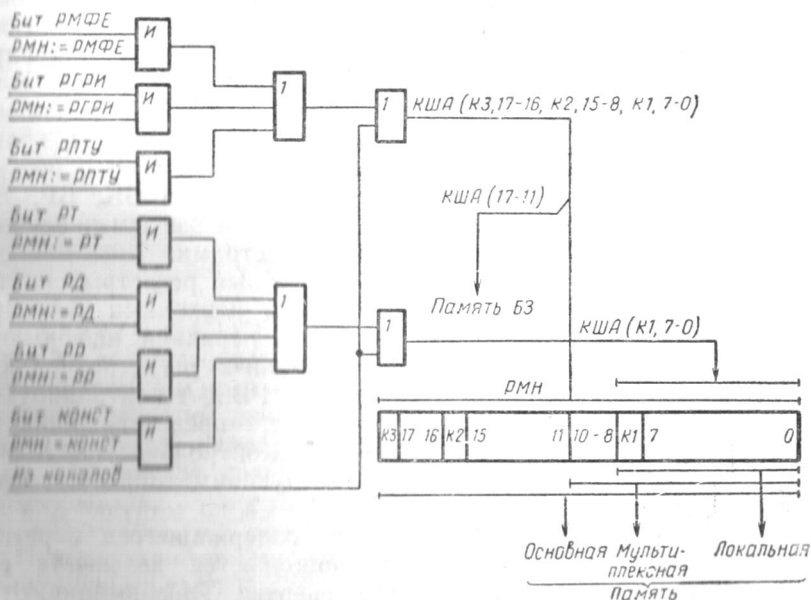


Рис. 4.4. Схема формирования адреса оперативной памяти

В регистр РМН информация заносится:
из адресных регистров МФЕ, ГРИ, ПТУ, откуда берутся два байта с двумя контрольными битами и два бита расширения с выработкой контрольного бита схемой свертки (рис. 4.4). Восемнадцатый бит этих регистров используется для выработки сигнала сбоя по адресации;

из регистров Т, Р, Д, содержащих восемь информационных и один контрольный бит (в младшие биты РМН);

из 44—52-го бита РМК восьмибитовая константа с контрольным битом;

из селекторных каналов 18 информационных (0—17) и три контрольных бита. Восемнадцатый бит из селекторного канала выдается в ОП для выработки сигнала сбоя по адресации.

Передача информации из ЦП в РМН осуществляется по микрооперациям поля АДРЕС. Адрес заносится в РМН парафазно по кодовым шинам адреса (КША).

4.5. Служебные регистры

Во время работы машины в различных блоках генерируется информация, отражающая состояние этих блоков и состояние выполняемой программы, а также информация, указывающая на наличие ошибок в процессе обработки данных. Эта информация на определенных этапах работы машины для принятия решения о последующих действиях должна анализироваться. С целью обеспечения возможности микропрограммного анализа и пересылки этой информации в различные блоки машины триггеры, хранящие ее, скомпонованы в восьмибитовые регистры БК, БР, БС, БД, БЗ и О, называемые служебными регистрами.

Служебные регистры связаны с выходами и входами БА. Правильность передачи информации из служебных регистров на входы РА и РВ БА контролируется по схеме, представленной на рис. 4.5. Контроль осуществляется следующим образом (рис. 4.5). Сумма по модулю два бит слова, содержащегося в регистре, дополняется до нечета схемой свертки. Информация, отражающая результат дополнения, заносится в 9-й (контрольный)

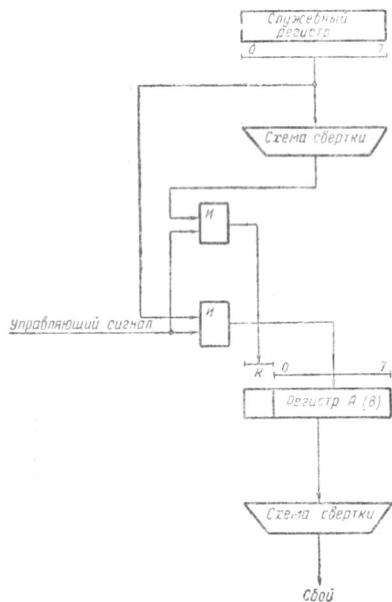


Рис. 4.5. Блок-схема организации контроля передачи информации из служебных регистров на входы БА

бит РА и РВ БА. Имеющиеся на выходах РА и РВ БА схемы свертки суммируют по модулю два биты девятибитового слова, содержащиеся в регистрах. Четная сумма указывает на неверную передачу информации. Индикация об ошибке заносится в регистр ошибок (РО), и происходит прерывание по сбою машины.

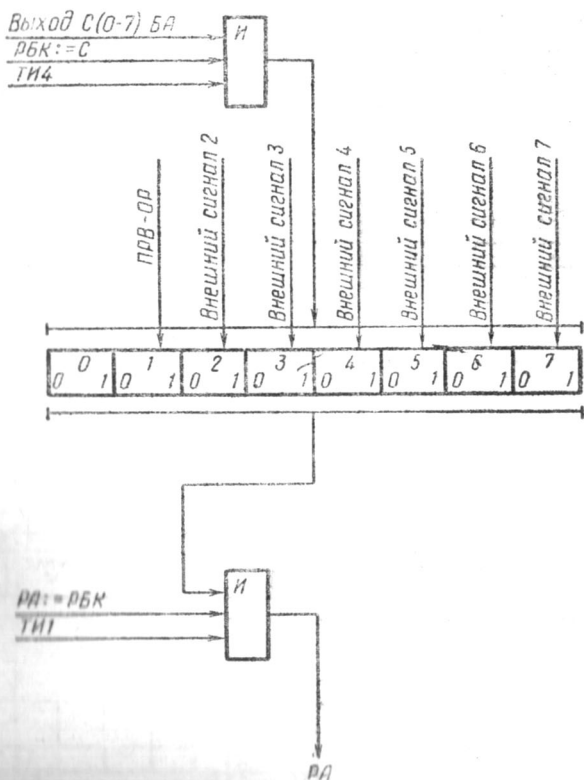


Рис 4.6. Схема регистра БК

Регистр БК (РБК) (рис. 4.6) служит для хранения запросов на внешние прерывания. Триггеры 2—7-го бит регистра хранят запросы на прерывания, поступающие от внешних объектов по шести внешним линиям. Сигналы от таймера и кнопки прерывания на пульте управления заносятся в 0-й и 1-й биты соответственно. Единичное состояние триггера отражает наличие запроса на прерывание. Занесение информации в РБК из БА происходит по сигналу РБК:=С с дешифратора поля С.

Передача информации из РБК на вход РА БА осуществляется по сигналу РА:=РБК.

Запрос на прерывание от таймера устанавливается микропрог-

раммно — занесением в первый бит РБК единицы из БА. Запросы на остальные внешние прерывания устанавливаются аппаратно.

Регистр БР (РБР) (рис. 4.7) служит для хранения маски системы, маски контроля машины и запросов на вводо-выводные прерывания.

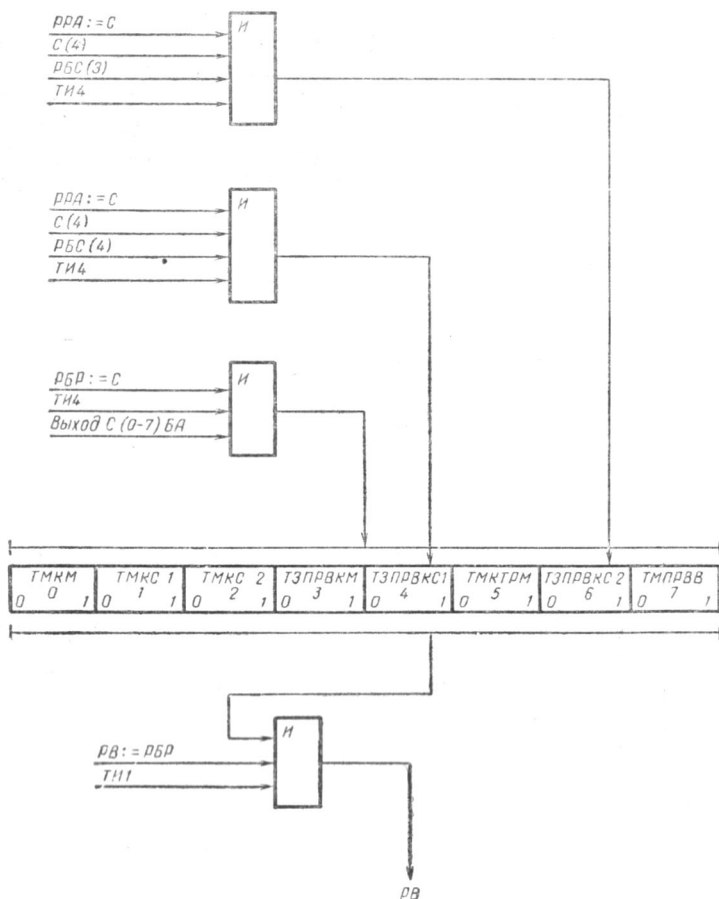


Рис. 4.7. Схема регистра БР

Биты РБР(0) (ТМКМ), РБР(1) (ТМКС1), РБР(2) (ТМКС2), РБР(7) (ТМРРВВ) хранят маски мультиплексного, 1-го и 2-го селекторных каналов и внешних прерываний соответственно (т.е. маску системы). Единичное состояние триггера указывает на разрешение прерывания, а нулевое — на запрещение (маскирование).

Биты РБР(3) (ТЗПРВМК), РБР(4) (ТЗПРВКС1), РБР(6) (ТЗПРВКС2) хранят запросы на прерывания от мультиплексного, 1-го и 2-го селекторных каналов соответственно. Единичное состояние триггера означает наличие запроса.

Бит РБР (5) (ТМКТРМ) хранит маску контроля машины. Единичное состояние триггера указывает на то, что прерывание по сбою разрешено, а нулевое — на то, что прерывание замаскировано.

Все триггеры регистра управляются микропрограммно, т.е. их состояние изменяется засылкой в них информации из БА. Кроме того, триггеры ТЗПРВКС1 и ТЗПРВКС2 устанавливаются в единичное состояние с помощью аппаратных цепей. Установка происходит одновременно с занесением информации в РА первого или второго селекторного канала соответственно. На работу второго селекторного канала указывает единичное состояние РБС(3).

Занесение информации в РБР из БА происходит при наличии в поле С микрокоманды кода микрооперации РБР: = С.

Передача информации из РБР на вход РВ БА происходит при наличии в поле В микрокоманды кода микрооперации РВ: = РБР.

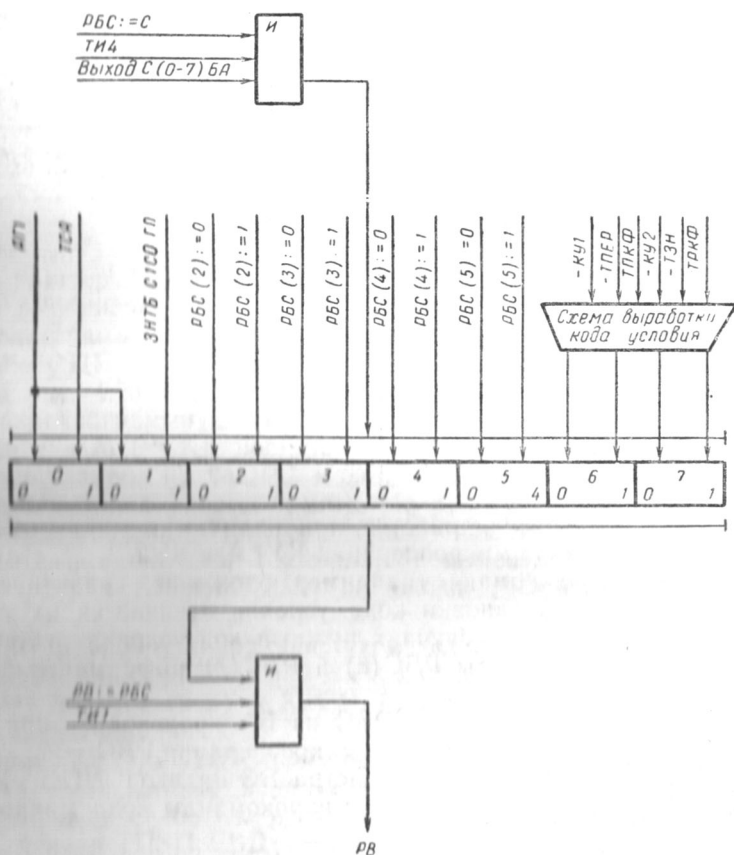


Рис. 4.8. Схема регистра БС

Регистр БС (РБС) (рис. 4.8) служит для хранения признаков сбоя по адресации и защите, универсальных указателей и кода условия.

Бит 0 устанавливается в единичное состояние аппаратно при обнаружении ошибки по адресации при обращении к памяти вслед за установкой триггера сбоя по адресации (ТСА) по синхроримпульсу ТИ4.

Бит 1 устанавливается в единичное состояние аппаратно при нарушении защиты памяти сигналом ЗНТБС1 СОГЛ.

Биты 2—5 являются универсальными индикаторами, используемыми для организации ветвлений в микропрограммах. Устанавливаются в единичное состояние при наличии в микрокоманде в поле УСТАНОВ микроопераций 1БС2, 1БС3, 1БС4, 1БС5, а в нулевое — при наличии микроопераций 0БС2, 0БС3, 0БС4, 0БС5.

Биты 6, 7 хранят код условия. Значение устанавливаемого кода условия зависит от триггеров БА: ТПЕР, ТПКФ, ТЗН, ТРКФ, отражающих результат выполнения действий БА. Алгоритм установки кодов условий двух типов приведен в табл. 4.2.

Таблица 4.2

Тип кода условия	Значения триггеров байта состояния БА	Значения 6-го и 7-го бит РБС
КУ1		
$\Sigma=0$	0РКФ, 0ПЕР	00
$\Sigma<0$	1РКФ, 1ТЗН, 0ПЕР	01
$\Sigma>0$	1РКФ, 0ТЗН, 0ПЕР	10
Переполнение	1ПЕР	11
КУ2		
$\Sigma=0$, нет переноса	0РКФ, 0ПКФ	00
$\Sigma\neq 0$, нет переноса	1РКФ, 0ПКФ	01
$\Sigma=0$, перенос	0РКФ, 1ПКФ	10
$\Sigma\neq 0$, перенос	1РКФ, 1ПКФ	11

Код условия устанавливается при наличии в микрокоманде в поле УСТАНОВ кода микрооперации КУ1 или КУ2.

Для некоторых команд (например, сложение, сравнение десятичное) алгоритм установки кода условия отличается от указанного в табл. 4.2. В этих случаях нужный код условия устанавливается занесением в биты РБС(6) и РБС(7) информации с выхода БА.

Занесение информации в РБС из БА происходит при наличии в поле С микрокоманды кода микрооперации РБС:=С.

Передача информации из регистра БС на вход РВ БА осуществляется при наличии в поле В микрокоманды кода микрооперации РВ:=РБС.

Регистр БД(РБД) (рис. 4.9) служит для хранения информации, отражающей состояние различных блоков машины. Назначение триггеров регистра следующее:

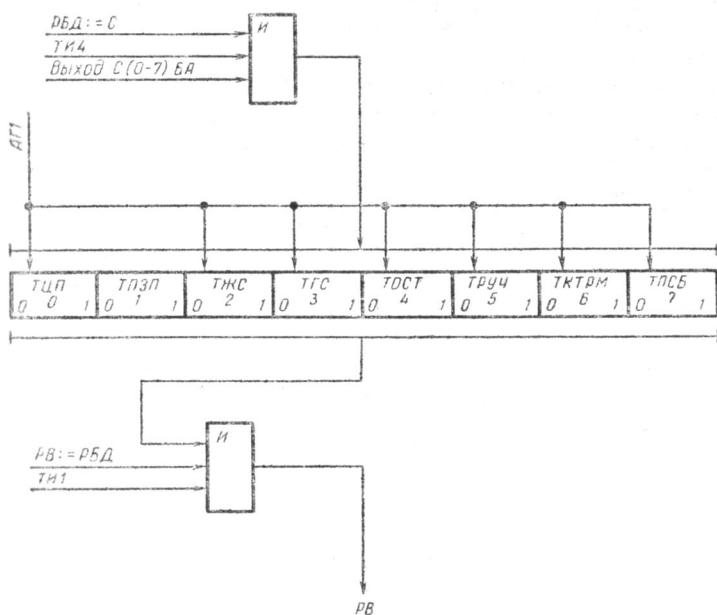


Рис. 4.9. Схема регистра БД

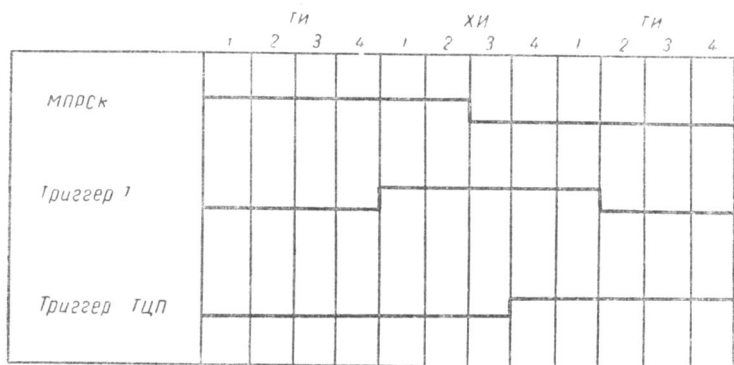
1) триггер ТЦП РБД(0) — единичное состояние его указывает на выполнение мультиплексным или селекторным каналом микропрограммой приостановки (МПРС), нулевое состояние — на работу ЦП.

На рис. 4.10 изображена схема приема процессором запроса на микропрограммную (мультиплексно-селекторную) приостановку (сигнал МПРСК), поступающего из блока общих сигналов каналов. Запрос на МПРС может быть принят только после окончания цикла работы памяти. На работу с памятью указывает единичное состояние триггера конца памяти (ТКЦП). Триггер находится в единичном состоянии от момента передачи адреса в РМН (сигнал КША:=КША) до записи (регенерации) информации.

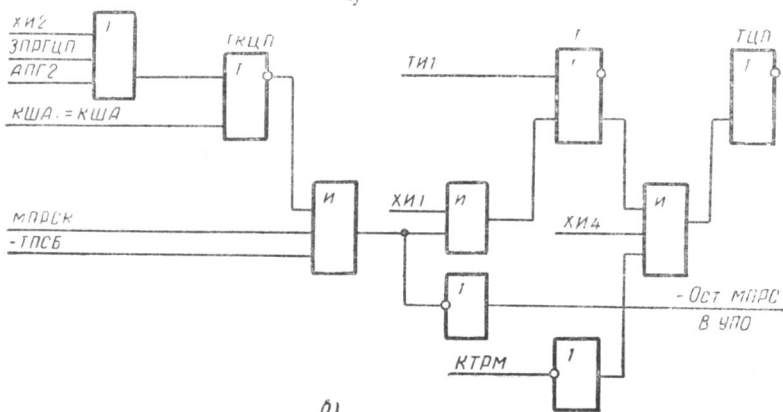
Триггер ТЦП устанавливается в единичное состояние после окончания цикла работы памяти и при отсутствии сигналов ТПСБ (триггер первого сбоя) и КТМ (контроль машины), указывающих на сбой машины.

Сброс триггера ТЦП производится микропрограммно в конце приостановки. Триггер Т необходим для согласования временных параметров сигналов (см. временную диаграмму на рис. 4.10);

2) триггер ТПЗП РБД(1) устанавливается в единичное состояние при нажатии на пульте управления кнопки «Загрузка». Единичное состояние указывает на выполнение процессором последова-



а)



б)

Рис. 4.10. Прием процессором запроса на микропрограммную приостановку: а — временная диаграмма приема запроса; б — схема приема запроса

тельности действий по загрузке начальной программы. В конце микропрограммы загрузки начальной программы триггер микропрограммно сбрасывается. В нулевое состояние триггер устанавливается также при нажатии на пульте управления кнопок «Питание вкл.» и «Гашение»;

3) триггер ТГС РБД(3) находится в единичном состоянии во время выполнения микропрограммы гашения системы. Сброс и установка триггера производится микропрограммно;

4) триггер ТЖС РБД(2) — единичное состояние этого триггера указывает на ждущее состояние ЦП. Состояние триггера определяется 14-м битом текущего ССП. Управляется микропрограммно;

5) триггер ТОСТ РБД(4) — установка триггера в единичное состояние влечет за собой переход ЦП в остановленное состояние перед выборкой очередной команды;

6) триггер ТОСТ (рис. 4.11) устанавливается в единичное состояние: при нажатии на пульте кнопки «Останов»; после оконча-

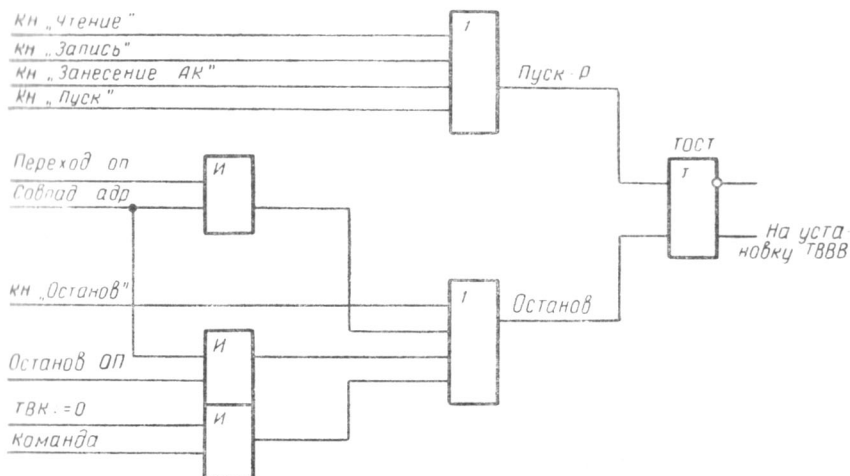


Рис. 4.11. Управление триггером ТОСТ

ния выполнения команды в режиме КОМАНДА; при совпадении адреса в РМН и адреса, набранного на тумблерном наборе пульта в случае установки переключателя «Сравнение адресов» в положение ОСТАНОВ ОП или ПЕРЕХОД ОП; в конце пультowych микропрограмм чтения, записи, занесения адреса команды (АК).

В нулевое состояние триггер устанавливается аппаратно с помощью следующих кнопок на пульте управления: «Пуск», «Чтение», «Запись», «Занесение АК»;

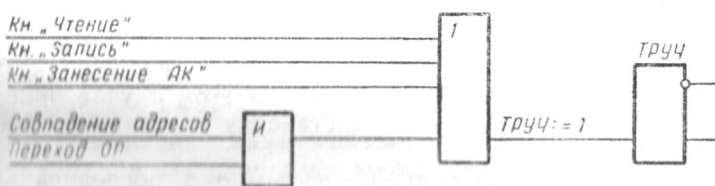


Рис. 4.12. Цепи управления триггером ТРУЧ

7) триггер ТРУЧ РБД (5) (рис. 4.12) находится в единичном состоянии во время выполнения процедур реализации пультowych режимов чтения, записи, занесения адреса команды, а также при совпадении адреса в РМН и адреса, набранного на тумблерном наборе пульта в случае установки переключателя «Сравнение адресов» в положение ПЕРЕХОД ОП. При совпадении адресов это влечет за собой выполнение микропрограммы занесения адреса команды, которая считывает информацию с переключательного на-

бора «Адрес команды» и заносит ее в РМФЕ в качестве нового адреса команды.

В конце пультовых микропрограмм триггер микропрограммно сбрасывается;

8) триггер ТКТРМ РБД(6) служит для занесения в адресный регистр ПП (РАПП) фиксированного адреса 0008 первой микрокоманды микропрограммы обработки сбоя машины. В единичное состояние он устанавливается при обнаружении сбоя машины (сигнал КТРМ) (рис. 4.13); сбрасывается по сигналу СБРОС РО (сброс регистра ошибок) или синхроимпульсом ТИ1;

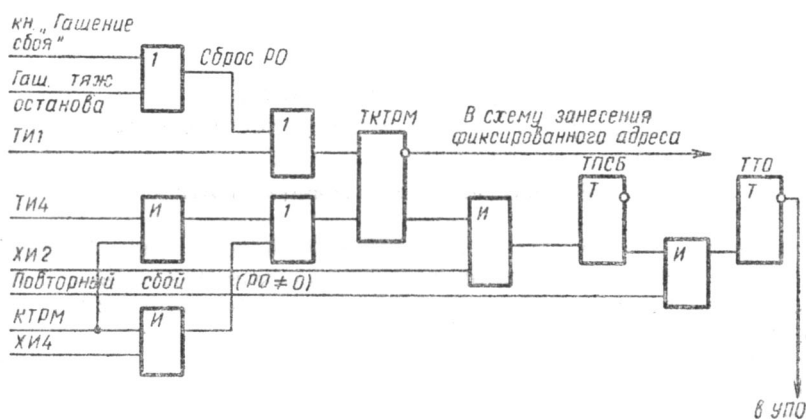


Рис. 4.13. Схема обработки сбоя машины

9) триггер ТПСБ РБД(7) устанавливается в единичное состояние при появлении сбоя. Единичное состояние триггера указывает, что схемами контроля обнаружен сбой. В этом случае прием запросов на микропрограммную приостановку из каналов блокируется (см. рис. 4.10).

Сбрасывается триггер микропрограммно в последней микрокоманде команды ЗАГРУЗИТЬ ССП после того, как будет выполнена супервизорная программа обработки машинной ошибки. При обнаружении повторной ошибки во время выполнения программы обработки машинной ошибки происходит тяжелый останов.

Занесение информации в РБД из БА происходит при наличии в поле С микрокоманды кода микрооперации РБД: = С.

Передача информации из регистра БД на вход РВ БА происходит при наличии в поле В микрокоманды микрооперации РВ: = РБД.

Регистр О (РО) служит для фиксирования сбоев в машине.

Регистр БЗ (РБЗ) является информационным регистром блока защиты.

Глава 5

ОПЕРАТИВНАЯ ПАМЯТЬ И БЛОК ЗАЩИТЫ ОСНОВНОЙ ПАМЯТИ

5.1. Структура оперативной памяти

Как уже отмечалось, в состав оперативной памяти процессора входят три типа логически самостоятельных блоков:

ОП — основная память емкостью 64К, 128К и 256К байт;

ЛП — локальная память емкостью 256 байт;

МП — мультиплексная память емкостью 768 и 1536 байт.

Время такта обращения к любому типу памяти составляет 1 мкс. За это время из памяти читается (записывается) четно-нечетная пара байт (18 бит, каждый байт с контрольным битом).

Оперативная память со своими информационным и адресным регистрами представлена на рис. 5.1. Информационным регистром для всех типов памяти является двухбайтовый регистр ИЗ (РНЗ). Возможно использование этого регистра как двух отдельных однобайтовых (РН и РЗ). При обращении ко всем типам памяти используется один и тот же адресный регистр МН (РМН).

Цикл обращения к памяти разбит на два такта.

Цикл чтения содержит следующие такты:

ЧТЕНИЕ — происходит считывание информации по адресу, находящемуся в РМН, и занесение ее в РНЗ;

РЕГЕНЕРАЦИЯ — содержимое РНЗ записывается в память по тому же адресу. Этот такт необходим потому, что такт ЧТЕНИЕ разрушает информацию в адресуемой ячейке памяти.

Цикл записи содержит следующие такты:

СТИРАНИЕ — отличается от такта ЧТЕНИЕ только тем, что прочитанная информация не заносится в РНЗ;

ЗАПИСЬ — аналогичен такту РЕГЕНЕРАЦИЯ.

Так как в каждый данный момент возможно обращение только к одной из трех типов памяти (основной, мультиплексной или локальной), то с целью экономии оборудования мультиплексная и локальная память выполняются встроенными в основной блок. По-

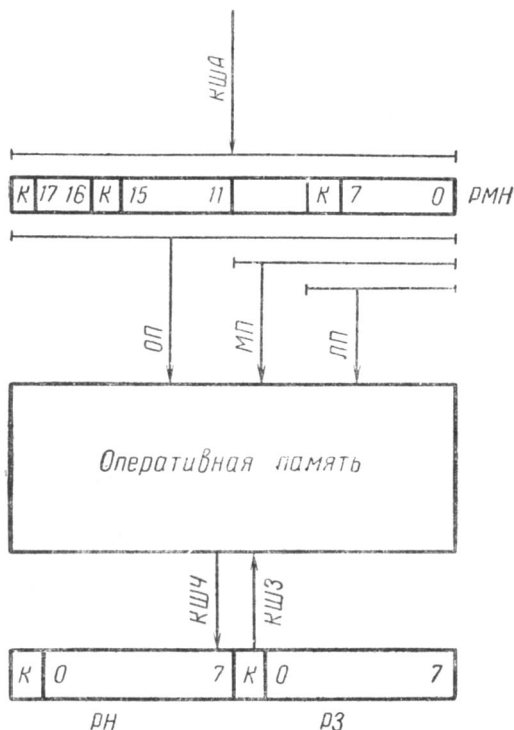


Рис. 5.1. Блок-схема оперативной памяти

этому, помимо команд ЧТЕНИЕ, ЗАПИСЬ, из процессора в оперативную память по одной из трех шин поступает признак типа памяти.

Адрес локальной памяти расположен в битах 0—7 РМН, мультиплексной — в битах 0—10. Начальный адрес мультиплексной памяти (в двоичном коде) 001 0000 0000.

Структура и расположение информации в локальной и мультиплексной типах памяти показаны на рис. 5.2.

В локальной памяти размещены:

16 регистров общего назначения;
четыре регистра (64-битовых) для операций с плавающей запятой;

текущее ССП (64 бита);

буфер команды (8 бит), который используется для хранения кода операции выполняемой команды;

буфер прерывания (8 бит), хранящий информацию для ввод-выводного прерывания;

область каналов — шесть (64-битовых) регистров, служащих для хранения информации регистров процессора при мультиплексной приостановке и селекторном зацеплении;

	0	1	2	3	4	5	6	7
0X				0				
1X				1				
2X				2				
3X				3				
4X				4				
5X				5				
6X				6				
7X				7				
8X				8				
9X				9				
AX				A				
BX				B				
CX				C				
DX				D				
EX				E				
FX				F				
УСУ	Код операции	Инд. А	Инд. В	Инд. С	Инд. D	Инд. E	Инд. F	Инд. G
00X	Расширенный адрес следующего КСК	← Адрес следующего КСК →			Состояние канала	СЧД	Расширенный адрес данных	← Адрес в ячейке работы
01X								← Адрес в ячейке работы
02X								
03X								

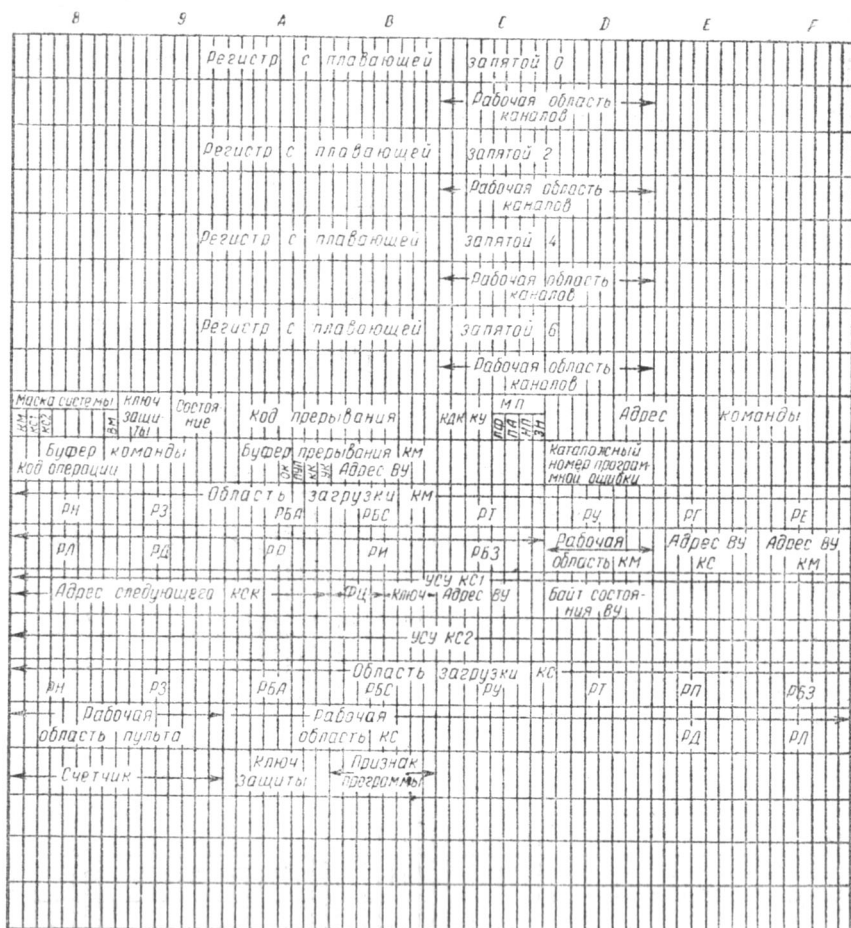
Рис. 5.2. Структура и расположение информации

рабочая память процессора — 16 регистров (32-битовых), используемых для хранения различной информации микропрограмм для выполнения которых недостаточно аппаратных регистров процессора.

5.2. Блок связи с оперативной памятью

В состав блока связи с оперативной памятью (БО) входят: РНЗ, регистр-дублер Н131 (РН131), узлы занесения информации в РНЗ, узлы контроля и индикации содержимого РНЗ.

РНЗ состоит из двух однобайтовых регистров Н и З, имеющих по одному контрольному биту.



в локальной и мультиплексной типах памяти

При чтении информации из памяти по кодовым шинам чтения (КШЧ) в РНЗ поступают два байта информации с двумя контрольными битами. Управляет занесением информации из памяти в РНЗ (рис. 5.3) сигнал РАЗР. ЗН (разрешение занесения), вырабатываемый при чтении информации каналом или процессором, если нет сбоя по адресации (сигнал ТСА), и при стирании информации (сигнал ТСТ), если возникает сбой по защите (сигнал ТНК) и нет сбоя по адресации.

Таким образом, разрешается занесение информации из ОП в РНЗ при стирании информации в запрещенной области. Следующий такт ЗАПИСЬ восстановит содержимое испорченной ячейки.

Информация в РНЗ из ОП заносится по синхросигналу ТИ43 или СИ43, если происходит обмен информацией между селекторным каналом и основной памятью.

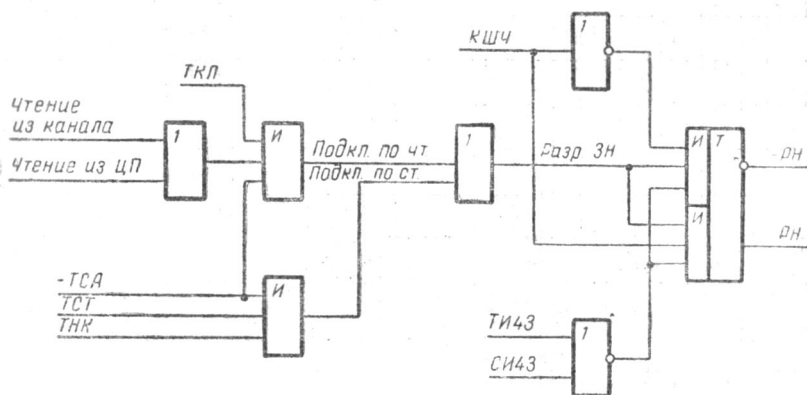


Рис. 5.3. Схема занесения информации в один бит РН из оперативной памяти

С выхода арифметическо-логического блока в РН или РЗ передается байт информации с контрольным битом в зависимости от наличия сигнала $РН:=С$ или $РЗ:=С$ с дешифратора поля $С$ микрокоманды, который управляет занесением информации в РН или РЗ соответственно. Информация в РН или РЗ с выхода БА заносится по синхросигналу ТИ4 (рис. 5.4).

При наличии сбоя по защите сигнал ОТКЛ.АК (отключение БА и каналов от РНЗ) запрещает занесение информации с выхода БА в РНЗ, чтобы не испортить информацию в ОП.

При работе каналов с ОП информация в РНЗ заносится или побайтно, или по два байта, в зависимости от наличия сигналов $РЗ:=РЗК$ и $РН:=РНК$, управляющих занесением информации из каналов (рис. 5.5).

При обмене информацией между селекторным каналом и основной памятью информация в РНЗ заносится по синхросигналу СИ2, а при обмене информацией с мультиплексным каналом — по синхросигналу ТИ3. Если во время работы ОП с каналами возник

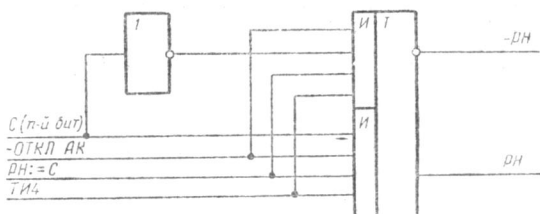


Рис. 5.4. Схема занесения информации в РН из БА

кает сбой по защите, то сигнал ОТКЛ.АК запрещает занесение информации в РНЗ.

Вся информация, заносимая РНЗ, с его выходов заносится в РН131.

Если происходит селекторная аппаратная приостановка, то начинается обмен информацией между селекторным каналом и основной памятью. Для того чтобы сохранить информацию, которая была в РНЗ перед селекторной приостановкой, по синхросигналу

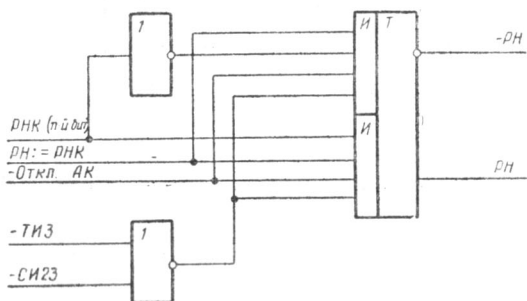


Рис. 5.5. Схема занесения информации в РН из каналов

СИ1 перекрываются входы РН131. По окончании работы с каналом по сигналу РНРЗ:=РН1РЗ1, поступающему из блока синхронизации, информация из РН131 возвращается в РНЗ (рис. 5.6).

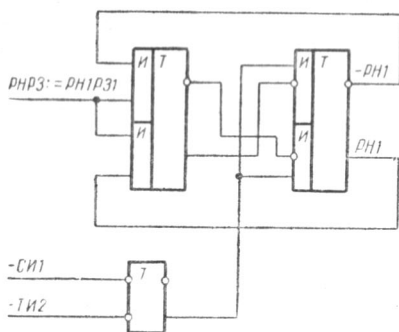


Рис. 5.6. Схема связи РН и РН1

Информация с выходов РНЗ поступает на входы БА в каналы и по кодовым шинам записи (КШЗ) в оперативную память.

Информация, поступающая в РНЗ, контролируется на нечетность и наличие двухпроводного сбоя. При наличии сбойной информации в РН, РЗ выдаются сигналы сбоя, поступающие в регистр ошибок (РО).

Сигналы сбоев вырабатываются при любом занесении неверной информации в РНЗ из ОП или БА. Чтобы не было повторных сбоев от одной и той же сбойной информации и не устанавливался триггер тяжелого останова, в начале микропрограммы обработки машинной ошибки по синхросигналу ХИ2 сбрасывается триггер, разрешающий выработку сигнала сбоя (рис. 5.7).

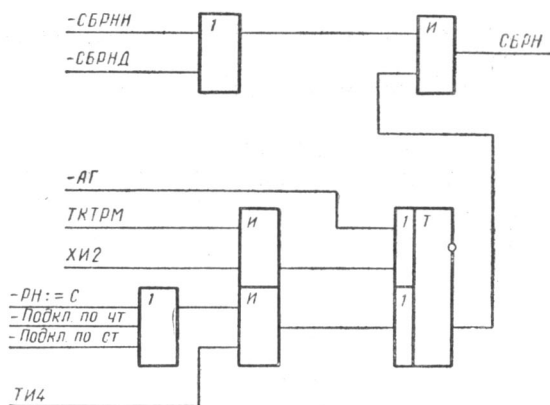


Рис. 5. 7. Схема выработки сигнала сбоя РН (СБРН)

Сигналы сбоя НЗ, возникшие при работе с селекторным каналом, не поступают в РО, а выдаются в канал.

5.3. Назначение и принцип работы блока защиты памяти

ЭВМ ЕС-1020 может использоваться в мультипрограммном режиме работы, т. е. в режиме одновременной обработки нескольких программ. В этом режиме в основной памяти может одновременно находиться кроме супервизора несколько рабочих программ. Но так как под воздействием одной рабочей программы может меняться информация другой рабочей программы или же супервизора, в машине предусмотрены средства защиты основной памяти, разрешающие супервизору доступ к любой области памяти (возможность модернизации любой программы), а рабочей программе — доступ только к отведенным ей областям памяти.

Основная память разделена на «страницы» емкостью 2048 байт каждая. Страницам соответствуют ключи памяти — четырехбитовые слова, в каждом из которых могут быть записаны числа от 0 до 15.

На рис. 5.8. приведена структура основной памяти емкостью 16К байт, разби-

0 - 2047	0	} Ключи памяти
2048 - 4095	5	
4096 - 6143	3	
6144 - 8191	7	
8192 - 10239	5	
10240 - 12287	3	
12288 - 14335	7	
14336 - 16383	5	

Рис. 5.8. Структура (страницы) основной памяти

тая на «страницы» с соответствующими ключами. Ключи памяти хранятся в запоминающем устройстве ключей памяти (ЗУКП), емкость которого (128 слов) рассчитана на максимальный объем основной памяти 256К байт.

Каждый раз при обращении к основной памяти происходит автоматическое обращение (автоматический режим) к памяти ЗУКП и чтение ключа памяти, соответствующего данной странице.

К средствам защиты, кроме ключа памяти, относится четырехбитовый ключ защиты, размещаемый в битах 8-11 текущего ССП. Доступ к основной памяти гарантируется только в том случае, если ключ памяти соответствует ключу защиты, т.е. если они поразрядно равны или ключ защиты в текущем ССП равен нулю.

В машине ЕС-1020 возможны два способа защиты памяти: защита по «чтению и записи» и защита по «записи».

При защите первым способом независимо от того, какое обращение происходит к данной странице (только чтение или же запись новых данных), при несовпадении ключей вырабатывается программное прерывание.

При защите вторым способом программное прерывание может произойти только при несовпадении ключей и попытке модифицировать данные страницы.

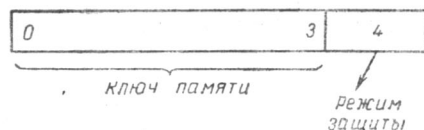


Рис. 5.9. Структура ключа памяти

Для задания способа защиты страницы служит в ключе памяти дополнительный пятый бит (рис. 5.9). Единичное состояние триггера режима защиты соответствует защите по «чтению и записи», нулевое — по «записи».

Заполнение памяти ЗУКП, т.е. установка ключей памяти и проверка кода ключа памяти, выполняется двумя командами: **УСТАНОВИТЬ КЛЮЧ ПАМЯТИ** и **ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ**.

С помощью привилегированной команды **УСТАНОВИТЬ КЛЮЧ ПАМЯТИ** можно изменить любую ячейку ЗУКП, т.е. ключ памяти любой страницы основной памяти. Эта команда имеет формат *RR*.

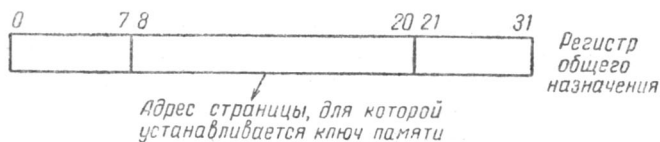


Рис. 5.10. Структура регистра общего назначения

Требуемый ключ памяти содержится с 24-го по 28-й бит универсального регистра, адрес которого указывается полем *R1*. Бит 28 используется для задания режима защиты.

Адрес страницы записывается в универсальном регистре, адрес которого указывается полем *R1*. Для задания адреса используются 24 младших бита 32-битового регистра общего назначения. При адресации целых страниц по 2048 байт каждая адрес страницы содержится только в битах 8—20 (рис. 5.10).

5.4. Состав и режим работы блока защиты

Средства защиты основной памяти реализуются с помощью блока защиты, в состав которого входят:

запоминающее устройство ключей памяти (ЗУКП), рассчитанное на максимальный объем основной памяти (128 шестибитовых слов);

регистр блока защиты (РБЗ), старшие биты которого являются информационным регистром ЗУКП и хранят ключ памяти (0—4), а младшие — ключ защиты (4'—7);

узел сравнения (УСБЗ), вырабатывающий сигнал при несопадении ключа памяти с ключом защиты;

узел выработки контрольного бита (УВКР) при передаче информации из РБЗ на вход РА БА;

Структура ячейки ЗУКП показана на рис. 5.11.

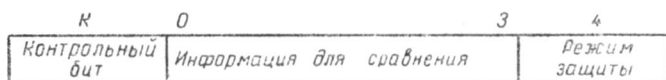


Рис. 5.11. Структура ячейки ЗУКП

Схема связей блока защиты с другими блоками процессора представлена на рис. 5.12.

Блок защиты может работать в одном из следующих трех режимов: режиме чтения из ЗУКП, режиме записи в ЗУКП и автоматическом.

Режим чтения из ЗУКП необходим для выполнения команды ПРОЧИТАТЬ КЛЮЧ ПАМЯТИ. По этой команде информация из определенной ячейки ЗУКП передается в регистр общего назначения. В микропрограмме этой команды содержимое РБЗ (К—4) — ключ памяти — передается на вход РА арифметического блока для последующей передачи в регистр общего назначения. Так как в РБЗ имеется два бита (4 и 4'), которые передаются на вход четвертого бита РА, то их выбор производится с помощью триггера ТЦП, находящегося в нулевом состоянии при выполнении любых команд. Только в том случае, если канал разделит оборудование вычислительного устройства (мультиплексно-селекторная приостановка), этот триггер переводится в единичное состояние.

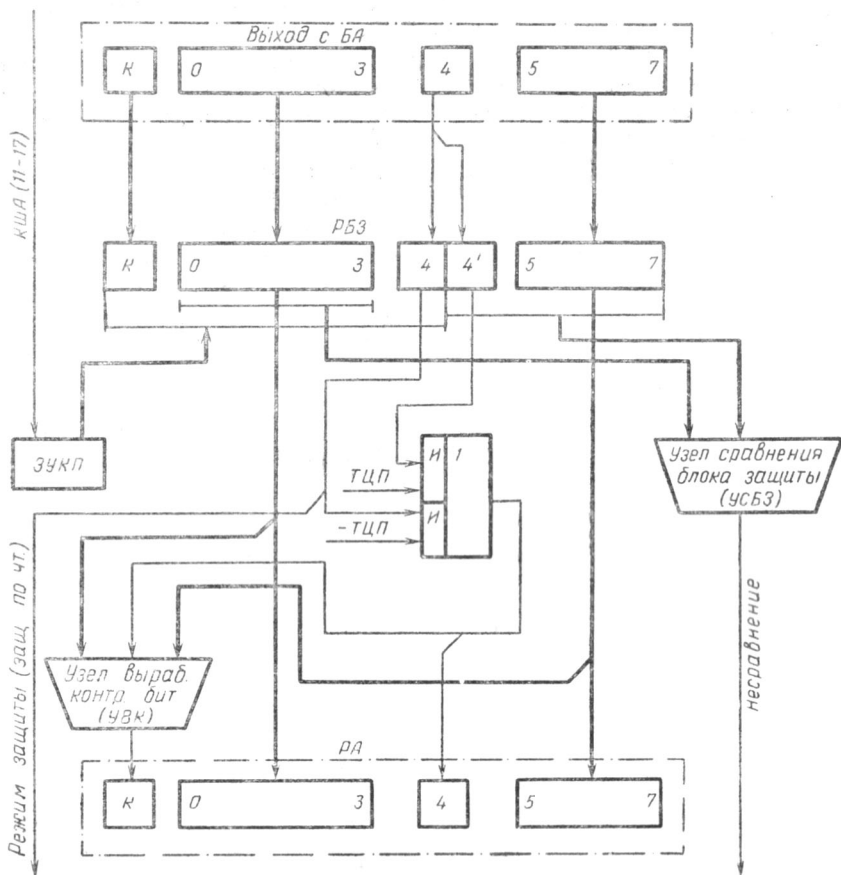


Рис. 5.12. Схема связей блока защиты с другими блоками процессора

Режим записи в ЗУКП необходим для выполнения команды УСТАНОВИТЬ КЛЮЧ ПАМЯТИ. По этой команде информация из регистра общего назначения передается в ячейку ЗУКП. В микропрограмме этой команды содержимое регистра общего назначения через выход БА передается в РБЗ (К,0—3) и записывается в ЗУКП.

Автоматический режим включается при каждом обращении к основной памяти и заключается в чтении ключа памяти из ячейки ЗУКП, адрес которой определяется старшими битами адресного регистра основной памяти (РМН). Старшие биты РМН (17—11) определяют страницу, к которой идет обращение.

Далее ключ памяти сравнивается с ключом защиты текущего ССП, который находится в битах 4'—7 РБЗ. Сравнение ключей

происходит в узле сравнения. При их несравнении вырабатывается сигнал НЕСРАВНЕНИЕ, свидетельствующий о том, что ключи не совпали и возможно нарушение защиты, т. е. программное прерывание. Окончательная выработка запроса на программное прерывание происходит в блоке прерываний, где анализируются режим защиты и способ обращения к памяти.

На время мультиплексно-селекторной приостановки ключ защиты текущего ССП необходимо передать в локальную память, т. е. освободить биты 4'—7 РБЗ для приема ключа защиты из управляющего слова устройства (УСУ). Передача ключа защиты в локальную память производится микропрограммой мультиплексно-селекторной приостановки через вход РА арифметическо-логического устройства с помощью триггера ТЦП, находящегося в единичном состоянии.

При работе селекторного канала во время аппаратной приостановки передача ключа защиты из РБЗ (4'—7) блокируется и он передается из канала по отдельным шинам РКЗК (0—3).

Глава 6

АРИФМЕТИЧЕСКО-ЛОГИЧЕСКИЙ БЛОК

6.1. Назначение и принцип работы

Арифметическо-логический блок (БА) предназначен для побитовой обработки или передачи в регистры процессора и каналов данных и адресов, подаваемых на его входы (А и В).

Блок может выполнять следующие операции:

сложение двоичное — $(A+B)$ ДВ;

сложение десятичное — $(A+B)$ ДЕС;

вычитания двоичные — $(A-B)$ ДВ, $(B-A)$ ДВ;

вычитания десятичные — $(A-B)$ ДЕС, $(B-A)$ ДЕС;

операцию ИЛИ — $(A \vee B)$;

операцию И — $(A \& B)$;

суммирование по модулю два — $(A \oplus B)$;

операцию ИЛИ над данными входа А и инвертированными данными входа В — $(A \vee \bar{B})$;

операцию И над инвертированными данными входа А и данными входа В — $(\bar{A} \& B)$;

передачу на выход БА данных со входа А — А ТРАНЗИТ;

передачу на выход БА данных со входа В — В ТРАНЗИТ;

сдвиг данных входа В вправо на один бит — СДВИГ В ВПРАВО;

сдвиг данных входа В влево на один бит — СДВИГ В ВЛЕВО.

При выполнении операции $(A+B)$ ДВ на выходе БА формируются результат суммирования кодов, поданных на входы А и В, и значения входного (межбайтового) переноса. Специальные схемы запоминают значение выходного переноса и переполнение разрядной сетки БА.

Признаком переполнения является неравенство значений переносов из первого и нулевого бит байта.

Примеры.

1. К числу 22 (00010110) прибавить число 92 (01011100) с учетом входного переноса

		Биты								
		0	1	2	3	4	5	6	7	
(+22)		0	0	0	1	0	1	1	0	— вход А
	+									
(+92)		0	1	0	1	1	1	0	0	— вход В
(+114 или 115)		0	1	1	1	0	0	1	X	— окончательный результат
		0	0	0	1	1	1	0	0	— межбитовые переносы (X — значение межбайтового переноса)
	↑									
Перенос из бита 0 (межбайтовый)										

Переполнение отсутствует (переносы из бит 0 и 1 равны, оба — нули).

2. К числу 100(01100100) прибавить число 100(01100100) с учетом входного переноса:

		Биты								
		0	1	2	3	4	5	6	7	
(+100)		0	1	1	0	0	1	0	0	— вход А
	+									
(+100)		0	1	1	0	0	1	0	0	— вход В
	↑									
Перенос из бита 0 (межбайтовый)										

Имеется место переполнение разрядной сетки (переносы из бит 0 и 1 не равны).

3. К числу — 114(10001110) прибавить число 22(00010110) с учетом входного переноса:

		Биты								
		0	1	2	3	4	5	6	7	
(-114)		1	0	0	0	1	1	1	0	— вход А
	+									
(+22)		0	0	0	1	0	1	1	0	— вход В
(-92 или -93)		1	0	1	0	0	1	0	X	— окончательный результат
		0	0	0	1	1	1	1	0	— межбитовые переносы (X — значение межбайтового переноса)
	↑									
Перенос из бита 0 (межбайтовый)										

Переполнение отсутствует (переносы из бит 0 и 1 равны, оба — нули).

Отрицательные результаты на выходе БА представлены в дополнительном коде.

При выполнении операции (A—B) ДВ или (B—A) ДВ на выходе БА формируется результат суммирования кода входа А (или В) с обратным кодом входа В (или А) и инвертированным значением входного переноса, имитирующим добавление единицы в младший разряд вычитаемого (превращение обратного кода в дополнительный). При выполнении этих операций запоминается инвертированное значение выходного переноса. Это необходимо для того, чтобы при вычитании чисел, содержащих несколько байт, единица добавлялась только в младший разряд младшего

байта, а при обработке последующих байт дважды инвертированное (при запоминании и учете) значение переноса несло истинную информацию.

Пример. Из числа 256 (0000 0001 0000 0000) нужно вычесть число 100 (0000 0000 0110 0100). Для простоты рассматривается двухбайтовое двоичное представление этих чисел.

Межбайтовому переносу предварительно определено нулевое значение. Обработка в БА младших байт операндов:

	Биты	
	0 1 2 3 4 5 6 7	
	0 0 0 0 0 0 0 0	— вход А
	1 0 0 1 1 0 1 1	— инвертированное значение входа В
	1 0 0 1 1 1 0 0	— выход БА
	0 0 0 0 0 0 1 1 1	— межбитовые переносы (инвертированное значение межбайтового входного переноса)
↑		
Межбайтовый перенос		

В триггере хранения межбайтового переноса запоминается его инвертированное значение — «1».

При обработке в БА старших байт операндов получается следующее:

	Биты	
	0 1 2 3 4 5 6 7	
	0 0 0 0 0 0 0 1	— вход А
	1 1 1 1 1 1 1 1	— инвертированное значение входа В
	0 0 0 0 0 0 0 0	— межбитовые переносы
	↑	— Инвертированное значение триггера хранения межбайтового переноса
	0 0 0 0 0 0 0 0	— выход БА

Переполнение отсутствует (значения переносов из бит 0 и 1 равны).

В результате получено двоичное число 0000 0000 1001 1100, соответствующее десятичному +156.

При вычитании 100 из 256 будет иметь место следующее.

Обработка младших байт операндов:

	Биты	
	0 1 2 3 4 5 6 7	
	0 1 1 0 0 1 0 0	— вход А
	1 1 1 1 1 1 1 1	— инвертированное значение входа В
	1 1 1 1 1 1 1 1	— межбитовые переносы
↑		— Инвертированное значение входного переноса
Межбайтовый перенос		
(в схеме хранения запоминается «0»)		

0 1 1 0 0 1 0 0 — выход БА

Обработка старшего байта:

	Биты	
	0 1 2 3 4 5 6 7	
	0 0 0 0 0 0 0 0	— вход А
	1 1 1 1 1 1 1 0	— инвертированное значение входа В
	0 0 0 0 0 0 0 1	— переносы
	↑	— Значение межбайтового переноса (инвертированное значение схемы хранения)
	1 1 1 1 1 1 1 1	— выход БА

Переполнение отсутствует.

В результате получено двоичное число 1111 1111 0110 0100, т. е. дополнительный код числа — 156.

При выполнении операций $(A+B)$ ДЕС, $(A-B)$ ДЕС, $(B-A)$ ДЕС на входы БА должны быть поданы десятичные данные, представленные в коде 8421.

Сложение и вычитание десятичных чисел производятся также, как и двоичных, но для получения правильного результата необходима дополнительная обработка данных, учитывающая избыточность кода 8421 и заключающаяся в том, что в каждую десятичную цифру одного из слагаемых вводится дополнительная цифра 6: при вычитании — в процессе получения дополнения вычитаемого, а при сложении — принудительно, перед началом суммирования.

Эта цифра выводится из каждой десятичной цифры либо в процессе суммирования (если был перенос из данной тетрады), либо принудительно при формировании окончательного результата (если не было переноса из данной тетрады).

Примеры.

1. К числу 38 (00111000) прибавить число 46 (01000110):

		Биты	
		0 1 2 3 4 5 6 7	
(38)		0 0 1 1 1 0 0 0	— вход В
(46)	+	0 1 0 0 0 1 1 0	— вход А
(66)	+	0 1 1 0 0 1 1 0	— добавление 66 к данным входа А
		0 1 1 1 1 1 0 X	— межбитовые переносы (X — значение межбайтового переноса)
Перенос из старшей тетрады (межбайтовый)	↑	↑	↑
		1 1 1 0 0 1 0 X	— результат суммирования (A+B+66)

Нулевое значение переноса из старшей тетрады свидетельствует о необходимости коррекции данной тетрады.

		1 1 1 0 0 1 0 X	— результат суммирования
	+	1 0 1 0	— вычитание 6 из старшей тетрады
(84 или 85)		1 0 0 0 0 1 0 X	— окончательный результат (выход БА)

2. Из числа 38 (00111000) вычесть число 46 (01000110):

		Биты	
		0 1 2 3 4 5 6 7	
(38)		0 0 1 1 1 0 0 0	— вход А
(46)		0 1 0 0 0 1 1 0	— вход В
	→	0 0 1 1 1 0 0 1 1	— межбитовые переносы
Перенос из старшей тетрады (межбайтовый)	↑	↑	↑
		1 1 1 0 0 1 0	— результат суммирования А и \bar{B} .

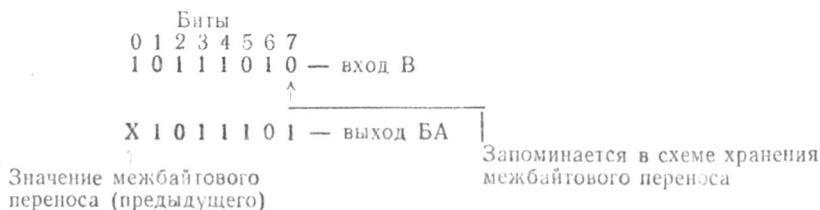
Нулевое значение переноса из старшей тетрады свидетельствует о необходимости коррекции данной тетрады.

$$\begin{array}{r}
 11110010 \text{ — результат суммирования} \\
 + \quad 1010 \quad \text{— вычитание 6 из старшей тетрады} \\
 \hline
 (92) \quad 10010010 \text{ — окончательный результат (выход БА)}
 \end{array}$$

Число 92 является дополнительным кодом результата вычитания числа 46 из 38.

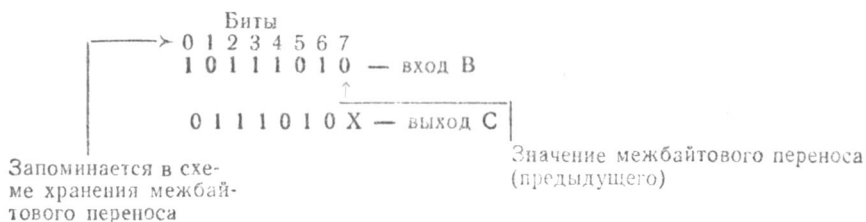
При выполнении операции СДВИГ В ПРАВО на выходе БА в битах 1—7 содержится значение бит 0—6 данных со входа В, в бите 0—состояние схемы, хранящей значение предыдущего межбайтового переноса. Бит 7 данных входа В запоминается схемой хранения межбайтового переноса.

Пример.



При выполнении операции СДВИГ В ЛЕВО на выходе БА в битах 0—6 содержится значение бит 1—7 данных со входа В, бите 7—состояние схемы, хранящей значение межбайтового переноса. Бит 0 данных входа В запоминается схемой хранения межбайтового переноса.

Пример.



При выполнении каждой микрокоманды обязательно выполняется одна из перечисленных операций, определяемых полем ФУНКЦИЯ микрокоманды. Четырехбитовое поле ФУНКЦИЯ микрокоманды может либо непосредственно определять одну из 15 операций, осуществляемых БА при выполнении данной микрокоманды (прямое задание операции), либо один из его кодов (0001) может указывать на то, что при выполнении данной микрокоманды операция определяется содержимым специального регистра (РКФ), входящего в состав БА (косвенное задание операции БА).

РКФ получает информацию заранее — при выполнении одной из предшествующих микрокоманд. В поле УСТАНОВ этой микро-

команды задается код (10111), разрешающий передачу в РКФ содержимого поля КМЛ, в котором задается один из 15 кодов операции БА.

При выполнении операций, заданных прямо или косвенно, производится раздельное запоминание и использование информации, необходимой для обработки последующих байт (межбайтовый перенос) и анализа результата обработки (равен ли байт результата нулю или нет).

Так, например, межбайтовый перенос, возникающий при выполнении операции $(A+B)ДВ$, заданной прямо, запоминается в триггере переноса прямой функции (ТППФ), а при выполнении той же операции, заданной косвенно,— в триггере переноса косвенной функции (ТПКФ).

Значение входного переноса при выполнении операции $(A+B)ДВ$, заданной прямо, определяется состоянием триггера ТППФ, а заданной косвенно — состоянием триггера ТПКФ.

Необходимость задания типа операции прямо или косвенно обусловлена следующими соображениями: 1) при обработке информации длиной более полуслова из-за необходимости модификации адресов операндов перед чтением очередной пары байтов информации происходит чередование обработки байтов данных и адресов операндов. Прямое и косвенное задание типа операции при обработке данных и адресов операндов позволяет как бы разделить потоки информации; 2) возможность косвенного задания типа операции позволяет более рационально строить микропрограммы, так как предварительное задание кода операции в РКФ приводит к возможности использования части, а иногда и целой микропрограммы при реализации близких по логике команд.

6.2. Структура блока

Арифметическо-логический блок состоит из ряда взаимосвязанных узлов, управляемых полями микрокоманды или внутренними сигналами, определяемыми полем ФУНКЦИЯ микрокоманды.

В блоке нет узлов обработки данных, предназначенных для выполнения какой-либо одной операции. При выполнении любой из 15 возможных операций каждый узел блока выполняет какую-то функцию, причем некоторые узлы могут выполнять одну из нескольких возможных.

Результатом взаимодействия узлов является выполнение заданной операции. Так, например, при выполнении операции двончного сложения один из узлов производит первое полусуммирование, другой — поразрядные переносы, а третий — второе полусуммирование. При выполнении операции И узел, производивший в операции $(A+B)ДВ$ первое полусуммирование, осуществляет операцию И.

Узлы, не принимающие участия в формировании результата выполняемой операции, передают данные, не изменяя их.

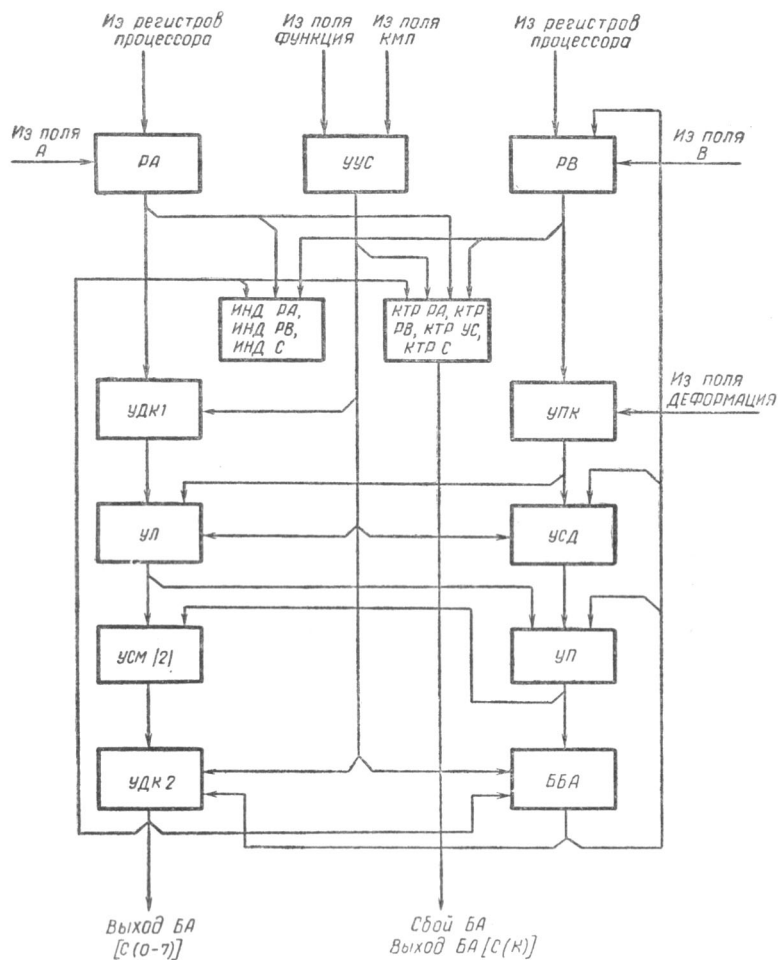


Рис. 6.1. Структурная схема БА

В состав БА (рис. 6.1) входят следующие узлы: RA и RB — входные девятибитовые регистры (восемь бит — информационных, один бит — контрольный);

УПК — узел перекосов, предназначенный для передачи информации из RB на входы узлов обработки данных потетрадно прямо или накрест, только старшей тетрады или только младшей тетрады прямо или накрест, а также для передачи тетрад с перекосом;

УУС — узел управляющих сигналов, осуществляющий выработку серии сигналов, управляющих работой узлов обработки данных арифметическо-логического блока. Управляется непосред-

ственно четырехбитным полем ФУНКЦИЯ микрокоманды и косвенно четырехбитным полем КМЛ (константа младшая) микрокоманды;

УДК1 — первый десятичный корректор, при выполнении операции десятичного сложения производящий добавление числа 6 (0110) к обеим тетрадам данных входного регистра РА. В остальных случаях информация регистра РА передается без изменения;

УЛ — узел логический, реализующий ряд логических операций над данными, поступающими с выходов первого десятичного корректора и узла перекосов;

УСД — узел сдвига вправо, предназначенный для подготовки данных при выполнении операций сдвига вправо и влево (сдвиг вправо на два бита и передача без изменений соответственно), а также для организации одного из операндов при выполнении операций сложения и вычитания (передача без изменений или инвертирование данных соответственно). При выполнении остальных операций на выходе узла сдвигов вправо вырабатывается нулевая информация;

УП — узел переносов, служащий для организации межразрядных и межбайтового переносов, необходимых при выполнении операций сложения и вычитания.

В этом узле производится также сдвиг информации влево на один бит (при выполнении операций сдвига вправо и влево), для чего на выходе УЛ вырабатывается нулевая информация, а сдвигаемые данные, поступившие на РВ, имитируют одно из слагаемых. При выполнении операций, не упомянутых в данном пункте, на выходе УП образуется нулевая информация;

УСМ|2| — узел суммирования по модулю два информации, подаваемой на его входы;

УДК2 — второй десятичный корректор, предназначенный для корректировки результатов сложения и вычитания десятичных чисел;

ББА — байт состояния арифметическо-логического блока, содержащий ряд запоминающих элементов, хранящих результаты побайтной обработки информации и схем организации сигналов, необходимых для выполнения операций сложения (вычитания) и сдвига;

КТР РА, КТР РВ, КТР УС, КТР С — узлы контроля, осуществляющие контроль четности входных данных, контроль правильности работы узлов управляющих сигналов и обработки данных. В узле КТР С формируется также контрольный бит байта результата. При обнаружении ошибки вырабатываются сигналы сбоя, которые передаются в регистр ошибок (РО) процессора;

ИНД РА, ИНД РВ, ИНД С — узлы индикации состояния РА и РВ и информации на выходе БА.

Обработка информации в БА производится следующим образом.

Подлежащие обработке операнды принимаются на РА и РВ и

хранятся там в течение машинного такта обработки информации (рис. 6.2). Принятая информация контролируется на четность.

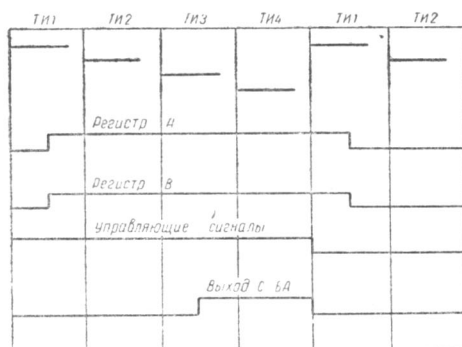


Рис. 6.2. Временная диаграмма работы арифметическо-логического блока

УС вырабатывает серию управляющих сигналов, соответствующих заданной операции. Правильность выработки этих сигналов проверяется в узле контроля КТР УС. При наличии сбоя по второму синхрипульсу данного такта (Tи2) производится установка триггера РО процессора, фиксирующего ошибку БА. Информация из РА (поток А) подается на первый десятичный корректор УДК1. При выполнении операций сложения десятичных чисел к обеим тетрадам байта добавляется цифра 6, в остальных случаях данные проходят через узел без изменений.

Информация с выхода УДК1 поступает на логический узел, куда подается также информация из РВ (поток В), прошедшая через узел перекосов. В зависимости от заданной операции в логическом узле производится одна из возможных логических операций над данными потоков А и В. Так, например, при выполнении операций сложения производится сложение по модулю два (первая полусумма). В операциях сдвига на выходе логического узла формируется нулевая информация.

Информация с выхода узла перекосов подается также на узел сдвига вправо, в котором в зависимости от выполняемой операции производится одно из следующих действий: инвертирование; подготовка данных для выполнения операции сдвига вправо; передача без изменения; организация нулевой информации, имеющая место лишь тогда, когда операция над данными полностью заканчивается в логическом узле (при выполнении всех логических операций и операций пересылки данных регистра А или В).

Результирующие данные из логического узла и узла сдвига вправо поступают в узел переносов (УП). В соответствии с условиями образования переносов информация на выходе УП органи-

зается следующим образом. Единичное значение данного бита узла переносов формируется, если для предыдущего бита значение первой полусуммы равно нулю, а значение одного из слагаемых — единице или значение полусуммы равно единице и в этот бит сформирована единица переноса из предшествующего бита. Информация с выхода УЛ воспринимается узлом переноса как первая полусумма, а с выхода УСД — как одно из слагаемых.

При выполнении арифметических операций на выходе узла переносов формируются значения межбитовых переносов.

При выполнении операций сдвига на выходе УЛ формируется нулевая информация и в соответствии с приведенным алгоритмом на выходе узла переносов формируется сдвинутая влево на один бит информация с выхода УСД.

При выполнении остальных операций нулевая информация формируется на выходе УСД и тогда в соответствии с приведенным алгоритмом на выходе узла переносов организуется нулевая информация.

Перенос из старшего (нулевого) бита фиксируется в специальном триггере, входящем в состав ББА. Состояние этого триггера учитывается как входящий перенос. Данные с выхода узла переносов суммируются по модулю два с результирующими данными узла логических операций в узле суммирования по модулю два.

При выполнении операций сложения и вычитания это — второе полусуммирование; для остальных же операций эта процедура не изменяет конечного результата, полученного либо в узле переносов (операции сдвига), либо в логическом узле (все остальные операции).

При выполнении операций над десятичными числами результирующие данные при необходимости корректируются (вычитается 6) во втором десятичном корректоре, являющемся выходным узлом арифметическо-логического блока.

Выходные данные контролируются в узле контроля выхода (КТР С). Там же производится формирование контрольного бита результирующего байта.

6.3. Описание узлов блока

Узел управляющих сигналов (УУС) состоит (рис. 6.3.) из четырехбитового регистра прямой функции (РПФ), хранящего на время одного машинного такта содержимое поля ФУНКЦИЯ микрокоманды; двух пятибитовых регистров косвенной функции (РКФ и РКФ ДОП), хранящих в течение нескольких машинных тактов код операции, заданной полем КМЛ микрокоманды, и код управления способом подачи тетрад информации из РВ на узлы обработки данных; триггера режима (ТРЕЖ); дешифратора управляющих сигналов, вырабатывающего заданную серию управляющих сигналов, и коммутатора, подающего на вход дешифратора управляющих сигналов содержимое регистра прямой

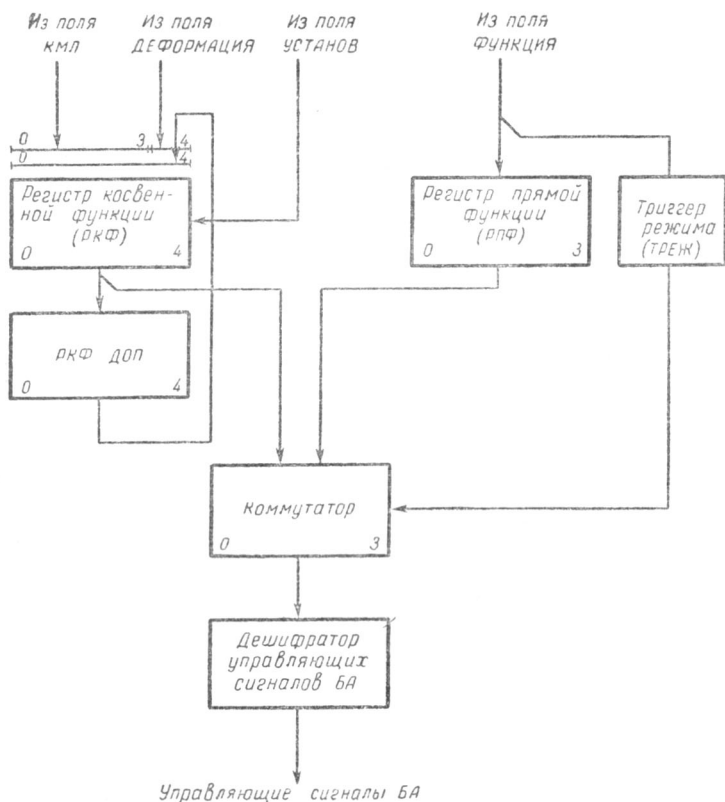


Рис. 6.3. Схема формирования управляющих сигналов

или косвенной функции в соответствии с состоянием триггера режима.

На вход РПФ поступают биты 9—12 РМК. Занесение информации с шин регистра микрокоманды производится по первому синхрои импульсу такта ТИ1.

На вход четырех бит РКФ (0—3) поступают потенциалы с битов 49—52 РМК.

Занесение информации из РМК в РКФ производится по второму синхрои импульсу такта при задании в поле УСТАНОВ микрокоманды кода ЗКФ, т. е. при наличии на выходе дешифратора поля УСТАНОВ сигнала РКФ: = КМЛ.

Установка РКФ(4) в единичное состояние производится по второму синхрои импульсу такта при задании в поле ДЕФОРМАЦИЯ В кода ПЕРЕКОС КОСВЕННОЙ ФУНКЦИИ, т. е. при наличии сигнала ПККФ на выходе дешифратора поля ДЕФОРМАЦИЯ В. Установка этого триггера в нулевое состояние производится по второму синхрои импульсу такта при задании в поле

УСТАНОВ микрокоманды кода ЗКФ, т. е. при наличии сигнала РКФ; =КМЛ и отсутствии в поле ДЕФОРМАЦИЯ В кода ПЕРЕКОС КОСВЕННОЙ ФУНКЦИИ.

Дополнительный регистр косвенной функции (РКФ ДОП) служит для хранения содержимого РКФ на время микропрограммной приостановки. Содержимое РКФ переписывается в РКФ ДОП по сигналу $XI1 \& (PVM = PAIP) \& (-TKTRM)$, вырабатываемому в блоке микропрограммного управления. Возврат информации в РКФ производится по сигналу $TI2 \& (PAIP := PVM)$, выработанному в последнем такте приостановки. Триггер режима устанавливается в единичное состояние, определяющее выполнение операции по косвенной функции (КФ), по первому синхронимпульсу такта (ТИ1) при наличии в поле ФУНКЦИЯ микрокоманды кода ВЫПОЛНИТЬ КОСВЕННУЮ ФУНКЦИЮ.

При всех остальных кодах в этом поле триггер режима по первому синхронимпульсу такта устанавливается в нулевое состояние, определяющее выполнение операции по прямой функции (ПФ).

Триггер режима управляет работой коммутатора.

Выходы коммутатора, представляющие собой двоичный четырехбитовый код операции в парафазной форме, поступают на дешифратор управляющих сигналов, который вырабатывает серии следующих управляющих сигналов (см. табл. 6.1).

Таблица 6.1

Операция	Код	Управляющие сигналы
AVB	0000	К,Л,М,И
ВЫИ.КФ	0001	К,И
(A-B) ДЕС	0010	К,Г,Н,У,В,С,И
(A-B) ДВ	0011	К,Г,У,В,С,И
A ТРАНЗИТ	0100	К,Л,И
A&B	0101	К,И
(B-A) ДЕС	0110	К,Г,Н,Х,В,С
(B-A) ДВ	0111	К,Г,Х,В,С
AVB̄	1000	К,Л,Г
B ТРАНЗИТ	1001	К,М
A⊕B̄	1010	Л,М
A&B̄	1011	М
СДВИГ В ПРАВО	1100	Х,В,Т,И
СДВИГ В ВЛЕВО	1101	Х,В,И
(A+B) ДЕС	1110	Л,М,Д,Н,Х,В,И
(A+B) ДВ	1111	Л,М,Х,В

Сигналы К, Л, М, Г определяют функцию, выполняемую логическим узлом. Комбинации этих сигналов позволяют осуществить в этом узле операции И, ИЛИ, СЛОЖЕНИЕ ПО МОДУЛЮ 2 и др.

Сигнал Д присутствует при операции десятичного сложения. При наличии его в УДК1 происходит добавление шести к каждой тетраде данных, поступающих из регистра А.

Сигнал Н присутствует при выполнении операций десятичного сложения и вычитания и управляет работой УДК2. При наличии его (и дополнительных условий) производится вычитание шести из каждой тетрады байта результата.

Сигналы Х, Т, У, определяют функцию, выполняемую узлом сдвигов вправо.

При наличии сигнала Х в узле происходит либо передача данных без изменения (при отсутствии сигнала Т), либо подготовка данных для выполнения операции сдвига вправо (при наличии сигнала Т).

При наличии управляющего сигнала У производится инвертирование данных.

Сигнал В участвует во всех арифметических операциях и операциях сдвига. Он управляет работой байта состояния арифметическо-логического блока, разрешает занесение межбайтового переноса в соответствующие триггеры (выходящий перенос) и в младший бит узла переносов (входящий перенос).

Сигнал С присутствует при операциях вычитания. Он управляет формированием значений переносов. При его наличии входящий и выходящий переносы принимают инвертированное значение относительно истинного.

Сигнал И вырабатывается для целей контроля. Он уравнивает четность количества управляющих сигналов с четностью числа единиц в коде операции, что для кода 0001, который может быть использован в целях диагностики, не выполняется.

Кроме перечисленных, в дешифраторе вырабатывается сигнал П7: =ТП, по которому седьмому биту узла переносов присваивается значение межбайтового переноса, хранящегося в специальных триггерах, или значение, определяемое полем УСТАНОВ. Этот сигнал вырабатывается при выполнении всех арифметических операций и операций сдвига влево.

Входной регистр А. Десятибитовый входной регистр А (РА) предназначен для приема и хранения на время обработки информации, источник которой задается полем А микрокоманды. При этом дешифратор поля А вырабатывает один из сигналов РА: =РН, РА: =РЗ, РА: =РЛ и т. д., по которому разрешается передача данных из РН, РЗ, РЛ и др. соответственно (рис. 6.4).

Передача данных в РА разрешается только при отсутствии сигнала РА: =РА, который вырабатывается при задании в поле А кода 0000. В этом случае информация в РА сохраняется. Занесение информации в РА производится по первому задержанному синхроимпульсу такта (Т11З). Если источником информации является один из десятибитовых регистров, то информация заносится во все биты РА, включая контрольный.

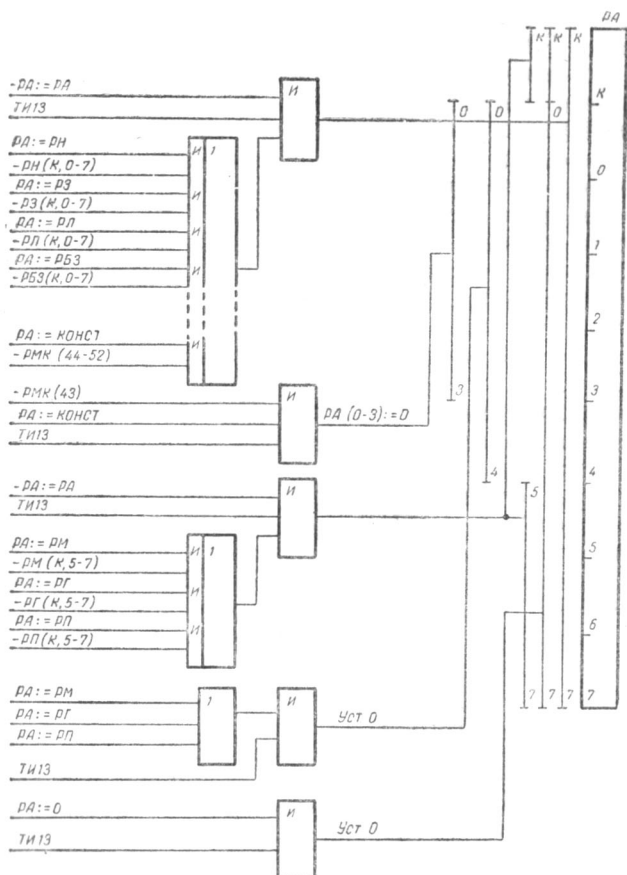


Рис. 6.4. Схема занесения информации в РА

Если источником информации является константа из поля микрокоманды (РМК (44—52)), то по сигналу $РА := КОНСТ$ возможны два варианта занесения информации: в зависимости от состояния поля М микрокоманды (РМК (42—43)) заносится либо длинная константа (восемь информационных бит и один контрольный — РМК (44—52)), либо короткая константа (четыре информационных и один контрольный — РМК (44,49—52)). При нулевом состоянии 43-го бита РМК вырабатывается сигнал $РА (0-3) := 0$, по которому 0, 1, 2, 3-й биты устанавливаются в нулевое состояние независимо от содержимого 45—48-го бит РМК.

Если источником информации является один из трехбитовых регистров М, Г, П, то в 5, 6, 7-й и контрольный биты РА передаются данные из этих регистров, а остальные биты по одному из сигналов $РА := РМ$, $РА := РГ$, $РА := РП$ устанавливаются в нулевое состояние.

В поле А микрокоманды можно задать занесение нулевой информации в РА. В этом случае по сигналу $РА:=0$ с дешифратора поля А в информационные биты РА заносятся нули, контрольный бит устанавливается в единичное состояние.

Входной регистр В. Девятибитовый регистр В (РВ) предназначен для приема и хранения на время обработки информации, источник которой задается полем В микрокоманды. Дешифратор поля В вырабатывает один из сигналов $РВ:=РН$, $РВ:=РЗ$ и т. д., по которому соответственно разрешается передача данных из регистров Н, З и т. д. Схема занесения данных в РВ аналогична схеме занесения в РА.

Передача данных в РВ разрешается только при отсутствии сигнала $РВ:=РВ$, который вырабатывается при задании в поле В кода 00000. В этом случае информация в РВ сохраняется. Занесение информации в РВ производится по первому задержанному синхрипульсу такта (ТИЗ).

Если источником информации является один из регистров процессора, перечисленных в поле В микрокоманды, то производится занесение информации во все биты РВ, включая контрольный, причем информация из всех регистров блока связи с внешними устройствами (РРЗ, РР4, РР5, РР6, РР9, РРБ, РРВ, РРГ, РРД, РРП, РРЕ, РРП) подается по одному общему сигналу $РВК(К,0-7)$ и ее занесение разрешается общим сигналом $РВ:=РВК$, вырабатываемым дешифратором поля В микрокоманды в случае, если источником информации является один из перечисленных регистров.

Если источником информации является константа из поля микрокоманды (РМК (44—52)), то возможны два варианта занесения информации. В зависимости от состояния поля М микрокоманды (РМК (42, 43)) заносится либо длинная, либо короткая константа.

Если заносится короткая константа, при нулевом состоянии 43-го бита регистра микрокоманды вырабатывается сигнал $РВ(0-3):=0$, по которому 0, 1, 2, 3-й биты РВ устанавливаются в нулевое состояние независимо от содержимого 45—48-го бит РМК.

В поле В микрокоманды можно задать занесение нулевой информации в РВ. В этом случае по сигналу $РВ:=0$ дешифратора поля В в информационные биты РВ заносятся нули. Контрольный бит при этом устанавливается в единичное состояние.

На входной регистр В можно подать также байт состояния арифметическо-логического блока. В этом случае по сигналу $РВ:=БА$ из дешифратора поля В в информационные биты регистра РВ производится передача данных из триггеров байта состояния.

Узел перекосов (УПК) позволяет осуществить следующие режимы передачи информации из РВ в узлы обработки:

ПРЯМО — информация через узел перекосов проходит без изменения;

НАКРЕСТ — старшая тетрада информации из РВ подается на выход младшей тетрады узла перекосов, младшая тетрада информации РВ — на выход старшей тетрады узла перекосов;

МЛАДШИЕ ПРЯМО — на выход узла перекосов передается только младшая тетрада информации из РВ, в старшую тетраду выхода узла перекосов заносятся нули;

СТАРШИЕ ПРЯМО — на выход узла перекосов передается только старшая тетрада информации из РВ, в младшую тетраду выхода узла перекосов заносятся нули;

МЛАДШИЕ НАКРЕСТ — младшая тетрада информации из РВ передается на выход старшей тетрады узла перекосов, в младшую тетраду выхода узла перекосов заносятся нули;

СТАРШИЕ НАКРЕСТ — старшая тетрада информации из РВ передается на выход младшей тетрады узла перекосов, в старшую тетраду выхода узла перекосов заносятся нули;

ПЕРЕКОС — младшая тетрада информации из РВ передается на выход старшей тетрады узла перекосов, на выход младшей тетрады узла перекосов передаются данные, хранящиеся в буфере перекоса. Старшая тетрада информации из РВ запоминается в буфере перекоса для использования ее в последующем такте передачи с перекосом. При передаче с перекосом нескольких байт можно осуществить сдвиг информации влево на одну тетраду;

ПЕРЕКОС КОСВЕННОЙ ФУНКЦИИ — осуществляет предварительное задание режима ПЕРЕКОС (занесение единицы в 4-й бит РКФ). При этом происходит прямая передача информации на вход узлов обработки данных. При выполнении операции, заданной косвенно, режим передачи данных РВ определяется состоянием 4-го бита РКФ. При единичном состоянии этого бита проис-

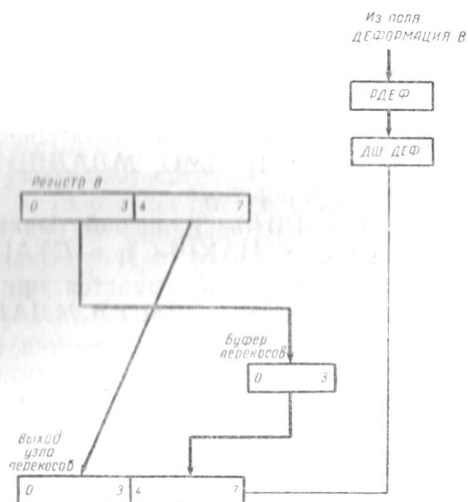


Рис. 6.5. Схема передачи данных в режим ПЕРЕКОС

ходит передача данных с перекосом, при нулевом — режим определяется полем ДЕФОРМАЦИЯ В.

Узел перекосов управляется трехбитовым полем ДЕФОРМАЦИЯ В микрокоманды. Он состоит из трех триггеров, запоминающих на один такт состояние поля ДЕФОРМАЦИЯ В (РДЕФ), дешифратора состояния этих триггеров (ДШ ДЕФ), буфера перекоса и комбинационной схемы, организующей заданную передачу тетрад на выход узла перекоса (рис. 6.5).

Занесение информации в РДЕФ осуществляется по каждому первому синхроримпульсу такта (ТИ1). Дешифратор состояния этих триггеров управляет способом подачи тетрад на выход узла перекосов в соответствии с заданным режимом.

На выход старшей тетрады комбинационной схемы узла перекосов (ПК (0—3)) может быть передана либо соответствующая, либо младшая тетрада РВ, либо нули. В соответствии с этим вырабатываются следующие управляющие сигналы: $ПК(0-3) := РВ(0-3)$, $ПК(0-3) := РВ(4-7)$, $ПК(0-3) := 0$.

Сигнал $ПК(0-3) := РВ(0-3)$ вырабатывается при передаче данных из РВ без изменения (ПРЯМО), при передаче только старшей тетрады (СТАРШИЕ ПРЯМО) и при предварительном задании перекоса.

Сигнал $ПК(0-3) := РВ(4-7)$ вырабатывается при передаче данных из РВ в режимах НАКРЕСТ, МЛАДШИЕ НАКРЕСТ, ПЕРЕКОС, а также при выполнении перекоса, заданного предварительно в регистр косвенной функции ($-КФ \& РКФ(4)$).

Сигнал $ПК(0-3) := 0$ вырабатывается при передаче данных из РВ в режимах МЛАДШИЕ ПРЯМО И СТАРШИЕ НАКРЕСТ.

На выход младшей тетрады комбинационной схемы узла перекосов может быть передана либо младшая, либо старшая тетрада РВ, либо нули, либо данные, хранившиеся в буфере перекосов. В соответствии с этим вырабатываются следующие управляющие сигналы: $ПК(4-7) := РВ(4-7)$, $ПК(4-7) := РВ(0-3)$, $ПК(4-7) := 0$, $ПК(4-7) := БПК$.

Сигнал $ПК(4-7) := РВ(4-7)$ вырабатывается при передаче данных из РВ в режимах ПРЯМО, МЛАДШИЕ ПРЯМО и при предварительном задании перекоса.

Сигнал $ПК(4-7) := РВ(0-3)$ вырабатывается при передаче данных из РВ в режимах НАКРЕСТ и СТАРШИЕ НАКРЕСТ.

Сигнал $ПК(4-7) := 0$ вырабатывается при передаче данных из РВ в режимах СТАРШИЕ ПРЯМО и МЛАДШИЕ НАКРЕСТ.

Сигнал $ПК(4-7) := БПК$ вырабатывается при передаче данных из РВ в режиме ПЕРЕКОС, а также при выполнении перекоса, заданного предварительно в регистр косвенной функции ($-КФ \& РКФ(4)$).

Параллельно с сигналом $ПК(4-7) := БПК$ вырабатывается сигнал $БПК := РВ(0-3)$, по которому старшая тетрада информации из РВ запоминается в буфере перекоса для использования ее в последующем такте передачи данных с перекосом.

Буфер перекося состоит из двух четырехбитовых триггерных регистров. В одном из них (РБПК1) производится запоминание старшей тетрады данных РВ по четвертому синхриимпульсу такта (ТИ4) при наличии сигнала БПК := РВ(0—3). По сигналу БПК := 0 из дешифратора поля УСТАНОВ микрокоманды производится установка этого регистра в нулевое состояние. В другом (РБПК2) по каждому первому синхриимпульсу такта (ТИ1) осуществляется перепись данных из первого регистра. При наличии управляющего сигнала ПК(4—7) := БПК эти данные передаются на выход младшей тетрады узла перекося.

Логический узел (УЛ) предназначен для выполнения ряда логических операций над данными. Он управляется сигналами К, Л, М, Г.

Информация на выходах (единичном и нулевом) каждого бита УЛ определяется управляющими сигналами и входной информацией и формируется в соответствии с выражениями:

$$\begin{aligned}
 +L(P) &= K \& DK(P) \& PK(P) \vee L \& DK(P) \& (\neg PK(P)) \vee M \& (\neg DK(P)) \\
 &\quad \& PK(P) \vee G \& (\neg DK(P)) \& (\neg PK(P)); \\
 -L(P) &= (\neg K) \& DK(P) \& PK(P) \vee (\neg L) \& DK(P) \& (\neg PK(P)) \\
 &\quad \vee (\neg M) \& (\neg DK(P)) \& (PK(P)) \vee (\neg G) \& (\neg DK(P)) \& (\neg PK(P)),
 \end{aligned}$$

где P принимает значения от 0 до 7.

При выполнении той или иной операции в БА задается определенный набор управляющих сигналов.

Так, например, при выполнении операции сложения задаются сигналы Л и М и в УЛ производится суммирование по модулю два (первая полусумма).

При выполнении логических операций задается набор сигналов, позволяющий осуществить заданную логическую операцию, которая и заканчивается в логическом узле. В этом случае данные через остальные узлы проходят без изменения.

Узел сдвига вправо (УСД) представляет собой комбинационную схему, организующую передачу данных узла перекося на вход узла переносов без изменения, инверсных или подготовленных для выполнения операции сдвига вправо.

Работой узла управляют сигналы Х, У, Т. При наличии управляющего сигнала Х (и отсутствии сигнала Т) данные передаются без изменения. В случае управляющего сигнала У производится инвертирование данных. При наличии управляющего сигнала Т 0—5-й биты данных в узле сдвигаются вправо на два бита. В 1-й бит УСД из узла БАА поступает информация ($\pm П(0)$), учитывающая ранее установленное значение межбайтового переноса или текущее значение поля УСТАНОВ микрокоманды. В 0-й бит УСД заносится значение 7-го бита данных. При отсутствии управляющих сигналов Х, У, Т, т. е. при выполнении всех операций, кроме сложения, вычитания и сдвигов, на выходе узла организуется нулевая информация.

Узел переносов (УП) предназначен для организации межбитовых и межбайтового переносов.

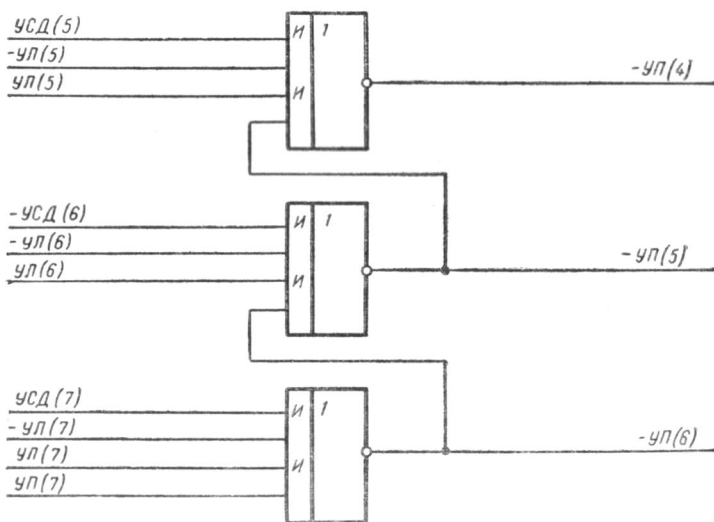
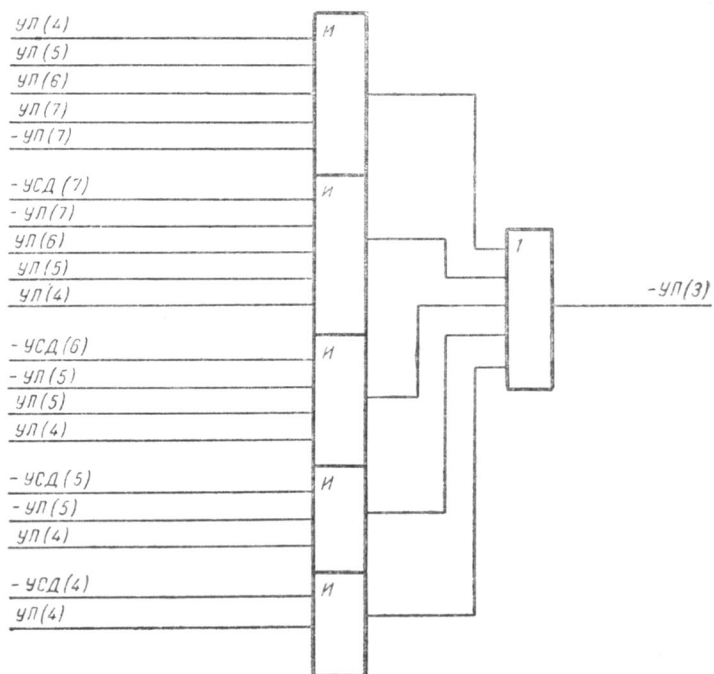


Рис. 6.6. Фрагмент схемы формирования межбитовых переносов (биты 6—3)

На вход УП поступают данные с выходов УЛ и УСД (полусумма и слагаемое из РВ соответственно), а также входной перенос, определяемый состоянием схемы хранения межбайтового переноса или полем УСТАНОВ микрокоманды.

Схема формирования межбитовых переносов состоит из двух одинаковых частей, организующих переносы в пределах одной тетрады (рис. 6.6). Входным переносом для младшей тетрады является сигнал $\pm П(7)$, учитывающий либо ранее установленное значение межбайтового переноса, либо текущее значение поля УСТАНОВ. Входным переносом для старшей тетрады служит выходящий перенос младшей тетрады ($П(3)$). Биты 6, 5, 4, 2, 1, 0 узла переносов формируются последовательно, старшие же биты тетрад — параллельно.

Первый десятичный корректор (УДК1) представляет собой комбинационную схему, управляемую сигналом Д. Организует либо передачу данных потока А без изменения, либо при наличии сигнала Д производит добавление шести (0110) к обоим тетрадам данных.

Содержимое бит 3 и 7 при добавлении шести (0110) не изменится и поэтому, минуя УДК1, подается на вход УЛ.

Поскольку в машине предусматривается контроль десятичных данных и при нецифровых комбинациях выполнение команды прекращается, узел реализует добавление шести только к возможным цифровым комбинациям.

Второй десятичный корректор (УДК2) является выходной ступенью арифметическо-логического блока. На вход его поступают данные с выхода узла суммирования по модулю два. Режим работы определяется сигналом Н из УУС, который в сочетании со значением выходного переноса из каждой тетрады данных определяет, нужна ли корректировка данной тетрады. При необходимости корректировки комбинационная схема осуществляет вычитание шести из данной тетрады.

Биты 3 и 7 при коррекции результата не изменяются и поэтому, минуя УДК2, с выхода узла суммирования по модулю два поступают на выход С БА, где имеется сигнал С(5) УПЛ, принимающий единичное значение при наличии хотя бы одной единицы в битах 0—5 выхода С БА. Этот сигнал принимает участие в формировании сбоя по адресации.

Байт состояния арифметическо-логического блока (ББА) предназначен для запоминания информации, являющейся результатом побайтовой обработки данных, и выработки сигналов $\pm П(7)$ и $\pm П(С)$. Данные ББА используются либо при обработке последующих байт, либо при анализе выполнения операций.

Если происходит прерывание выполнения какой-либо микропрограммы с использованием БА для удовлетворения причин, вызывающих прерывание, необходимо сохранить байт состояния, записав его в память машины, и вернуть перед началом продолжения прерванной микропрограммы. При выводе ББА в поле В микрокоманды задается код БА, по которому содержимое триггеров

ББА передается на РВ (рис. 6.7). Задавая в поле ФУНКЦИЯ этой микрокоманды операцию В ТРАНЗИТ, а в поле С — какой-либо регистр процессора, можно передавать содержимое триггеров ББА на любой регистр и в последующих микрокомандах записать его в память машины.

ТЭН	ТРКФ	ТРПФ	ТПКФ	ТППФ	ТПЕР	ТНДД	ТЧЕТ
0	1	2	3	4	5	6	7

Рис. 6.7. Побитовое распределение триггеров байта состояния БА при передаче их содержимого на вход РВ

При вводе ББА на РА заносится информация регистра процессора, содержащая предварительно прочитанный из памяти байт состояния. В поле ФУНКЦИЯ микрокоманды задается операция А ТРАНЗИТ, а в поле УСТАНОВ — код ИГН (игнорировать). При этом вырабатывается сигнал БА: = С, по которому информация с выхода арифметическо-логического блока передается в триггеры ББА.

Анализ состояния триггеров производится в начале такта работы БА, т. е. анализируется результат выполнения предыдущей операции, изменившей состояние рассматриваемого триггера. Ряд триггеров имеет принудительную установку в нулевое состояние (а некоторые — и в единичное состояние) сигналами поля УСТАНОВ микрокоманды.

Таблица 6.2

Код поля УСТАНОВ	Мнемоника	Сигнал дешифратора поля УСТАНОВ	Действие
00001	ГАНШ	БА: = 0	Установка в нулевое состояние триггеров ТНДД, ТРПФ, ТРКФ, ТППФ, ТПКФ, ТПЕР байта состояния БА
01010	ОППФ	ТППФ: = 0	Установка ТППФ в нулевое состояние или присвоение нулевого значения входящему переносу
01011	1ППФ	ТППФ: = 1	Установка ТППФ в единичное состояние или присвоение единичного значения входящему переносу
01100	ОПКФ	ТПКФ: = 0	Установка ТПКФ в нулевое состояние или присвоение нулевого значения входящему переносу
01101	1ПКФ	ТПКФ: = 1	Установка ТПКФ в единичное состояние или присвоение единичного значения входящему переносу
01110	ОРПФ	ТРПФ: = 0	Установка в нулевое состояние триггера результата прямой функции
01111	ОРКФ	ТРКФ: = 0	Установка в нулевое состояние триггера результата косвенной функции
10110	ИГН	ИГН	Игнорирование входящего и выходящего переносов

В таблице 6.2 даны значения поля УСТАНОВ и перечень соответствующих им установок триггеров или сигналов БАА. Эти установки изменяют состояние триггера в середине такта работы машины и, следовательно, не мешают анализу результата выполнения предыдущей операции. В выполняемой операции учитывается состояние триггера, задаваемое полем УСТАНОВ.

В состав байта состояния входят следующие триггеры.

Триггер знака (ТЗН) хранит информацию о состоянии старшего, нулевого бита байта результата. Изменяет свое состояние в конце такта работы БА, запомнив значение нулевого бита байта результата.

Триггер четности байта (ТЧЕТ) служит для хранения информации о состоянии младшего, седьмого бита байта результата. Изменяет свое состояние в конце такта работы БА, запомнив значение седьмого бита байта результата.

Триггер результата косвенной функции (ТРКФ) предназначен для хранения информации о результате выполнения операции, выполняемой по косвенной функции. Устанавливается в единичное состояние по результату любой операции, выполняемой с помощью косвенной функции, если результирующий байт не равен нулю. В случае, если байт результата равен нулю, состояние ТРКФ не изменяется. Имеется возможность установки ТРКФ в нулевое состояние заданием кода ОРКФ или ГАШ в поле УСТАНОВ микрокоманды.

Триггер результата прямой функции (ТРПФ) хранит информацию о результате выполнения операции, выполняемой по прямой функции. Устанавливается в единичное состояние при выполнении любой операции, выполняемой при помощи прямой функции, если результирующий байт не равен нулю. В случае, если байт результата равен нулю, состояние ТРПФ не изменяется. Имеется возможность установки ТРПФ в нулевое состояние заданием кодов ОРПФ или ГАШ поля УСТАНОВ микрокоманды.

Триггер переполнения (ТПЕР) служит для хранения информации о наличии или отсутствии переполнения восьмибитной сетки БА. Изменяет свое состояние в конце такта работы БА при выполнении операций сложения, вычитания, сдвига влево данных на один бит. Наличие переполнения характеризуется несогласованием переносов из первого и нулевого бит байта информации.

Триггер неверных десятичных данных (ТНДД) хранит информацию о правильности задания десятичных данных. При выполнении операций десятичной арифметики в случае нецифровой комбинации входных данных происходит установка триггера в единичное состояние.

Триггер переноса прямой функции (ТППФ) хранит информацию о наличии или отсутствии межбайтового переноса в операциях сложения (вычитания) или значение выдвигаемого бита в операциях сдвига, выполняемых по прямой функции.

При выполнении операции СЛОЖИТЬ в узле переносов учитывается значение ТППФ, задаваемое полем УСТАНОВ микро-

команды или установленное при выполнении предыдущей операции (сложения, вычитания, сдвига), и ему присваивается истинное значение полученного межбайтового переноса (при наличии переноса ТППФ — в единичном состоянии, при отсутствии — в нулевом).

При выполнении операции ВYЧЕСТЬ в блоке переносов учитывается инвертированное значение ТППФ, задаваемое полем УСТАНОВ или полученное при выполнении предыдущей операции (сложения, вычитания, сдвига). В конце такта работы БА триггеру ТППФ присваивается инвертированное значение полученного межбайтового переноса (при наличии переноса ТППФ — в нулевом состоянии, при отсутствии — в единичном).

Такой алгоритм позволяет произвести добавление единицы в операнд при организации дополнительного кода вычитаемого, так как нулевое значение входящего переноса при обработке младшего байта числа будет воспринято единицей. При обработке последующих байт значения межбайтовых переносов, дважды инвертированные, будут учитываться истинными.

При выполнении операции СДВИГ В ВПРАВО в старший (нулевой) бит сдвигаемого байта заносится значение триггера ТППФ, задаваемое полем УСТАНОВ или установленное при выполнении предыдущей операции (сложения, вычитания, сдвига). В конце такта работы БА в ТППФ передается значение младшего (седьмого) бита сдвигаемого байта.

При выполнении операции СДВИГ В ВЛЕВО в младший (седьмой) бит сдвигаемого байта заносится значение триггера ТППФ, задаваемое полем УСТАНОВ или установленное при выполнении предыдущей операции (сложения, вычитания, сдвига). В конце такта работы БА в ТППФ передается значение старшего (нулевого) бита сдвигаемого байта.

Если в поле УСТАНОВ микрокоманды задано значение ИГН, то ТППФ не изменяет своего состояния по результату выполнения операции. Возможна установка ТППФ в нулевое и единичное состояние заданием в поле УСТАНОВ микрокоманды кодов ГАШ, ОППФ, ИППФ. Триггер ТППФ устанавливается в состояние, заданное полем УСТАНОВ, только при выполнении операций, результат которых не изменяет состояния этого триггера.

Триггер переноса косвенной функции (ТПКФ) хранит информацию о наличии или отсутствии межбайтового переноса в операциях сложения (вычитания) и значение выдвигаемого бита в операциях сдвига, выполняемых с помощью косвенной функции. Триггер изменяет свое состояние в конце такта работы БА при выполнении операций сложения, вычитания, сдвигов влево и вправо с помощью косвенной функции, если в этих микрокомандах в поле УСТАНОВ микрокоманды не задано значение ИГН.

При выполнении этих операций ТПКФ используется так же, как и ТППФ. Установка ТПКФ в нулевое и единичное состояния заданием в поле УСТАНОВ микрокоманды кодов ГАШ, ОКФ и ИПКФ производится только при выполнении операций, результат которых не изменяет состояния этого триггера.

В состав ББА входят схемы формирования сигналов $\pm П(С)$ и $\pm П(7)$. Сигналы формируются следующим образом.

Межбайтовый перенос, полученный при выполнении какой-либо микрокоманды и хранящийся в триггерах переноса прямой или косвенной функции, используется при обработке последующих байт информации при выполнении операций сложения (вычитания) и сдвигов вправо и влево. Результат текущей операции может изменить состояние триггера ТППФ или ТПКФ до окончания ее выполнения, поэтому в начале каждой операции синхронимпульсом ТИ13 (при отсутствии в поле УСТАНОВ кода ИГН) значение ТППФ или ТПКФ переписывается в триггер результирующего переноса ТП. При наличии в поле УСТАНОВ микрокоманды кода ИГН производится установка ТП в нулевое состояние.

Триггер результирующего переноса участвует в формировании сигналов $+П(С)$, $-П(С)$ и $+П(7)$, $-П(7)$, используемых соответственно узлом сдвига вправо и узлом переноса при выполнении текущей операции в случае, если полем УСТАНОВ микрокоманды не задана установка триггера ТППФ или ТПКФ в нулевое или единичное состояние.

Если же в поле УСТАНОВ микрокоманды задается принудительная установка триггера ТППФ и операция выполняется по прямой функции или триггера ТПКФ и операция выполняется по косвенной функции, то формирование сигналов $\pm П(С)$ и $\pm П(7)$ производится в соответствии со значениями поля УСТАНОВ микрокоманды.

6.4. Выполнение операций в блоке

Все операции, выполняемые арифметическо-логическим блоком, по алгоритму работы можно разделить на три группы:

все логические операции и операции передачи данных с РА или РВ;

все арифметические операции;

операции сдвигов вправо или влево на один бит данных РВ.

К первой группе относятся операции: $A \vee B$, $A \& B$, $A \vee \overline{B}$, $A \oplus B$, $\overline{A} \& B$, А ТРАНЗИТ и В ТРАНЗИТ.

При выполнении этих операций обработку данных производит только логический узел. Данные в других узлах формируются таким образом, чтобы результат, полученный в логическом узле, не изменился. Для этого узел сдвига вправо формирует нулевую информацию, которая, пройдя через узел переносов, суммируется по модулю два с результатом, полученным в логическом блоке, не изменяя его.

Узлы десятичных корректоров (первого и второго) при выполнении этих операций пропускают данные, не изменяя их.

На рис. 6.8 показана схема обработки данных узлами БА при выполнении операций $A \vee B$, $A \vee \overline{B}$, А ТРАНЗИТ.

Ко второй группе относятся операции двоичного и десятичного

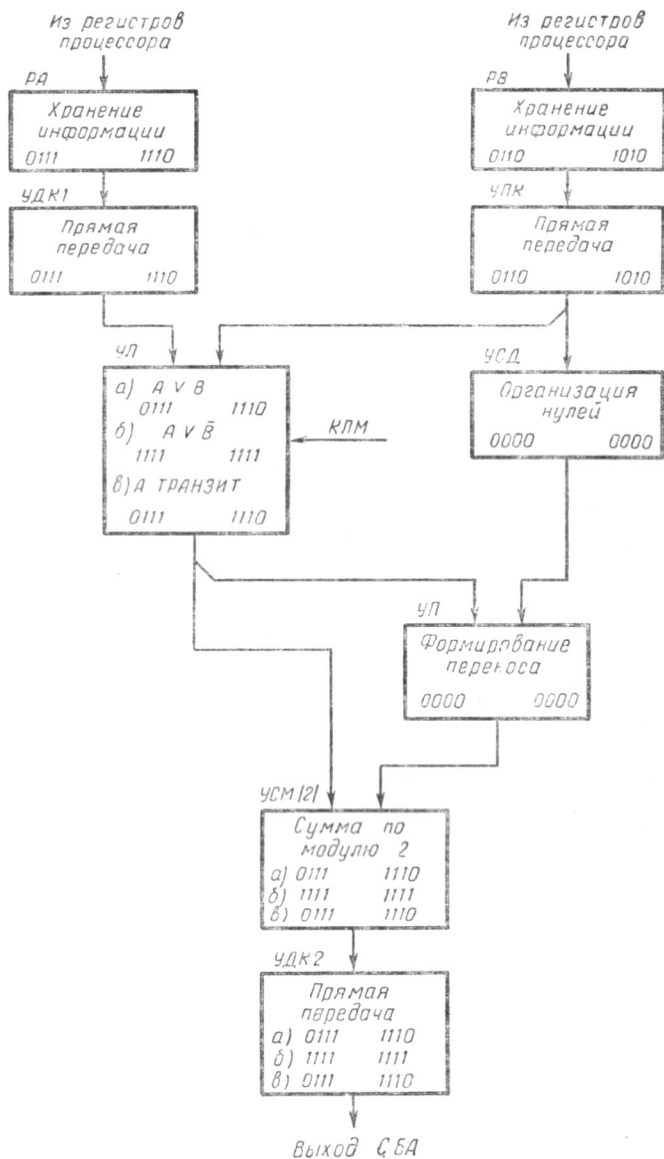


Рис. 6.8. Схема обработки данных узлами БА при выполнении операций $A \vee B$ (а), $A \vee \bar{B}$ (б), А ТРАНЗИТ (в)

сложения и вычитания: $(A+B)ДВ$, $(A-B)ДВ$, $(B-A)ДВ$, $(A+B)ДЕС$, $(A-B)ДЕС$, $(B-A)ДЕС$.

При выполнении операции $(A+B)ДВ$ в логическом узле производится суммирование данных с $РА$ и $РВ$ по модулю два (первое полусуммирование). Результат первого полусуммирования и операнд с $РВ$, прошедший без изменения через узел сдвига вправо, подаются в узел переносов, в котором образуются значения межбитовых и межбайтового переносов. Межбайтовый перенос поступает в схему хранения. Входной перенос учитывает ранее установленное состояние схемы хранения межбайтового переноса или текущее значение поля $УСТАНОВ$ микрокоманды.

Данные с выходов логического узла и узла переносов (первая полусумма и значения межбитовых переносов) подаются на вход узла суммирования по модулю два, где формируется вторая полусумма (окончательный результат).

Узлы десятичных корректоров при выполнении операции $(A+B)ДВ$ передают данные без изменений. На рис. 6.9 приведена схема обработки данных узлами $БА$ при выполнении операции $(A+B)ДВ$.

При выполнении операции $(A-B)ДВ$ в логическом узле производится операция ЭКВИВАЛЕНТ над данными $РА$ (уменьшаемое) и $РВ$ (вычитаемое), тождественная суммированию по модулю два уменьшаемого с инвертированным вычитаемым.

Результирующие данные с выхода логического узла и вычитаемое, инвертированное в узле сдвигов вправо, подаются в узел переносов, где образуются межбитовые и межбайтовый переносы. Межбайтовый перенос (инвертированное значение) поступает на схему хранения.

Входной перенос учитывает инвертированное значение ранее установленного состояния схемы хранения межбайтового переноса или текущего состояния поля $УСТАНОВ$ микрокоманды.

Такой алгоритм позволяет произвести добавление единицы в операнд при организации дополнительного кода вычитаемого, так как исходное нулевое значение входного переноса при обработке младшего байта числа будет воспринято как единичное. При обработке последующих байт значения межбайтовых переносов, дважды инвертированные, будут учитываться истинными.

Данные с выходов логического узла и узла переносов (межбитовые переносы) подаются на вход узла суммирования по модулю два, где организуется окончательный результат.

Узлы десятичных корректоров при выполнении операции $(A-B)ДВ$ передают данные без изменений. Схема обработки данных узлами $БА$ при выполнении операции $(A-B)ДВ$ также приведена на рис. 6.9.

При выполнении операции $(B-A)ДВ$ в логическом узле производится операция ЭКВИВАЛЕНТ над данными с $РА$ (вычитаемое) и $РВ$ (уменьшаемое), тождественная в рассматриваемом случае суммированию по модулю два данных $РВ$ (уменьшаемое) с инвертированными данными регистра A (вычитаемое).

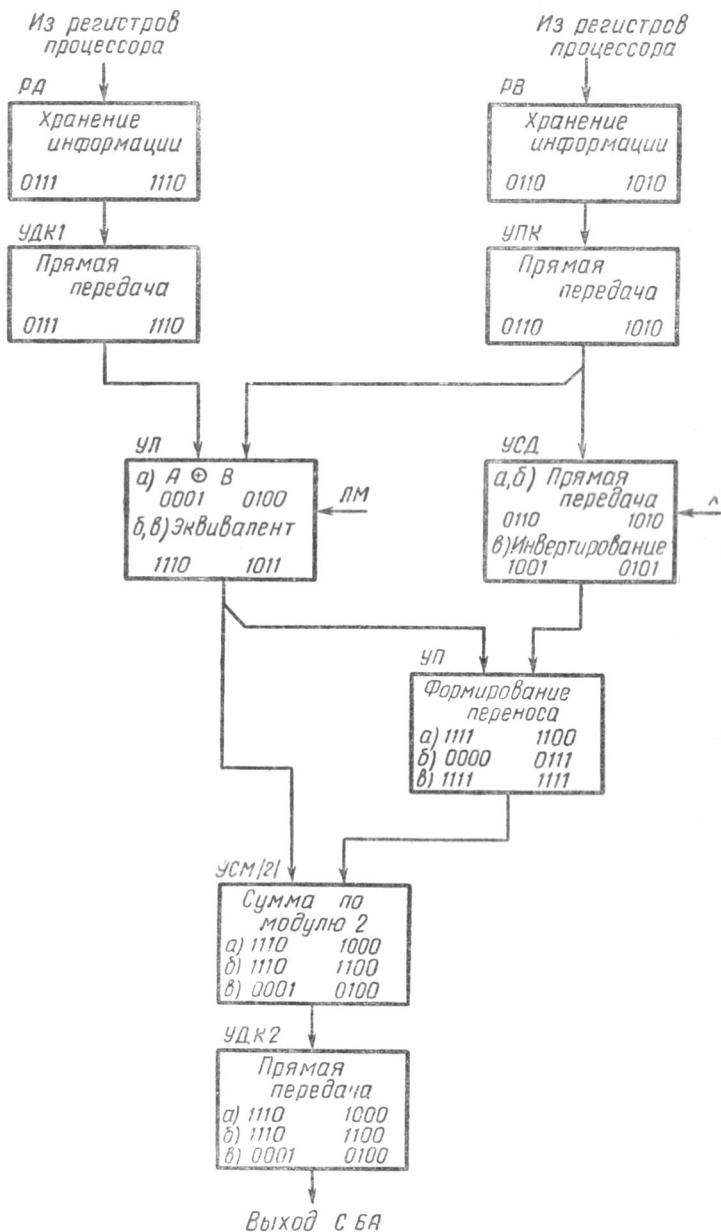


Рис. 6.9. Схема обработки данных узлами БА при выполнении операций $(A+B)ДВ$ (а), $(B-A)ДВ$ (б), $(A-B)ДВ$ (в)

Результирующие данные с выхода логического узла и уменьшаемое с РВ, прошедшее без изменения через узел сдвигов вправо, подаются в узел переносов, где образуются значения межбитовых и межбайтового переносов.

Межбайтовый перенос (инвертированное значение) поступает на схему хранения.

Входной перенос учитывает инвертированное значение ранее установленного состояния схемы хранения межбайтового переноса или текущего состояния поля УСТАНОВ микрокоманды.

Данные с выходов логического узла и узла переносов подаются в узел суммирования по модулю два, где формируется окончательный результат.

Узлы десятичных корректоров при выполнении операции $(B-A)_{ДВ}$ передают данные без изменений. Схема обработки данных узлами БА при выполнении операции $(B-A)_{ДВ}$ показана на рис. 6.9.

Выполнение операций сложения и вычитания десятичных чисел $(A+B)_{ДЕС}$, $(A-B)_{ДЕС}$, $(B-A)_{ДЕС}$, представленных в коде 8421, отличается от выполнения аналогичных операций над двоичными числами только работой узлов десятичной коррекции.

При выполнении операций $(A+B)_{ДЕС}$ в первом десятичном корректоре в каждую тетраду данных РА добавляется цифра шесть (0110). Во втором десятичном корректоре производится анализ истинности результата, полученного с выхода узла суммирования по модулю два. В зависимости от результата анализа второй десятичный корректор либо передает данные из узла суммирования по модулю два без изменения, либо производит вычитание шести из тетрад этих данных.

При выполнении операций $(A-B)_{ДЕС}$ и $(B-A)_{ДЕС}$ первый десятичный корректор передает данные без изменений. Второй десятичный корректор производит анализ правильности результата, полученного с выхода узла суммирования по модулю два. В зависимости от результата анализа данные через узел десятичного корректора проходят без изменения или же из тетрад данных вычитается цифра шесть.

На рис. 6.10 показана схема обработки данных узлами БА при выполнении операций $(A+B)_{ДЕС}$, $(A-B)_{ДЕС}$, $(B-A)_{ДЕС}$.

Операция сдвига влево на один бит данных регистра В (СДВИГ В ВЛЕВО) производится в узле переносов. При этом используется алгоритм образования межбитовых переносов, заключающийся в том, что если данный бит первой полусуммы имеет нулевое значение, а одно из слагаемых — единичное или нулевое значение, то перенос в последующий бит имеет единичное или нулевое значение соответственно.

При выполнении операции сдвига влево на один бит сдвигаемые данные подаются на РВ.

Независимо от информации, находящейся в РА и РВ, логический блок вырабатывает нулевую информацию, которая подается на вход узла переносов (имитируется первая полусумма). Данные

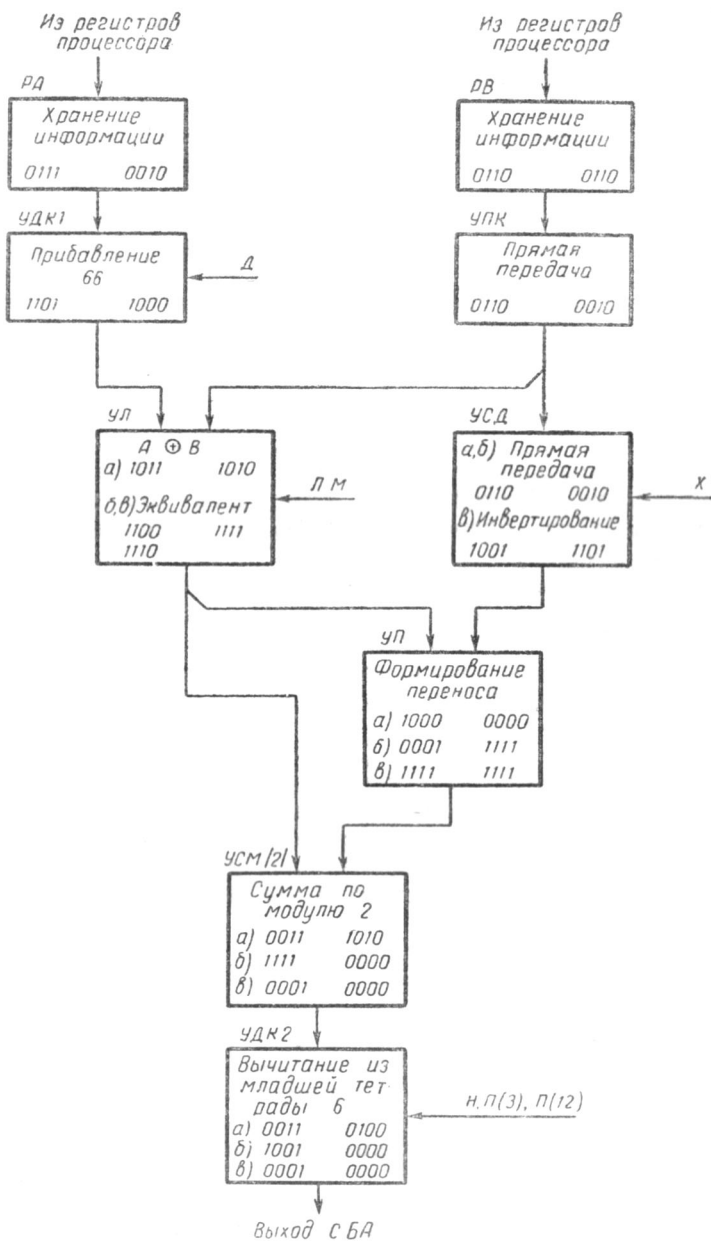


Рис. 6.10. Схема обработки данных узлами БА при выполнении операций (А+В) ДЕС (а), (В-А) ДЕС (б), (А-В) ДЕС (в)

с RB, пройдя без изменения через узел сдвига вправо, также подаются на вход узла переносов (имитируется одно из слагаемых). На выходе узла переносов получаются значения межбитовых переносов, соответствующие сдвинутым влево на один бит данным регистра В.

Крайний левый бит сдвигаемых данных определяет при этом значение межбайтового переноса, который подается на схему хранения.

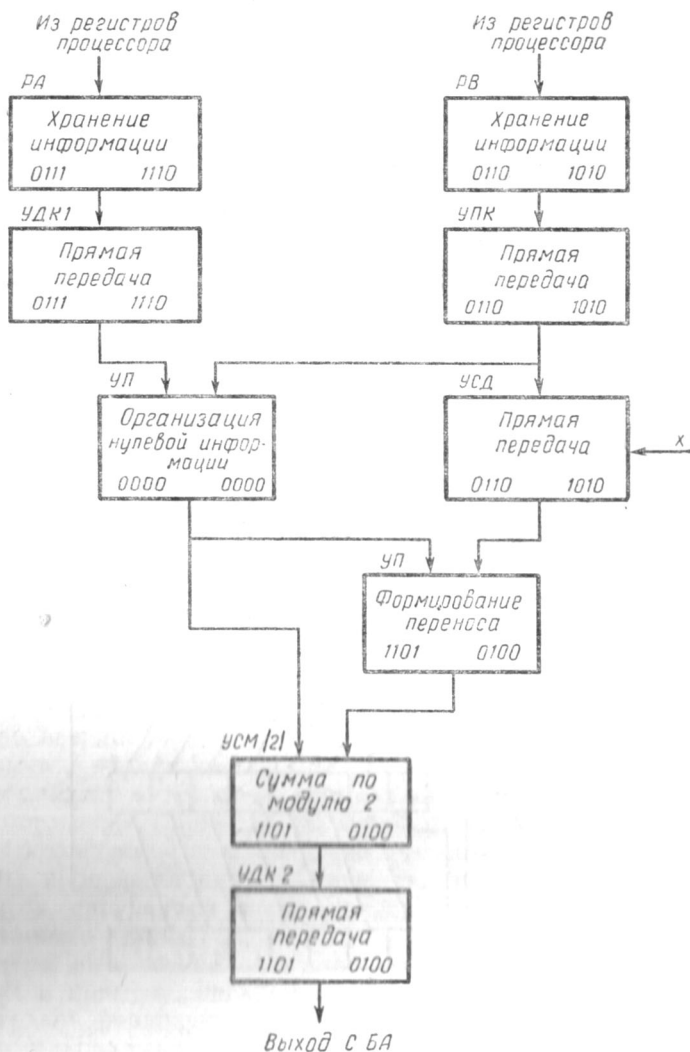


Рис. 6.11. Схема обработки данных узлами БА при выполнении операции СДВИГ В ВЛЕВО на один бит

Нулевая информация выхода логического узла, суммируясь по модулю два с выходными данными узла переносов, не изменяет его результата.

Узлы десятичных корректоров при выполнении этой операции пропускают данные без изменения.

На рис. 6.11 показана схема обработки данных узлами БА при выполнении операции сдвига влево на один бит.

Подготовка к выполнению операции сдвига вправо на один бит содержимого РВ производится в узле сдвигов вправо, завершение операции происходит в узле переносов при сдвиге информации влево на один бит. Крайний правый бит байта сдвигаемых данных запоминается в схеме хранения межбайтовых переносов. Операция выполняется следующим образом. Сдвигаемые данные подаются на РВ. В узле сдвига вправо производится сдвиг 0—5-го бит информации вправо на два бита. В 1-й бит УСД заносится информация, учитывающая ранее установленное значение межбайтового переноса или текущее значение поля УСТАНОВ микрокоманды. В 0-й бит УСД заносится значение 7-го бита данных.

В логическом узле, независимо от данных, находящихся в РА и РВ, вырабатывается нулевая информация.

Данные с выходов узла сдвига вправо и логического узла подаются на вход узла переносов, в котором осуществляется сдвиг данных УСД влево на один бит, причем вместо входящего переноса в младший бит узла подается 6-й бит сдвигаемых данных. Таким образом, информация РВ оказывается сдвинутой вправо на один бит, а 7-й бит данных, имитируя межбайтовый перенос, поступает на схему хранения (рис. 6.12). Далее обработка данных ведется так же, как при выполнении сдвига влево.

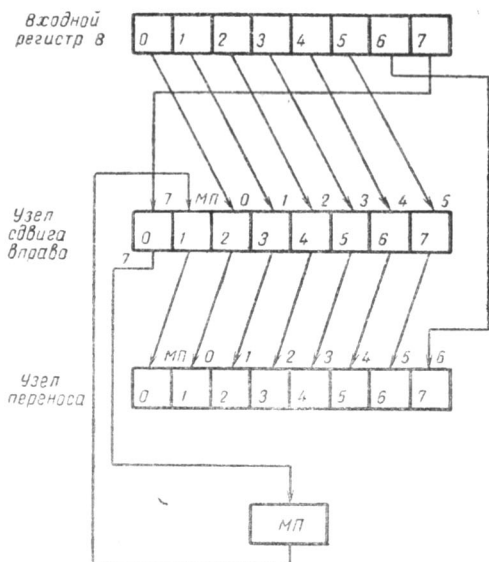


Рис.6.12. Схема передачи содержимого РВ при сдвиге вправо на один бит: МП—схема хранения межбайтового переноса

6.5. Обработка чисел в блоке

Алгоритм выполнения сложения и вычитания чисел зависит от того, в каком коде обрабатываемое число представлено в машине. Наиболее просто сложение и вычитание выполняются над числами, представленными в двоичной системе счисления в дополнительных кодах.

В этом случае при выполнении сложения необходимо произвести суммирование чисел, обращаясь со знаковым битом так же, как с остальными битами, а при выполнении вычитания — суммирование уменьшаемого с инвертированным значением кода числа вычитаемого, подав единицу на вход переноса младшего бита числа (суммирование уменьшаемого с дополнением кода вычитаемого).

Если обрабатываемые двоичные числа представлены в машине в прямых кодах, то алгоритм выполнения сложения и вычитания несколько усложняется. При этом прежде всего необходимо произвести анализ знаков числа. Если знаки одинаковые, то выполняется заданное действие (сложение, вычитание), если же знаки разные, то действие, противоположное заданному (при заданном сложении — вычитание, и наоборот). При выполнении сложения необходимо произвести суммирование основных бит чисел, а знаковому биту результата присвоить значение знакового бита первого слагаемого. При выполнении вычитания необходимо суммировать большее слагаемое с дополнительным кодом меньшего слагаемого, присвоив результату знак большего слагаемого.

В ЭВМ ЕС-1020 обрабатываются числа с фиксированной запятой, представленные в дополнительном коде, числа с плавающей запятой, представленные в прямом коде, и десятичные числа, представленные в прямом коде 8421.

Из изложенного выше следует, что алгоритм работы арифметическо-логического блока (БА) при выполнении операций $(A+B)_{ДВ}$ и $(A-B)_{ДВ}$ соответствует алгоритму обработки двоичных чисел, представленных в дополнительных кодах. Следовательно, при выполнении сложения (вычитания) чисел с фиксированной запятой правильный результат получается при последовательной обработке в БА операциями $(A+B)_{ДВ}$, $(A-B)_{ДВ}$.

При выполнении сложения (вычитания) чисел с плавающей запятой необходимо (дополнительно к операциям $(A+B)_{ДВ}$, $(A-B)_{ДВ}$) произвести ряд действий, связанных с определением знаков чисел, сравнением чисел, присвоением результирующему числу истинного знака.

Алгоритмы выполнения сложения (вычитания) чисел, представленных в прямых кодах, разрабатываются при написании микропрограмм, реализующих эти операции. Наличие в микропрограмме микрокоманд, содержащих дополнительные операции, усложняет микропрограммы и приводит к потере быстродействия.

При выполнении сложения (вычитания) десятичных чисел также необходимо дополнительно к операциям $(A+B)_{ДЕС}$, $(A-B)_{ДЕС}$,

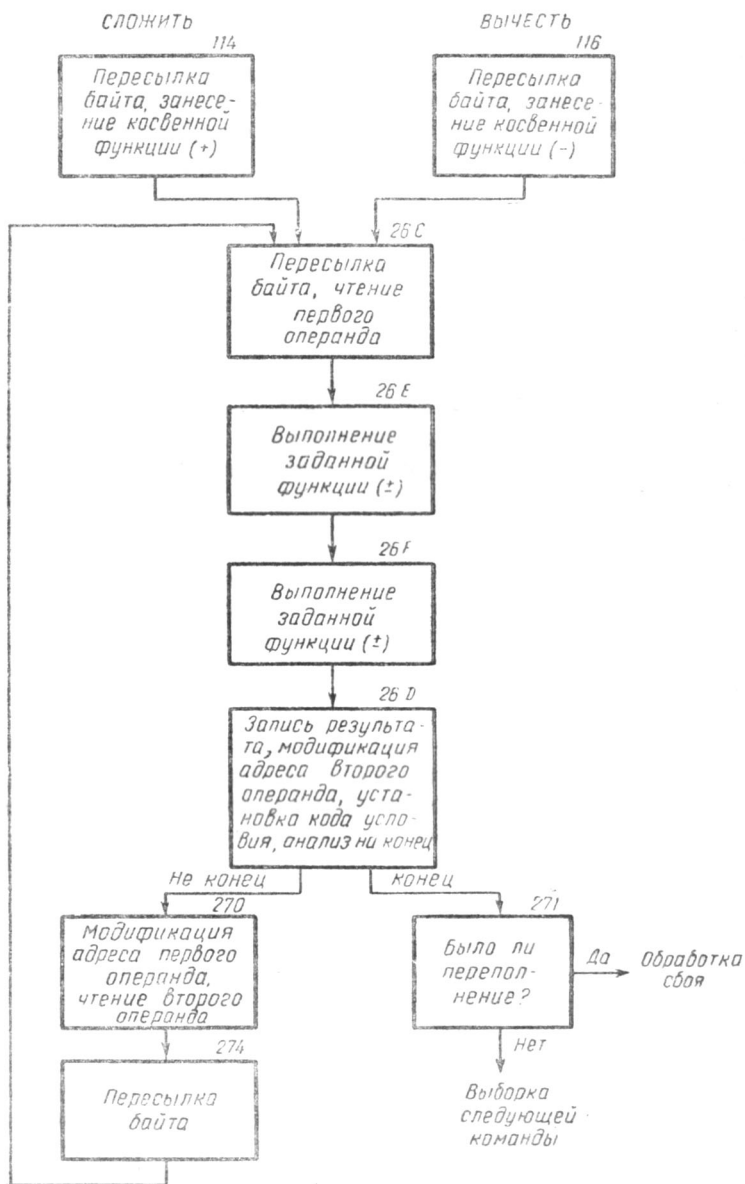


Рис. 6.13. Алгоритм выполнения команд СЛОЖИТЬ и ВЫЧЕСТЬ

(В—А)ДЕС произвести ряд операций по определению знаков чисел, сравнению чисел, присвоению знака результату.

На рис. 6.13 и табл. 6.3 приведен алгоритм и микропрограмма выполнения команд СЛОЖИТЬ и ВЫЧЕСТЬ формата RR для чисел с фиксированной запятой. Исходными операндами в этих командах являются тридцатидвухбитовые числа, расположенные в локальной памяти (общих регистрах). Результат сложения (вычитания) записывается по адресу первого операнда. Результатом выполнения команды является также установка кода условия.

На рисунке и в таблице отсутствует процедура выборки команды, в результате которой организуется адреса младших байт операндов (в регистрах И и У) и выход на микропрограмму реализации выбранной команды.

Таблица 6.3

СИМП ЕС-1020

ЛИСТ 01

ИДЕНТИФИКАТОР СЛВЧ

СЛОЖИТЬ, ВЫЧЕСТЬ

АДР	ФУНКЦИЯ	ПАМЯТЬ	УСТАН	УСЛ1	УСЛО	КОНСТ	ПЕРЕХОД
114	Д=Н		ЗКФ			(+)	26С
116	Д=Н		ЗКФ			(-)	27Е
26С	Л=З	У ЧТЛ					26Е
27Е	Л=З	У ЧТЛ					26Е
26Е	З=З*Л						26F
26F	Н=Н*Д						26D
26D	И=ВНС	ЗПЛ	КУ1		1БС4		270
270	У=ВУС	И ЧТЛ			1ПЕР		274
271	=		1БС4				274:
275	=						028 ППФЗ
274	Д=Н КОНЕЦ СЛВЧ	РГ					26С

Для рассматриваемых команд в процессе выборки команды производится чтение младшей пары байт второго операнда из локальной памяти в информационные регистры Н и З.

Микропрограмма реализации этих команд начинается с микрокоманд с адресами 114 и 116 (соответственно для команд СЛОЖИТЬ и ВЫЧЕСТЬ), в которых производится пересылка одного из байтов второго операнда из информационного регистра РН в регистр процессора РД. Пересылка производится через блок БА, в котором в это время выполняется операция А ТРАНЗИТ. Одновременно в регистр косвенной функции (РКФ) заносится код операции (А+В)ДВ или (А-В)ДВ для команды ВЫЧЕСТЬ.

Далее для команд СЛОЖИТЬ и ВЫЧЕСТЬ выполняется микрокоманда с адресом 26С, в которой аналогично описанному

производится пересылка байта второго операнда из РЗ в РЛ и чтение в освободившиеся информационные регистры памяти РН и РЗ двух младших байт первого операнда.

В микрокомандах с адресами 26E и 26F выполняется операция сложения или вычитания, заданная в РКФ над двумя младшими байтами операндов с занесением результата в РН и РЗ. Выполнение косвенной функции на языке ЯМП имеет мнемонику *.

В микрокоманде с адресом 26D производится запись результата в память (по адресу первого операнда, который хранится в адресном регистре МН со времени чтения байт первого операнда). В этой же микрокоманде организуется адрес старшей пары байт второго операнда путем выполнения операции В ТРАНЗИТ с передачей на вход В в режиме СТАРШИЕ ПРЯМО, а также аппаратурная установка кода условия и анализ на окончание выполнения обработки операндов (анализ на единичное состояние триггера БС4 в поле УСЛО). В случае, если обработка не закончена (триггер БС4 в нулевом состоянии), выполняется микрокоманда с адресом 270, в которой аналогично описанному производятся организация адреса старшей пары байт первого операнда, чтение старшей пары байт второго операнда и установка в единичное состояние триггера БС4.

В микрокоманде с адресом 274 производится регенерация памяти и пересылка байта второго операнда в регистр РД. Далее вновь производится выполнение микрокоманд с адресами 26С, 26E, 26F и 26D. Переход с последней микрокоманды идет по адресу 271, так как триггер БС4 уже установлен в единичное состояние.

В микрокоманде с адресом 271 производится анализ состояния триггера переполнения, который устанавливается в блоке БА при выполнении арифметических операций.

Если переполнение не было зафиксировано, происходит переход на микропрограмму выборки следующей команды, если переполнение было зафиксировано — на микропрограмму обработки сбойной ситуации.

Глава 7

СИСТЕМА ПЕРЫВАНИЙ

7.1. Понятие о супервизоре и структура слова состояния программы

Программы, с помощью которых решаются различные прикладные задачи, называются проблемными. Для их управления с целью сокращения времени простоя машины, обеспечения непрерывной работы, уменьшения степени вмешательства в ее работу в память помещают управляющие программы.

Простейший тип управляющей программы используется для управления загрузкой проблемных программ. Действия при этом выполняются в следующей последовательности:

1) подготавливается для ввода лента, содержащая проблемные программы и связанные с ними данные;

2) оператор загружает управляющую программу в основную память;

3) управляющая программа осуществляет загрузку проблемной программы и передает ей управление;

4) проблемная программа считывает данные и выполняет задачу;

5) после завершения выполнения проблемная программа не выдает инструкцию останова машины, а возвращает управление управляющей программе;

6) управляющая программа загружает затем следующую проблемную программу и передает ей управление;

7) эта операция продолжается до тех пор, пока все проблемные программы не будут выполнены.

Однако управляющая программа может выполнять и другие, более сложные функции, одна из которых состоит в запуске (иницировании) операций ввода-вывода. В этом случае проблемная программа передает управление управляющей программе каждый раз при необходимости выполнения операции ввода-вывода. При этом операции считывания и записи, необходимые для передачи данных между основной памятью и устройствами ввода-

вывода, могут осуществляться с помощью управляющей программы. Операция ввода-вывода, выполняемая с помощью управляющей программы, может состоять из ряда инструкций, которые кроме запуска устройства ввода-вывода, включают проверку ошибочных состояний, общего состояния устройства ввода-вывода и т. д.

Такое расширение функций управляющей программы вызывает соответствующее увеличение занимаемого ею объема основной памяти. С этой целью те секции управляющей программы которые используются не слишком часто, размещают на внешних быстродействующих устройствах ввода-вывода, таких, как дисковое запоминающее устройство. В основной памяти содержатся только те секции, которые необходимы для управления выполнением проблемных программ. Часть управляющей программы, постоянно находящаяся в основной памяти, называется супервизорной программой, или *супервизором*. По мере надобности супервизор вызывает остальные секции управляющей программы с дисков в основную память. Как было указано, выполнение любой программы производится под управлением ССП.

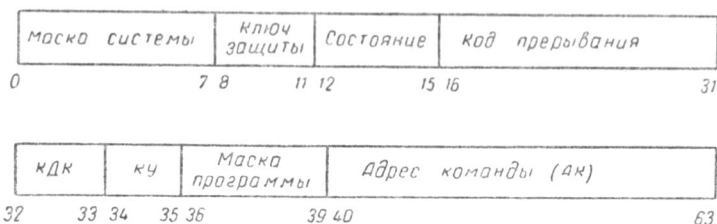


Рис. 7.1. Структура ССП

На рис. 7.1 показана структура слова состояния программы (ССП). Биты 0—7 ССП содержат маску системы: маску мультиплексного канала (бит 0), маски первого и второго селекторных каналов (биты 1 и 2 соответственно), не используются биты 3—6, маску внешних прерываний (бит 7). Биты 8—11 хранят ключ защиты памяти. Биты 12—15 определяют: код (КОИ-8 или ДКОИ) работы системы (бит 12)*, маску прерываний от схем контроля машины (бит 13), состояние «Ожидание» (бит 14 равен единице), состояние «Задача» — выполнение проблемной программы (бит 15 равен единице). Биты 16—31 хранят код прерывания. Биты 32, 33 содержат код длины выполняемой команды, состояние этих бит указывает количество полуслов в команде в двоичной системе счисления (01, 10, 11). Биты 34, 35 предназначены для кода при-

* КОИ-8 и ДКОИ различаются способом представления знака и зоны в изображении десятичных чисел: зона в ДКОИ—1111, в КОИ-8—0101; знак «—» в ДКОИ—1101, в КОИ-8—1011; знак «+» в ДКОИ—1100, в КОИ-8—1010. Единичное состояние бита 12 соответствует КОИ-8

нака результата, который вырабатывается при выполнении ряда команд и может быть использован командами условных переходов. Биты 36—39 содержат маску программы: маску переполнения с фиксированной запятой (бит 36), маску десятичного переполнения (бит 37), маску исчезновения порядка (бит 38), маску потери значимости (бит 39). Биты 40—63 предназначены для адреса команды.

Адрес команды текущего ССП используется для выборки соответствующей команды. Как только команда будет выбрана, содержащийся в ССП адрес команды сразу же изменится на адрес следующей команды. Прерывания же могут происходить только после того, как выполнение команды будет закончено. Поэтому в поле адреса команды старого ССП будет содержаться не адрес последней выполненной команды, а адрес следующей, которая выполнялась бы, если бы прерывание не произошло.

Для того чтобы получить адрес последней выполненной команды, содержимое поля адреса команд старого ССП должно быть уменьшено на значение длины этой команды с помощью супервизора.

7.2. Действия при прерывании и возврат к прерванной программе

После завершения выполнения проблемной программы управление работой системы должно быть передано супервизору.

Машинный сбой (например, схема контроля обнаружила, что обрабатываемый байт информации имеет четное количество единиц) или программные ошибки (например, указан нечетный адрес операнда длиной в полуслово в инструкции или переполнение при выполнении команды сложения и т. п.) также вызывают автоматический переход к супервизору, а не останов машины.

Такой автоматический переход к супервизору называется *прерыванием*. Текущая последовательность команд при этом прерывается и происходит переход к новой последовательности команд. Все внешние и внутренние условия, вызывающие автоматическое обращение к супервизору, называются причинами прерываний, каждая из которых может быть отнесена к определенному классу прерывания:

1. Прерывание по контролю машины (контроль машины)

Возникает при сбое машины по сигналам от схем контроля

2. Прерывание по ошибкам программы (программное)

Вызывается одной из 15 программных ошибок

3. Прерывание при обращении к супервизору (вызов супервизора)

Происходит в результате выполнения команды ВЫЗОВ СУПЕРВИЗОРА

4. Прерывание по сигналам от внешних источников (внешние)

Могут быть вызваны нажатием кнопки «Прерывание» на пульте оператора, сигналом от таймера и другими причинами

5. Прерывание от ввода-вывода (вводо-выводные)

Возникает после окончания операций ввода-вывода

Каждому классу прерываний соответствуют в основной памяти фиксированные ячейки длиной в двойное слово для размещения в них так называемых старого и нового ССП (табл. 7.1), а также секции в супервизоре, которые обрабатывают прерывания различным образом.

Таблица 7.1

Адрес			Содержимое ячеек
десятичный	шестнадцатеричный	двоичный	
24	18	0001 1000	Старое ССП для внешних прерываний Старое ССП для обращения к супервизору
32	20	0010 0000	
40	28	0010 1000	Старое ССП для программных прерываний Старое ССП для прерываний по контролю машины
48	30	0011 0000	
56	38	0011 1000	Старое ССП для прерываний по вводу-выводу Новое ССП для внешних прерываний
88	58	0101 1000	
96	60	0110 0000	Новое ССП для обращения к супервизору Новое ССП для программных прерываний
104	68	0110 1000	
112	70	0111 0000	Новое ССП для прерываний по контролю машины Новое ССП для прерываний по вводу-выводу
120	78	0111 1000	

Прерывание весьма похоже на ветвление, однако его функции гораздо сложнее, чем у простой команды перехода, которая изменяет лишь адрес инструкции в текущем ССП. При прерывании происходит замена текущего ССП целиком, осуществляемая путем запоминания его в качестве старого и выборки нового. Управление программой переходит при этом к новому ССП, которое становится текущим.

Таким образом при прерывании выполняются следующие действия: текущее ССП, управляющее программой до прерывания, автоматически запоминается в основной памяти и становится старым, а новое ССП автоматически выбирается из основной памяти на регистры процессора и становится текущим. Новая последовательность команд выполняется под управлением этого ССП.

Как видно из табл. 7.1, при сбое машины (прерывание по контролю машины) текущее ССП должно быть помещено в основную память по адресу 30*, а новое ССП выбрано по адресу 70. Если прерывание вызвано программной ошибкой, ССП, которое управляло программой до прерывания, автоматически записывается в ячейку 28. Затем новое ССП выбирается из памяти. Это ССП из ячейки 68 передает управление в ту секцию супервизорной про-

* В этой главе адресация дана в шестнадцатеричной системе счисления.

граммы, которая обрабатывает программные ошибки. Программа супервизора, обрабатывающая программные требования, будет анализировать информацию в ячейке 28, т. е. старое ССП.

Если в программе встречается команда ВЫЗОВ СУПЕРВИЗОРА, то в результате ее выполнения текущее ССП помещается в ячейку 20. Двойное слово выбирается из основной памяти по адресу 60 и загружается в регистры процессора. Это новое ССП, став текущим, передает управление той части супервизора, которая обрабатывает прерывания при вызове супервизора. Прерывание по этой команде используется проблемной программой для передачи управления супервизору.

Существует несколько случаев, когда проблемная программа должна вызвать супервизор. Основные из них:

необходимо сообщить супервизору о том, что проблемная программа выполнена; после этого супервизор может считать новую проблемную программу и загрузить ее ССП;

необходимо вызвать супервизор для того, чтобы запустить операцию ввода-вывода, требуемую для проблемной программы.

При нажатии кнопки «Прерывание» на пульте оператора, переполнении таймера* или появлении сигнала на внешней линии происходит внешнее прерывание. В этом случае текущее ССП автоматически запоминается в ячейке 18, т. е. становится старым. Двойное слово из ячейки 58, называемое новым ССП, выбирается из памяти и становится текущим.

Прерывание может быть также вызвано окончанием операции ввода-вывода. Прерывание от ввода-вывода вызывает запоминание ССП в ячейке 38, где оно хранится в качестве старого ССП. Затем новое ССП выбирается из памяти по адресу 78 и становится текущим. Это ССП передает управление в секцию супервизорной программы, обрабатывающей прерывания от ввода-вывода.

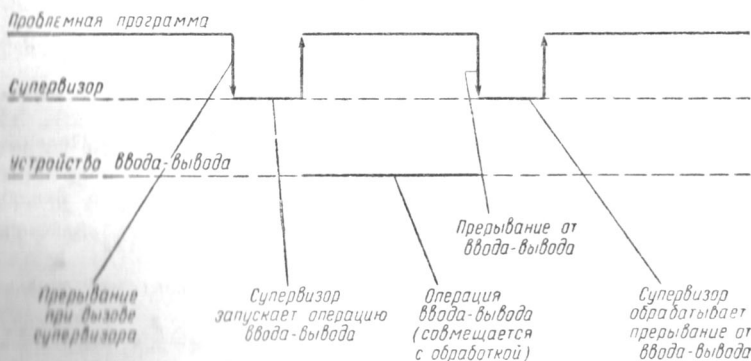


Рис. 7.2. Запуск операций ввода-вывода и прерывания от ввода-вывода

* Таймер служит для измерения относительного времени и представляет собой полноразрядное слово, хранящееся в специально отведенной для него ячейке основной оперативной памяти.

Причина прерывания	Код прерывания (биты 16—31)	Бит маски	Код длины команды	Выполнение команды
Ввод-вывод	(адреса ССП: старого — 56, нового — 120, приоритет 4)			
Каналы				
0	00000000 аааааааа	0	х	Завершается
1	00000001 аааааааа	1	х	Завершается
2	00000010 аааааааа	2	х	Завершается
3	00000011 аааааааа	3	х	Завершается
4	00000100 аааааааа	4	х	Завершается
5	00000101 аааааааа	5	х	Завершается
6	00000110 аааааааа	6	х	Завершается
Программа	(адреса ССП: старого — 40, нового — 104, приоритет 2)			
Код операции	00000000 00000001		1, 2, 3	Подавляется
Привилегированная операция	00000000 00000010		1, 2	Подавляется
Команда ИСПОЛНИТЬ	00000000 00000011		2	Подавляется
Защита	00000000 00000100		0, 2, 3	Подавляется или прекращается
Адресация	00000000 00000101		0, 1, 2, 3	Подавляется или прекращается
Спецификация	00000000 00000110		1, 2, 3	Подавляется
Десятичные данные	00000000 00000111		2, 3	Прекращается
Переполнение с фиксированной запятой	00000000 00001000	36	1, 2	Завершается
Деление с фиксированной запятой	00000000 00001001		1, 2	Подавляется или завершается
Десятичное переполнение	00000000 00001010	37	3	Завершается
Десятичное деление	00000000 00001011		3	Подавляется
Переполнение порядка	00000000 00001100		1, 2	Прекращается
Исчезновение порядка	00000000 00001101	38	1, 2	Завершается
Значимость	00000000 00001110	39	1, 2	Завершается
Деление с плавающей запятой	00000000 00001111		1, 2	Подавляется
Вызов супервизора	(адреса ССП: старого — 32, нового — 96, приоритет 2)			
Код операции в команде	00000000 чччччччч		1	Завершается
Внешний источник	(адреса ССП: старого — 24, нового — 88, приоритет 3)			
Таймер	00000000 1ппппппп	7	х	Завершается
Кнопка «Прерывание»	00000000 п1пппппп	7	х	Завершается
Внешние сигналы:				
2	00000000 пп1пппппп	7	х	Завершается
3	00000000 ппп1пппп	7	х	Завершается
4	00000000 пппп1ппп	7	х	Завершается
5	00000000 ппппп1пп	7	х	Завершается
6	00000000 пппппп1п	7	х	Завершается
7	00000000 ппппппп1	7	х	Завершается

Причина прерывания	Код прерывания (биты 16—31)	Бит маски	Код длины команды	Выполнение команды
Контроль машины (адреса ССП: старого — 48, нового — 112, приоритет 1)				
Сбой машины	00000000 00000000	13	x	Прекращается

Примечание. Условно обозначены: а — биты адреса устройства ввода-вывода, п — другие условия внешнего прерывания, ч — биты полей R1 и R2 команды ОБРАЩЕНИЕ К СУПЕРВИЗОРУ, х — значение этих бит непредсказуемо.

Прерывания от ввода-вывода происходят обычно по окончании операции ввода-вывода. Прерывание этого типа является эффективным способом сигнализации супервизору о том, что операция ввода-вывода закончена (рис. 7.2).

После того, как закончено выполнение программы обработки прерываний (вводо-выводных, внешнего и др.), целесообразно вернуться к выполнению прерванной проблемной программы. Это может быть сделано с помощью команды ЗАГРУЗИТЬ ССП, используемой супервизором для загрузки старого ССП обратно в регистр процессора, т. е. ССП становится снова текущим. В программах супервизора, обрабатывающих прерывания, эта команда обязательно должна быть последней.

Таким образом, обработка прерывания происходит следующим образом:

в момент прерывания текущее ССП, управляющее проблемной программой, запоминается в основной памяти в ячейке старого ССП, соответствующей классу, к которому относится причина прерывания. Старое ССП сообщает причину прерывания. В его поле адреса команды содержится адрес той команды, на которой было прервано выполнение проблемной программы;

новое ССП выбирается затем из памяти и становится текущим ССП. Это новое ССП (биты 40—63) указывает адрес первой команды программы обработки прерывания, которая является частью супервизора;

после окончания обработки прерывания последней командой программы супервизора, обрабатывающей прерывание, является команда ЗАГРУЗИТЬ ССП, старое ССП вновь становится текущим и происходит возврат к проблемной (прерванной) программе.

Причина прерывания может возникнуть в любой момент выполнения программы. Прерывание же может произойти после того, как выполнение текущей команды окончилось, а выполнение следующей не началось. Однако при программном прерывании или прерывании по контролю машины окончание команды может быть ускорено подавлением выполнения ее, если программная ошибка или сбой обнаружены во время выборки команды, или

прекращением выполнения команды, если ошибка или сбой обнаружены в процессе ее выполнения.

Если команда завершена, результаты заносятся в память, а код условия устанавливается так же, как и при обычном выполнении команды. Если выполнение команды прекращено, в память может быть занесен весь результат, часть его или ничего. Если команда подавляется, результаты в память не заносятся и код условия остается без изменения, что равносильно пропуску команды.

В табл. 7.2 указаны классы прерываний, причины, соответствующие каждому из этих классов, биты текущего ССП, указывающие программное запрещение отдельных причин (маски прерываний), и возможный код длины команды при различных вариантах окончания выполнения команды при прерываниях.

7.3. Приоритет прерываний

Во время выполнения команды могут возникнуть условия для возникновения нескольких прерываний. Программные прерывания и прерывания при обращении к супервизору взаимно исключают друг друга и поэтому вместе возникать не могут. Следовательно, одновременно могут появиться запросы на четыре типа прерываний: по контролю машины, программные или вызов супервизора, внешние, вводо-выводные.

Самый высокий приоритет имеет прерывание по контролю машины. Когда такое прерывание происходит, выполнение текущей команды прекращается. Программные прерывания или прерывание при обращении к супервизору, которое должно было бы произойти в результате выполнения этой текущей команды, игнорируется. Сбой машины обычно не влияет на вводо-выводные или внешние прерывания, продвижение таймера и передачу данных при вводе-выводе информации по селекторному каналу. При отсутствии машинного сбоя одновременно существующие запросы на прерывание воспринимаются в следующем порядке: программные или вызов супервизора, внешние, вводо-выводные.

При наличии запросов на программные (или вызов супервизора), внешние и вводо-выводные прерывания вначале текущее ССП запоминается в ячейке 28 основной памяти и новое ССП загружается на место текущего из ячейки 68. В конце этой микропрограммы загрузки ССП проверяются условия для внешних и вводо-выводных прерываний. При их наличии программа обработки программных ошибок не выполняется, а текущее ССП (им стало новое ССП программных прерываний) запоминается в ячейке 18 основной памяти. Новое ССП извлекается из ячейки 58 и становится текущим. В конце загрузки ССП обнаруживается запрос на вводо-выводное прерывание, поэтому программа обработки внешних прерываний также не исполняется. Снова текущее ССП (теперь им стало новое ССП внешних прерываний) запоминается в ячейке 38

основной памяти, а новое ССП загружается из ячейки 78 на место текущего. Так как текущим ССП теперь будет новое ССП прерывания от ввода-вывода, машина выполняет программу обработки прерывания от ввода-вывода, в конце которой осуществляется загрузка ССП из ячейки старого ССП для этого прерывания (ячейка 38 основной памяти). В результате текущим станет новое ССП внешних прерываний и следующей исполнится программа обработки внешних прерываний. В конце этой программы выполняется загрузка ССП из ячейки старого ССП внешних прерываний (18-я ячейка основной памяти). Так как в старом ССП внешних прерываний хранилось новое ССП программных прерываний, вводится программа обработки программных ошибок. В конце этой программы ССП загружается из ячейки 28 основной памяти, т. е. текущим станет ССП, которое управляло до возникновения трех прерываний, и, следовательно, процессор продолжит прерванную программу.

Таким образом, порядок обработки прерываний (выполнение прерывающих программ) обратен тому, в котором прерывания воспринимаются процессором, и соответствует их важности: вводо-выводные, внешние, программные (вызов супервизора).

Это относится к прерываниям, возникшим одновременно, при условии, что маски нового ССП, выбранные для каждого случая, разрешают другие типы прерываний.

Программист может создавать любой желаемый приоритет в исполнении программ прерываний установкой бит маски в ССП в соответствующее состояние.

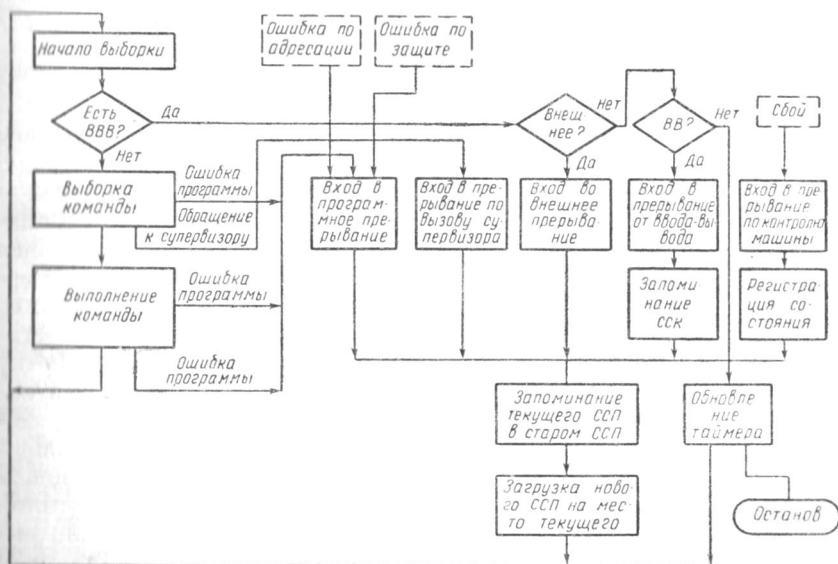


Рис. 7.3. Блок-схема прерываний: ВВВ — запрос на внешние, вводо-выводные прерывания, на обновление таймера, на останов.

Однако, чтобы не произошло прерывания программы, обрабатываемой прерыванием, до того, как она проделает необходимую работу, новое ССП должно замаскировать систему от дальнейших прерываний того же класса, что и обрабатываемое прерывание.

Общая блок-схема прерываний с учетом приоритетов между классами приведена на рис. 7.3.

7.4. Виды прерываний

Прерывание по контролю машины. Прерывание этого вида происходит при обнаружении сбоя схемами контроля машины, если 13-й бит маски контроля машины равен единице и переключатель «Контроль» на пульте управления находится в нулевом положении.

При появлении сигнала от схем контроля машины и разрешенном прерывании выполнение текущей команды прекращается, в РАПП заносится фиксированный адрес 0008_{16} первой микрокоманды микропрограммы входа в прерывание по контролю машины. Эта микропрограмма осуществляет:

запись состояния ЦП в диагностическую область основной памяти (регистрацию),

исправление контрольных бит регистров ЦП,

исправление контрольных бит ячеек локальной памяти.

Блок-схема микропрограммы входа в прерывание по контролю машины приведена на рис. 7.4.

В начале микропрограммы анализируется, когда произошел сбой. Если он произошел при первоначальной загрузке программы, то происходит «тяжелый останов», блокирующий выработку главных и рабочих синхросигналов. Если $ТПЗП=0$, то определяется индикатор сбоя: канал или ЦП. Если сбой произошел во время микропрограммной приостановки ($ТЦП=1$) или выполнения одной из команд ввода-вывода (кроме команды

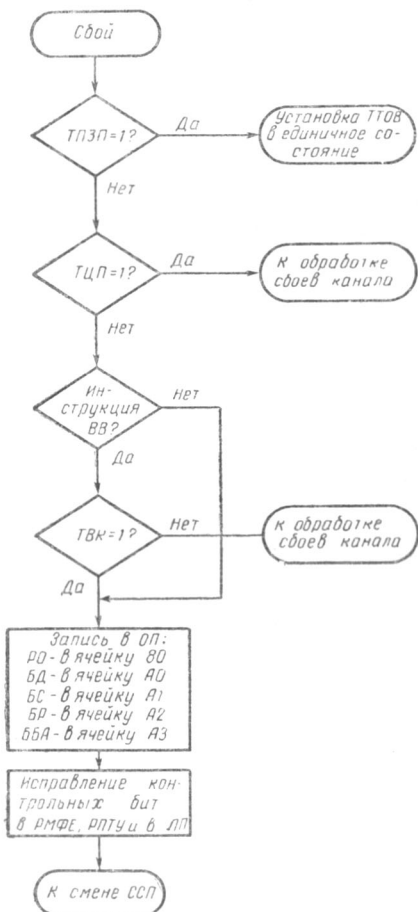


Рис. 7.4. Блок-схема микропрограммы входа в прерывание по контролю машины

ПРОВЕРИТЬ КАНАЛ), но не при выборке их ($ТВК=0$), то осуществляется переход к микропрограмме обработки сбоев канала, в противном случае — к микропрограмме записи состояния ЦП в диагностическую область основной памяти, которая начинается с записи в 80-ю ячейку основной памяти содержимого регистра ошибок.

В ячейки А0, А1, А2, А3 основной памяти записывается содержимое РБР, РБД, РБС и БАА. Затем в РМФЕ, РПТУ исправляются контрольные биты путем «прогона» содержимого этих регистров через БА.

Микропрограмма гашения локальной³ памяти выполняет исправление контрольных бит в регистрах общего назначения и регистрах с плавающей запятой, текущем ССП и информации, хранящейся в области разгрузки локальной памяти. После этого выполняется процедура смены ССП. Старое ССП запоминается в ячейке 30, причем код прерывания в нем становится равным нулю. Новое ССП загружается из ячейки 70.

Программные прерывания. Особые ситуации, возникающие в ходе выполнения программы из-за неверного использования команд, ошибочного задания адресов и данных, вызывают прерывания, называемые программными.

При программном прерывании текущее ССП записывается в ячейку 28, а новое берется из ячейки 68 постоянно распределенной области основной памяти.

Существует пятнадцать причин программных прерываний:

- некорректность кода операции (ПНКО),
- привилегированная операция (ПППО),
- некорректность команды ИСПОЛНИТЬ (ПНКИ),
- нарушение защиты памяти,
- неправильная адресация,
- неправильная спецификация (ПНСП),
- неправильные десятичные данные (ПНДД),
- переполнение с фиксированной запятой (ПДФЗ),
- некорректность деления с фиксированной запятой (ПДФЗ),
- десятичное переполнение (ПДПП),
- некорректность десятичного деления (ПДДН),
- переполнение порядка (ПППП),
- исчезновение порядка (ПИСП),
- потеря значимости (ППЗН),
- некорректность деления с плавающей запятой (ПДПЗ).

Коды прерывания, соответствующие этим причинам, указаны в табл. 7.2. Для каждой причины прерываний имеется отдельная микропрограмма входа в процедуру смены ССП. На рис. 7.5 а, г приведена структура микропрограммы для следующих причин прерываний: ПНКО, ПППО, ПНКИ, ПНСП, ПНДД, ПДФЗ, ПППП, ППЗН, ПДПЗ.

Для маскируемых программных прерываний в микропрограмме входа проверяется также маска. Структура таких микропрограмм для ПДФЗ, ПДПП, ПИСП дана на рис. 7.5 б, г. Маска для ППЗН

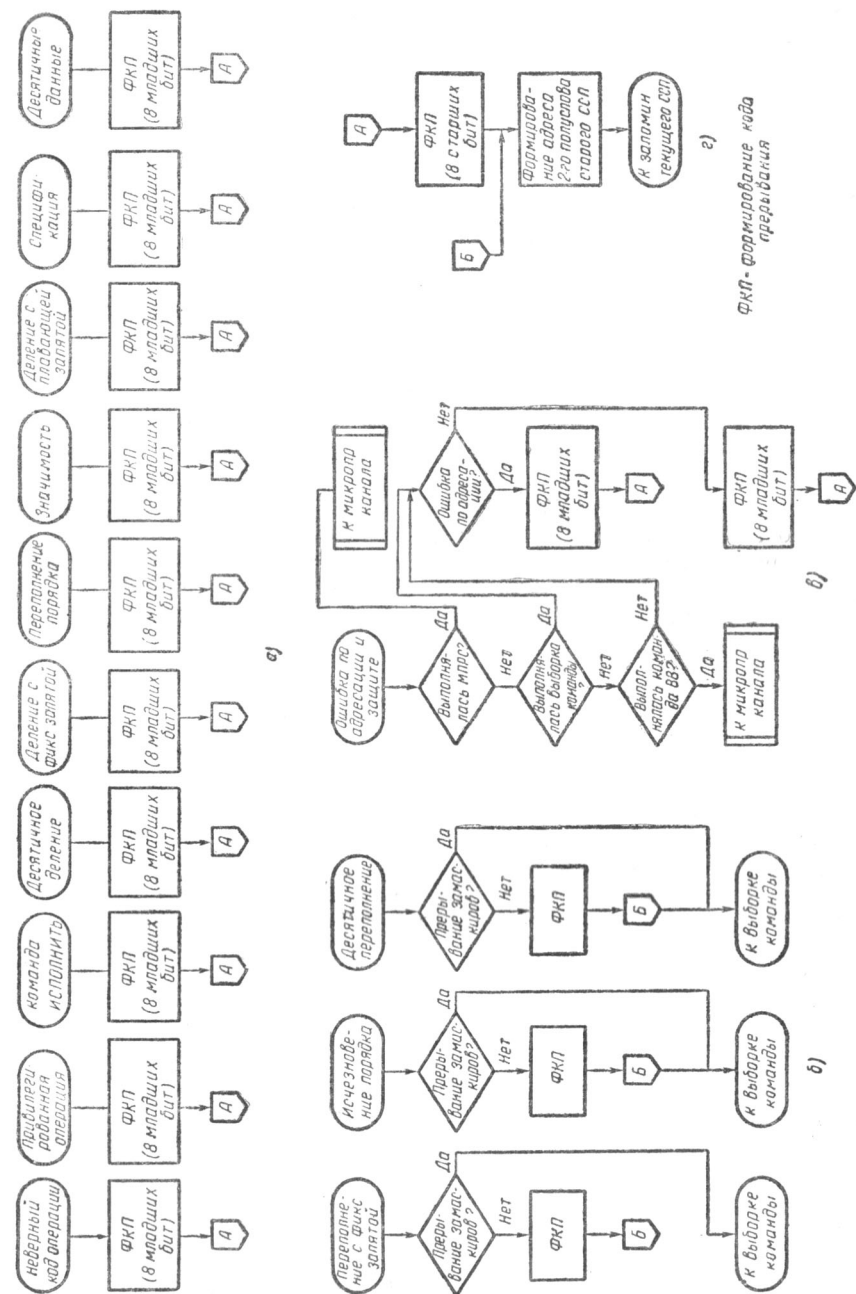


Рис. 7.5. Блок-схемы микропрограммы входов в программы прерываний: а—входы немаскированных прерываний, б—входы маскированных прерываний, в—входы для прерываний по защите и адресации, г—общая часть

проверяется в команде и определяет способ завершения команды.

Для любой команды может произойти только одно программное прерывание, которое и указывается в старом ССП. Наличие некоторого программного прерывания не исключает возможности одновременного существования причин для других программных прерываний. В старом ССП будет зафиксирована та ошибка, которая первой обнаружится в процессе выборки и выполнения команды.

Четыре из пятнадцати причин программных прерываний могут быть замаскированы маской программы, занимающей 36—39-й биты ССП. Программное прерывание может произойти только в том случае, если соответствующий бит маски равен единице. Если бит маски равен нулю, прерывание игнорируется. Программные прерывания не хранятся.

Все программные ошибки обнаруживаются микропрограммно, за исключением неверной адресации и нарушения защиты памяти, определяемых аппаратной логикой.

Неверная адресация вызвана обращением к несуществующей ячейке основной или мультиплексной памяти.

На рис. 7.6 изображена схема обработки ошибок по адресации.

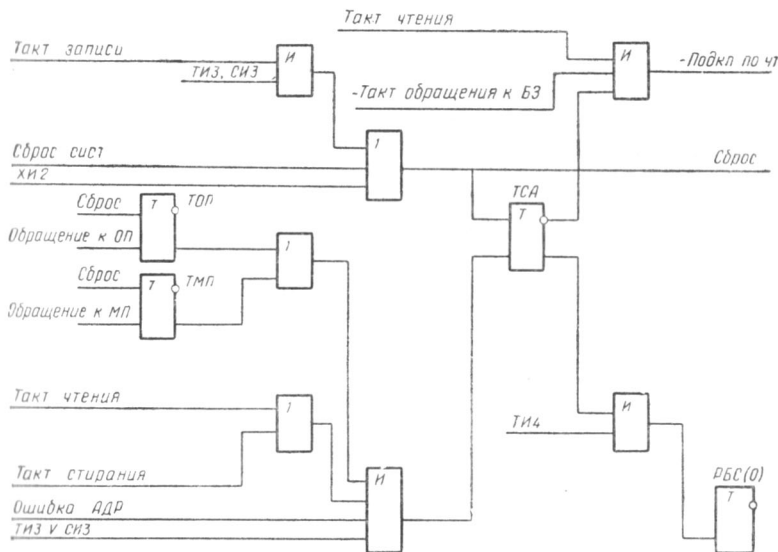


Рис. 7.6. Схема обработки ошибок по адресации

При обращении к несуществующей ячейке основной памяти выдается сигнал ОШИБКА АДР, указывающий на ошибку. Этот сигнал опрашивается в тактах чтения или стирания информации из основной или мультиплексной памяти, осуществляемых ЦП или каналами. При отсутствии обращения к памяти шина ОШИБКА АДР находится под низким потенциалом. В случае правильной работы низкий потенциал сохраняется и в момент чтения или стирания информации. Ошибка по адресации, обнаруженная во время

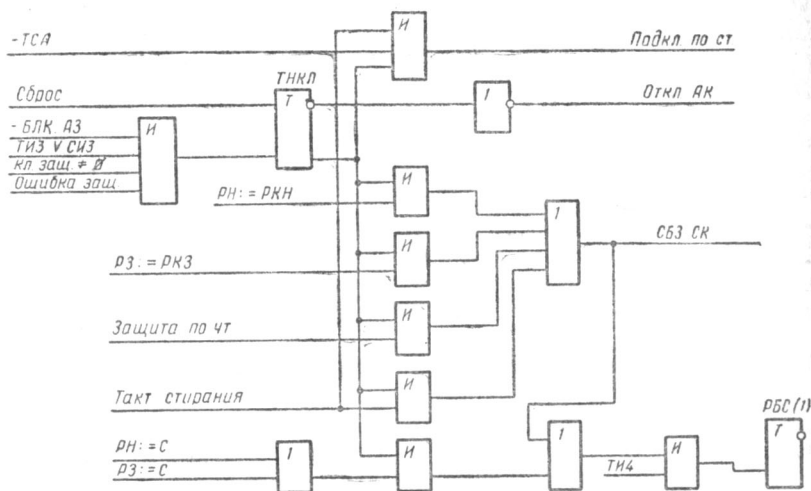


Рис. 7.7. Схема обработки ошибок по защите

работы ЦП или при выполнении микропрограммной приостановки, запоминается в триггере РБС (0).

Подключение входов регистра НЗ к выходным шинам данных ОП (КШЧ) происходит (вырабатывается сигнал ПОДКЛ. ПО ЧТ) только в такте чтения при отсутствии ошибки по адресации и обращении к блоку защиты (БЗ).

Нарушение защиты памяти в режиме защиты по «чтению и записи» происходит, если ключ памяти ячейки ОП, к которой произошло обращение, не соответствует ключу защиты в текущем ССП.

Если блок защиты работает в режиме защиты по «записи», то нарушение защиты памяти происходит при несоответствии ключей при попытке записи информации в память.

На рис. 7.7 приведена схема обработки ошибок по защите.

Факт несоответствия ключей отражается установкой триггера ТН4 в единичное состояние при появлении из блока защиты сигнала ОШИБКА ЗАЩ и отсутствии сигнала блокировки ошибок по адресации и защите БЛК АЗ.

При несоответствии ключа защиты ключу памяти (с целью предотвращения искажения информации после того, как извлекаемая из памяти информация занесена в РНЗ) происходит блокировка входных шин РНЗ сигналом ОТКЛ АК, по которому блокируется занесение информации из БА и каналов в РНЗ. Если несоответствие ключей обнаружено во время стирания информации из памяти, то с целью сохранения этой информации вырабатывается сигнал ПОДКЛ ПО СТ, по которому производится подключение КШЧ к РНЗ. Однако подключения не произойдет, если при этом будет обнаружен и сбой по адресации (сигнал ТСА).

При работе блока защиты в режиме защиты по «чтению и записи» несоответствие ключа памяти ключу защиты в текущем ССП указывает на ошибку по защите. Работу блока защиты в режиме защиты по «чтению и записи» определяет сигнал ЗАЩ ПО ЧТ (рис. 5.12).

При работе блока защиты в режиме защиты по «записи» несоответствие ключей отражает собой по защите только в том случае, если имели место попытки записи в память измененной информации со стороны ЦП или канала (т. е. присутствовали сигналы $RH := C$, $RZ := C$ или $RH := PKN$, $RZ := PKZ$ соответственно).

Независимо от режима работы блока защиты несоответствие ключей при стирании (триггер ТСТ при этом находится в единичном состоянии) классифицируется как нарушение защиты памяти. Ошибка по защите, обнаруженная при работе ЦП, а также при выполнении микропрограммой приостановки, запоминается в триггере РБС (1).

Ошибка при обращении к несуществующей ячейке памяти и нарушение защиты памяти, обнаруженные во время микропрограммного гашения системы и в остановленном состоянии, игнорируются с помощью триггера блокировки прерываний — ТБЛПРВ (рис. 7.8), устанавливаемого при наличии соответствующего кода в поле УСТАНОВ микрокоманды. На время селекторной аппаратной и микропрограммой (мультиплексно-селекторной) приостановок, происходящих при остановленном состоянии ЦП, блокировка снимается, так как в это время присутствуют сигналы АПРС или ТЦП соответственно.

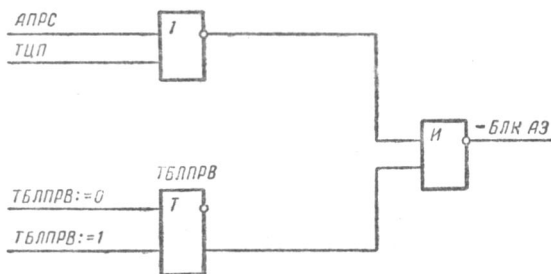


Рис. 7.8. Схема выработки сигнала БЛК АЗ

Ошибки по адресации и защите, хранящиеся в триггерах РБС (0) и РБС (1) при наличии определенных условий, устанавливаются в единичное состояние триггер ТЗА (рис. 7.9), в результате чего прекращается выработка тактовых импульсов и выполняется запуск холостых импульсов. Холостые импульсы сбрасывают адресный регистр ПП и заносят в него (по сигналу ТЗА) адрес 0004, являющийся адресом первой микрокоманды микропрограммы обработки сбоев по адресации и защите. Блок-схема этой микропрограммы представлена на рис. 7.5 в, г.

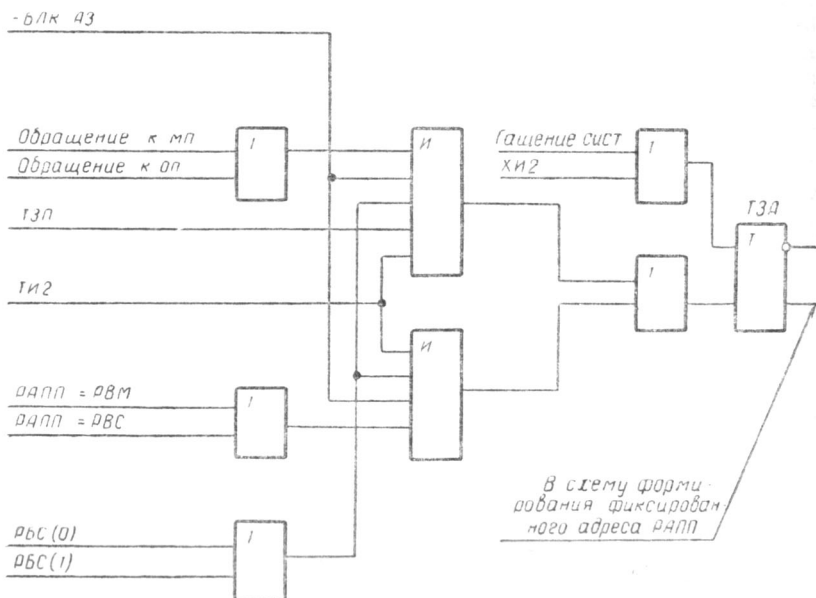


Рис. 7.9. Схема установки запроса на прекращение синхроимпульсов ТИ и на занесение фиксированного адреса

Микропрограмма вначале устанавливает, произошла ли ошибка во время выполнения микропрограммной приостановки (МПРС). Если ошибка обнаружена во время приостановки, то начинает выполняться каналная микропрограмма, обрабатывающая ошибку. После выполнения этой микропрограммы канал устанавливает запрос на ввод-выводное прерывание.

Если же сбой по адресации или защите происходит при работе ЦП, то микропрограмма анализирует, обнаружена ли ошибка во время выборки команды. Если сбой обнаружен во время выполнения команды, то по коду операции, хранящемуся в буфере команды (в локальной памяти), определяется, выполнялась ли команда ввода-вывода. Если эта команда выполнялась, то каналная микропрограмма фиксирует этот сбой в слове состояния канала (ССК). В случае ошибки при выполнении какой-либо другой команды, а также в выборке сбоя по адресации или защите определяется анализом состояния триггеров РВС (0) и РВС (1). Соответственно формируется код прерывания и осуществляется переход к микропрограммам смены ССП, т. е. производится программное прерывание.

При одновременном появлении ошибки по адресации или ошибки по защите и запроса из канала на микропрограммную приостановку в первую очередь будет выполнена приостановка. Содержимое РВС при микропрограммной приостановке будет занесено в область разгрузки локальной памяти, т. е. сбоя по адресации и за-

щите будут храниться до конца микропрограммной приостановки в локальной памяти. При выполнении микропрограммы разгрузки в конце микропрограммной приостановки запросы на прерывание по адресации или защите вновь занесутся в соответствующий триггер РБС. Микрооперация РАПП:=РВМ или РАПП:=РВС последней микрокоманды микропрограммной приостановки (рис. 7.9) установит ТЗА.

Внешние прерывания позволяют ЦП реагировать на сигналы, поступающие от таймера, кнопки «Прерывание» на пульте управления и внешних объектов.

Запрос на прерывание от внешнего источника может прийти в любой момент и в один и тот же момент времени может прийти несколько запросов. Запросы сохраняются в РБК до тех пор, пока они не будут восприняты ЦП. Если произошло прерывание от внешнего источника, все запросы, ожидающие рассмотрение, воспринимаются одновременно.

Каждый запрос рассматривается только один раз. Если до начала обработки придет несколько запросов от одного источника, все равно произойдет только одно прерывание.

Внешнее прерывание может произойти только в том случае, если бит в маске системы (7-й бит РБР) равен единице, а выполнение текущей команды закончено.

При этом прерывании текущее ССП запоминается в ячейке 18 основной памяти, а новое ССП берется из ячейки 58.

Источник прерывания определяется битами 24—31 в коде прерывания старого ССП. Остальная часть кода прерывания (биты 16—23) сбрасывается.

Каждому из восьми источников внешних прерываний соответствует один бит в коде прерывания старого ССП (табл. 7.3).

Таблица 7.3

Причина внешнего прерывания	Бит кода прерывания в старом ССП
Внешний сигнал 7	31
Внешний сигнал 6	30
Внешний сигнал 5	29
Внешний сигнал 4	28
Внешний сигнал 3	27
Внешний сигнал 2	26
Кнопка «Прерывание»	25
Таймер	24

Прерывание от внешних объектов возникает (если установлено дооборудование прямого управления), когда хотя бы на одной из входных линий внешних сигналов появится сигнал от объекта.

Из восьми входных линий внешних сигналов шесть (2—7) подсоединяются к ЦП (к входам 2—7-го бит РБК), а две оставшиеся (0—1) также подсоединяются к ЦП, но никаких функций не выполняют (если внешним оборудованием является другой ЦП, вход-

ные линии внешних сигналов одного ЦП соединяются с восемью выходными линиями другого ЦП). Шины внешних сигнальных линий входят в состав дооборудования для прямого управления.

Прерывание с пульта возникает при нажатии кнопки «Прерывание». Прерывание по таймеру происходит, когда слово в 50-й ячейке основной памяти становится отрицательной величиной (обычно содержимое слова представляет собой интервал времени, уменьшаемый с частотой сети), означающий, что первоначально установленный интервал времени истек, и этот факт сигнализируется ЦП как запрос на прерывание от таймера.

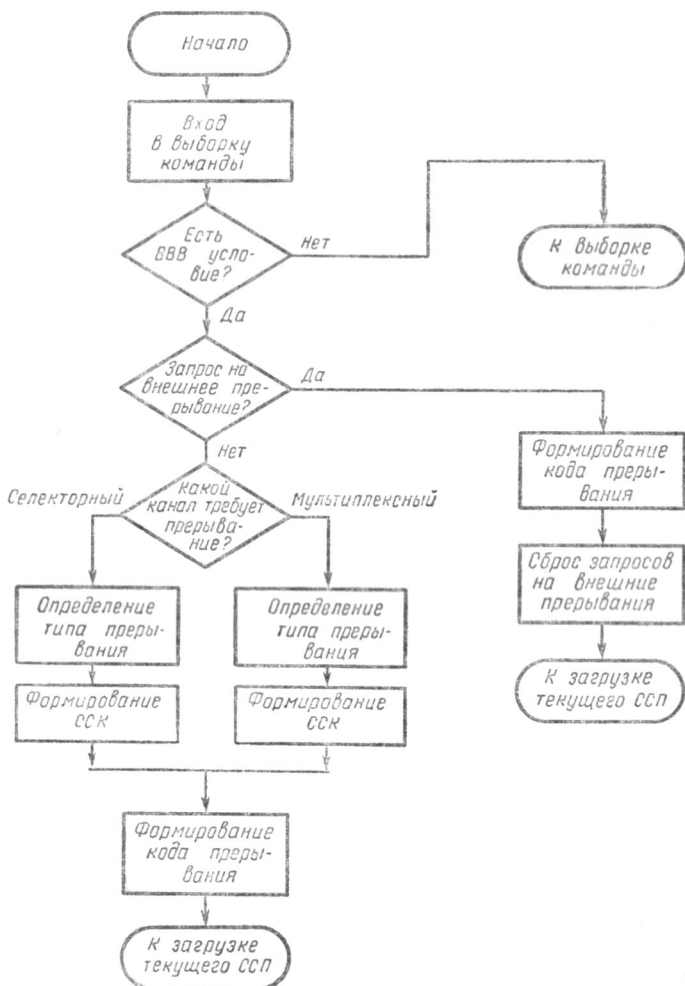


Рис. 7.10. Блок-схема микропрограммы входа во внешние и вводно-выводные прерывания

Если внешнее прерывание принято ЦП, то сначала выполняется микропрограмма входа во внешнее прерывание (см. рис. 7.10), в которой формируется адрес второго полуслова старого ССП внешнего прерывания для запоминания содержимого РБК в качестве кода прерывания. После этого выполняется общая для всех прерываний микропрограмма смены ССП.

Прерывание от ввода-вывода дает возможность реагировать ЦП на сигналы, поступающие от каналов и устройств ввода-вывода. Запросы на прерывание от ввода-вывода могут появиться в любой момент времени и одновременно может появиться несколько запросов. Запросы сохраняются в УВВ и каналах до тех пор, пока они будут восприняты ЦП. Среди запросов установлен приоритет.

Прерывание от ввода-вывода может произойти только после того, как выполнение текущей команды в ЦП закончено, а прерывание от канала, по которому пришел запрос, не замаскировано.

При прерывании текущее ССП заносится в ячейку 38 в качестве старого. В коде прерывания старого ССП биты 21—23 указывают номер канала, а биты 24—31 — номер устройства ввода-вывода, вызвавшего прерывание. Биты 16—20 старого ССП становятся равными нулям. Значение кода длины команды не определено, поэтому он принят равным 0. В качестве текущего загружается новое ССП из ячейки 78.

Перед входом в микропрограммы смены ССП выполняется микропрограмма формирования и запоминания в ячейке 40 основной памяти слова состояния канала (рис. 7.10).

Запросы внешних прерываний и ввода-вывода. Запросы на прерывания от мультиплексного (МК), селекторных каналов (КС1) и (КС2) хранятся соответственно в битах 3,4 и 6 РБР.

Установка запросов на прерывания выполняется микропрограммно или аппаратно по сигналам, поступающим из каналов (в случае программно управляемого прерывания).

Маски прерываний от канала и маска внешних прерываний также запоминаются микропрограммно в РБР (при загрузке нового ССП).

Внешние сигналы (2—7), сигналы запросов на прерывания от таймера и кнопки прерывания заносятся в РБК и хранятся там, пока прерывания не будут восприняты ЦП.

Наличие запросов на ввод-выводные или внешние прерывания проверяется после завершения выполнения текущей команды путем анализа в первой микрокоманде микропрограммы выборки условия прерывания программы. Это условие вырабатывается или при наличии запросов на прерывание от системы ввода-вывода, внешних объектов или запросов на останов машины, или запросов на продвижение таймера (рис. 7.11).

Если это условие присутствует, то после первой микрокоманды микропрограммы выборки команды выполняется ветвление в микропрограмму, определяющую причину прерывания программы. Блок-схема этой микропрограммы приведена на рис. 7.12.

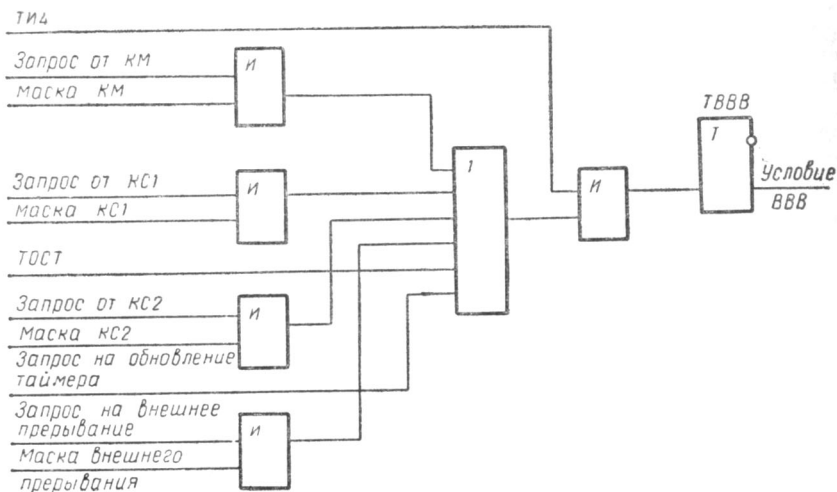


Рис. 7.11. Схема формирования условия ВВВ

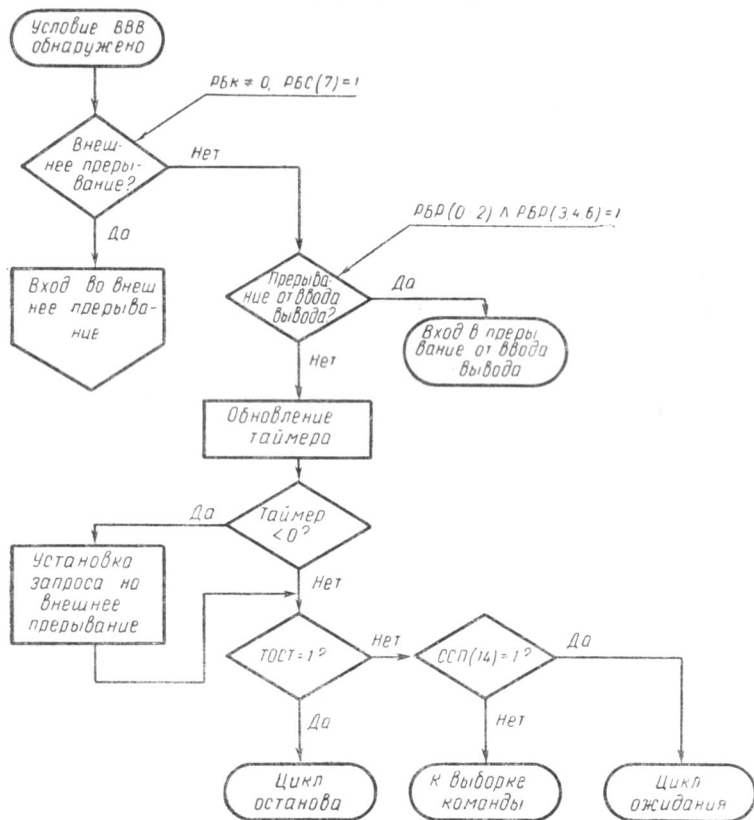


Рис. 7.12. Блок-схема микропрограммы, определяющей причины ВВВ

7.5. Запрещение прерываний посредством маскирования

Иногда возникновение прерывания нежелательно. Наиболее наглядно это видно на примере прерываний от ввода-вывода при одновременной работе с двумя или тремя каналами. Допустим, что произошло прерывание от мультиплексного канала, т. е. при завершении операции ввода-вывода по запросу от мультиплексного канала (КМ) текущее ССП автоматически запоминается в ячейку 38 старого ССП прерывания от ввода-вывода, чтобы сообщить о причине прерывания (номер канала и устройства). Новое ССП прерывания от ввода-вывода затем выбирается из ячейки 78 основной памяти и становится текущим. Адрес команды (40—63-й биты) этого нового ССП указывает первую команду программы супервизора, которая обрабатывает прерывания по вводу-выводу.

Если во время выполнения этой программы закончилась операция ввода-вывода в одном из селекторных каналов (КС), то он выдает запрос в процессор на прерывание. Если бы второе прерывание (от КС) было разрешено в тот момент, когда супервизор обслуживал первое прерывание (от КМ), текущее ССП необходимо было бы снова запомнить в ячейку 38. При этом старое ССП проблемной программы было бы уничтожено.

Маска системы. Для того чтобы супервизор мог предотвратить это второе нежелательное прерывание на время, пока не будет обработано первое, используется маска системы (биты 0—7 ССП), с помощью которой можно замаскировать (предотвратить) прерывание каждого из шести (в ЕС-1020 — двух) селекторных каналов, мультиплексного канала и внешнее прерывание.

Если все биты этой маски равны нулю, прерывания от всех каналов и внешние прерывания будут замаскированы. Каждый из каналов может быть замаскирован и отдельно, для чего соответствующий бит маски системы устанавливается в нулевое состояние.

Маска системы, определяющая возможность маскирования прерывания по вводу-выводу и внешнего прерывания, содержится в текущем ССП.

В рассматриваемом примере для предотвращения второго прерывания (от КС) на то время, пока первое (от КМ) будет полностью обработано, биты маски системы нового ССП, выбранного из ячейки 78, должны содержать нули (по крайней мере маска селекторного канала, выдавшего запрос на прерывание). Действительно, новое ССП после выборки станет текущим, а так как биты маски системы будут содержать нули, то второе прерывание не произойдет.

Замаскированные внешние прерывания и прерывания по вводу-выводу остаются необработанными. Как только соответствующие биты маски системы станут равны единице, будет обрабатываться следующее прерывание (от канала или внешнее).

Как и для других программ обработки прерывания, последней командой программы обработки прерываний ввода-вывода должна быть команда ЗАГРУЗИТЬ ССП. Старое ССП при этом должно

быть выбрано из основной памяти и вновь будет управлять проблемной программой в качестве текущего. Как только это произойдет, снова станет возможным прерывание от ввода-вывода, поскольку маска системы в ССП проблемной программы будет содержать единицу во всех битах.

Маска контроля машины. Прерывания по контролю машины могут быть замаскированы с помощью 13-го бита ССП. Машинные сбои будут игнорироваться и прерывание от схем контроля машины не произойдет, если этот бит содержит нуль. Но такое значение этого бита необычно при работе системы. Как правило, он равен единице. Следует также отметить, что на пульте управления системой есть переключатель, который может использоваться для останова при возникновении машинного сбоя или блокировки этого сбоя. При работе системы этот переключатель выключен и 13-й бит ССП равен единице. Это означает, что при возникновении машинного сбоя происходит прерывание по контролю машины. Если 13-й бит текущего ССП равен нулю, машинный сбой не вызывает прерывания в этот момент, и система продолжает выполнение операций. Однако информация о сбое хранится и прерывание будет обрабатываться тогда, когда 13-й бит ССП станет равным единице.

Маска программы. Программные прерывания могут быть вызваны 15 причинами (см. табл. 7.2). В некоторых случаях четыре из них могут не рассматриваться как программные ошибки. Например, если один из регистров общего назначения используется в программе в качестве счетчика, может возникнуть необходимость проверить его на переполнение, которое не должно расцениваться как ошибка в программе. Поэтому маска программы в текущем ССП доступна для программиста и с ее помощью можно замаскировать прерывания, вызванные в результате следующих четырех причин:

- переполнения с фиксированной запятой;
- десятичного переполнения;
- исчезновения порядка;
- потери значимости (значимость).

Маской программы являются биты 36, 37, 38 и 39 ССП.

Остальные 11 причин всегда расцениваются как программные ошибки и вызывают прерывания.

Прерывание по вызову супервизора также не может быть замаскировано.

7.6. Состояние процессора

Состояние «Ожидание». 14-й бит ССП характеризует состояние «Ожидание». Если этот бит равен нулю, команды выбираются и выполняются обычным способом: после выполнения очередной команды выбирается следующая команда, адрес которой указан в поле адреса команды ССП. Если же 14-й бит ССП равен единице, выборка и выполнение следующей команды не

производятся—система находится в состоянии ожидания до тех пор, пока не произойдет прерывание, которое изменит ССП. Это новое ССП должно содержать в 14-м бите нуль.

Изменить состояние «Ожидание» на состояние «Задача» могут только прерывания от ввода-вывода и внешние прерывания. Прерывания от схем контроля машины, программные прерывания или прерывания при обращении к супервизору могут происходить только тогда, когда процессор находится в состоянии «Задача» (работа) и занят выполнением проблемной программы.

Состояние «Задача». Существует несколько команд, выполняемых только в тех случаях, когда они выбираются супервизором. Если бы проблемная программа попыталась выполнить одну из этих команд, фиксировалась бы программная ошибка. Команды, которые могут выполняться только супервизором, но не проблемной программой, называются *привилегированными*.

Система узнает о том, что выполняется проблемная программа, если 15-й бит ССП равен единице. Выборка привилегированной команды в этом случае вызывает программное прерывание. Попытка выполнения привилегированной команды проблемной программой обнаруживается супервизором с помощью кода прерывания в старом ССП. Очевидно, что 15-й бит в пяти новых ССП, находящихся в постоянно распределенных ячейках памяти, должен содержать нуль. Единица в 15-м бите старого ССП указывает, что была прервана проблемная программа.

Изменение состояния. Супервизор может изменять состояние машины в любой необходимый момент путем изменения отдельных полей или всего текущего ССП. В то же время лишь некоторые из этих полей могут быть изменены проблемной программой:

Биты	Поле	Изменяется
0—7	Маска системы	Командой УСТАНОВИТЬ МАСКУ СИСТЕМЫ
8—11	Ключ защиты	При возникновении прерываний
12—15	Состояние	При возникновении прерываний
16—31	Код прерывания	При возникновении прерываний
32—33	Код длины команды	При выполнении программы
34—35	Код условия	При выполнении ряда команд
36—39	Маска программы	Командой УСТАНОВИТЬ МАСКУ ПРОГРАММЫ
40—63	Адрес команды	При выполнении программы

Как видно, некоторые поля текущего ССП изменяются под воздействием специальных команд. Другие поля могут быть изменены только при изменении всего ССП. Замена всего ССП полностью возможна двумя способами: путем прерывания и с помощью команды ЗАГРУЗИТЬ ССП.

Команда ЗАГРУЗИТЬ ССП является привилегированной. Она может использоваться только супервизором. В проблемной программе для перехода от состояния «Проблема» («Задача») к состоянию «Супервизор» может быть использована команда

ВЫЗОВ СУПЕРВИЗОРА. Кроме этой команды, поля текущего ССП изменяют еще две команды: **УСТАНОВИТЬ МАСКУ СИСТЕМЫ** и **УСТАНОВИТЬ МАСКУ ПРОГРАММЫ**.

Команда **УСТАНОВИТЬ МАСКУ СИСТЕМЫ** является привилегированной, поскольку маска системы связана с прерыванием по вводу-выводу. А так как управлять всеми операциями ввода-вывода должен супервизор, то эта команда и четыре команды ввода-вывода являются привилегированными.

Последняя команда не является привилегированной, поэтому она может использоваться проблемной программой для изменения маски программы в ССП. Кроме того, эта команда изменяет и 34-й и 35-й биты ССП, т. е. код условия.

7.7. Размещение текущего ССП

Выполнение программы происходит под управлением текущего ССП. Часть информации, содержащейся в текущем ССП, хранится в регистрах процессора, а часть—в локальной памяти, начиная с ячейки 88 (см. рис. 6.2 и табл. 7.4).

Маска системы хранится в РБР. Для масок каналов (одного мультиплексного и двух селекторных) отведены 0,1 и 2-й биты соответственно. Маска внешних прерываний хранится в 7-м бите РБР. При загрузке ССП маска системы запоминается как в указанных битах РБР, так и в ячейке 88 локальной памяти.

Ключ защиты текущего ССП хранится в 4—7-м битах РБЗ и в 0—3-м битах ячейки 89 локальной памяти.

Признак КОИ-8, маска контроля машины, признак состояния ОЖИДАНИЕ, признак состояния ЗАДАЧА хранятся в 4, 5, 6 и 7-м битах соответственно ячейки 89 локальной памяти. Кроме того, маска контроля машины хранится в 5-м бите РБР, признак ждущего состояния — во 2-м бите РБД.

Код прерывания в текущем ССП (ячейки 8А и 8В локальной памяти) не имеет смысла, он формируется во время прерывания и запоминается в старом ССП.

Код длины команды получается путем модификации двух первых бит кода операции команды, который хранится в буфере команды (ячейка 98 локальной памяти). Преобразование выполняется при прерывании и служит для отражения количества условий в текущей команде.

Код условия заносится в 6 и 7-й биты РБС. Во время выполнения команд, устанавливающих код условия, эти биты РБС могут использоваться микропрограммой как обычные универсальные триггеры. Позиции, обозначенные как код длины команды и код условия в локальной памяти (ячейка 8С), не имеют смысла.

Маска программы находится в локальной памяти (ячейка 8С) (биты 4—7).

Адрес команды текущего ССП хранится либо в РМФЕ, либо в локальной памяти (ячейки 8D, 8E и 8F). Адрес команды переписывается из РМФЕ в локальную память в тех случаях, когда в

ходе выполнения команды необходимо использовать РМФЕ для обработки информации.

Если адрес команды записывается в локальную память, триггер адреса команды (ТАК) устанавливается в единичное состояние.

Таблица 7.4

Месторасположение частей текущего ССП

Части текущего ССП	Месторасположение	
	ячейка локальной памяти	бит регистра ЦП
Маска системы	88	РБР (0,1,2,7)
Ключ защиты	89	РБЗ (4—7)
Признак КОИ-8	89	—
Маска контроля машины	89	РБР (5)
Признак состояния «Ожидание»	89	РБД (2)
Признак состояния «Задача»	89	—
Код прерывания	—	—
Код длины команды	—	—
Код условия	—	РБС (6—7)
Маска программы	8С	Нет
Адрес команды	8D, 8E, 8F	РМФЕ

7.8. Микропрограммы смены ССП

Микропрограммы смены ССП. Прерывание любого класса заключается в записи текущего ССП в ячейки старого ССП и загрузке нового ССП в качестве текущего. Микропрограммы для запоминания текущего ССП и загрузки нового ССП для всех пяти классов прерываний являются общими. Перед выполнением этих общих микропрограмм для каждого класса прерывания выполняется своя микропрограмма входа в прерывание, функции которой в основном заключаются в формировании кода прерывания и адреса второго полуслова старого ССП, по которому запоминается этот код.

Код прерывания формируется с помощью соответствующей константы микрокоманды. Адрес основной памяти, где расположено второе полуслово старого ССП, формируется также с помощью константы в РГРИ.

Микропрограмма запоминания текущего ССП начинается с установки нулевого ключа защиты (рис. 7.13). Сформированный код прерывания записывается во второе полуслово старого ССП обрабатываемого прерывания. Затем запоминается адрес команды (два байта без старших бит) в четвертом полуслове старого ССП. Так как адрес команды может храниться или в РМФЕ, или в локальной памяти, то предварительно анализируется триггер местонахождения адреса команды (ТАК). Если ТАК находится в единичном состоянии, то сначала АК загружается из локальной памяти в РМФЕ, а уже затем запоминается в старом ССП.

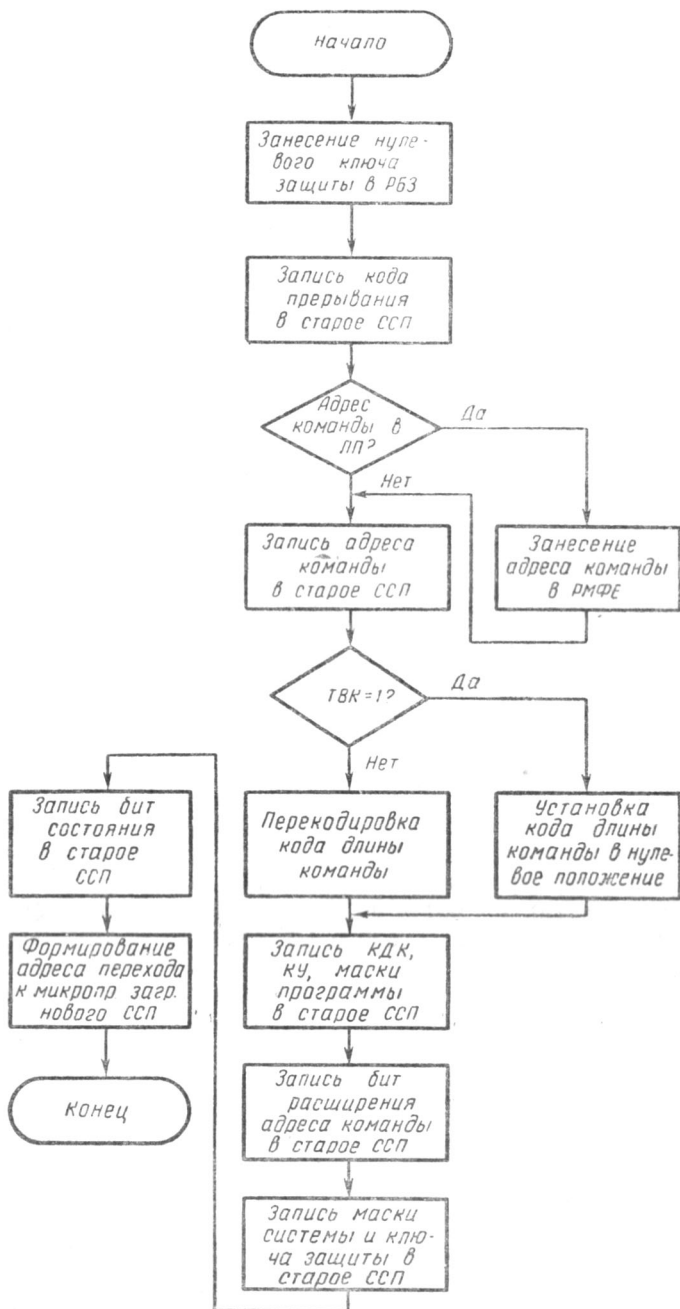


Рис. 7.13. Блок-схема микропрограммы запоминания текущего ССП

Следующими по порядку запоминаются в третьем полуслове старого ССП: код длины команды, код условия, маска программы и старшие биты адреса команды. Так как эти части текущего ССП «разбросаны» (код условия и старшие биты адреса команды хранятся на триггерах регистров ЦП, программная маска и исходная информация для кода длины команды—в локальной памяти), то происходит «собираение» их. При этом анализируется триггер выборки команды (ТВК), единичное состояние которого указывает, что прерывание произошло во время выполнения микропрограммы выборки команды. В этом случае код длины команды становится равным нулю, который указывает, что адрес команды, запомненный в старом ССП, не является адресом следующей команды.

Если же прерывание произошло после выборки, то код длины команды должен отражать длину прерванной команды в полусловах. С этой целью из буфера команды извлекается код операции, записанный туда во время выборки, и первые два бита его (0 и 1) изменяются по следующему правилу: код 00 преобразуется в 01, код 01 преобразуется в 10, код 10 не изменяется, код 11 не изменяется.

Получившийся код длины команды, 6 и 7-й биты РБС (код условия), программная маска из ячейки 8С локальной памяти, старшие биты адреса команд из РМ записываются в ячейку третьего полуслова соответствующего старого ССП.

Первое полуслово текущего ССП полностью хранится в локальной памяти (ячейки 88 и 89), поэтому оно читается из него и записывается без изменения в первое полуслово старого ССП. В конце микропрограммы запоминания ССП адрес первого полуслова старого ССП в РГРИ наращивается на величину 46, чтобы получить адрес четвертого полуслова нового ССП, так как микропрограмма загрузки ССП начинается с загрузки четвертого полуслова, т. е. адреса команды.

Микропрограмма загрузки нового ССП. Назначение микропрограммы—загрузить новое ССП из основной памяти в регистры процессора и локальную память, т. е. сделать его текущим. Блок-схема загрузки нового ССП приведена на рис. 7.14.

Начинается она с чтения четвертого полуслова нового ССП из основной памяти (по адресу в РГРИ) и занесения его в локальную память (ячейки 8Е и 8F) и РФ, РЕ. Затем из основной памяти извлекается третье полуслово ССП, причем код условия выделяется из него и заносится в 6 и 7-й биты РБС, а второй байт его (старшие биты адреса команд) передается в РМ. Третье полуслово после этого помещается в локальную память (ячейки 8С и 8D). Первое полуслово ССП, извлеченное из основной памяти, также заносится в локальную память (ячейки 88 и 89). Кроме того, маска системы запоминается в РБР (0, 1, 2 и 7-й биты), а маска контроля—в РБР (5-й бит). Признак состояния ожидания загружается во 2-й бит РБД.

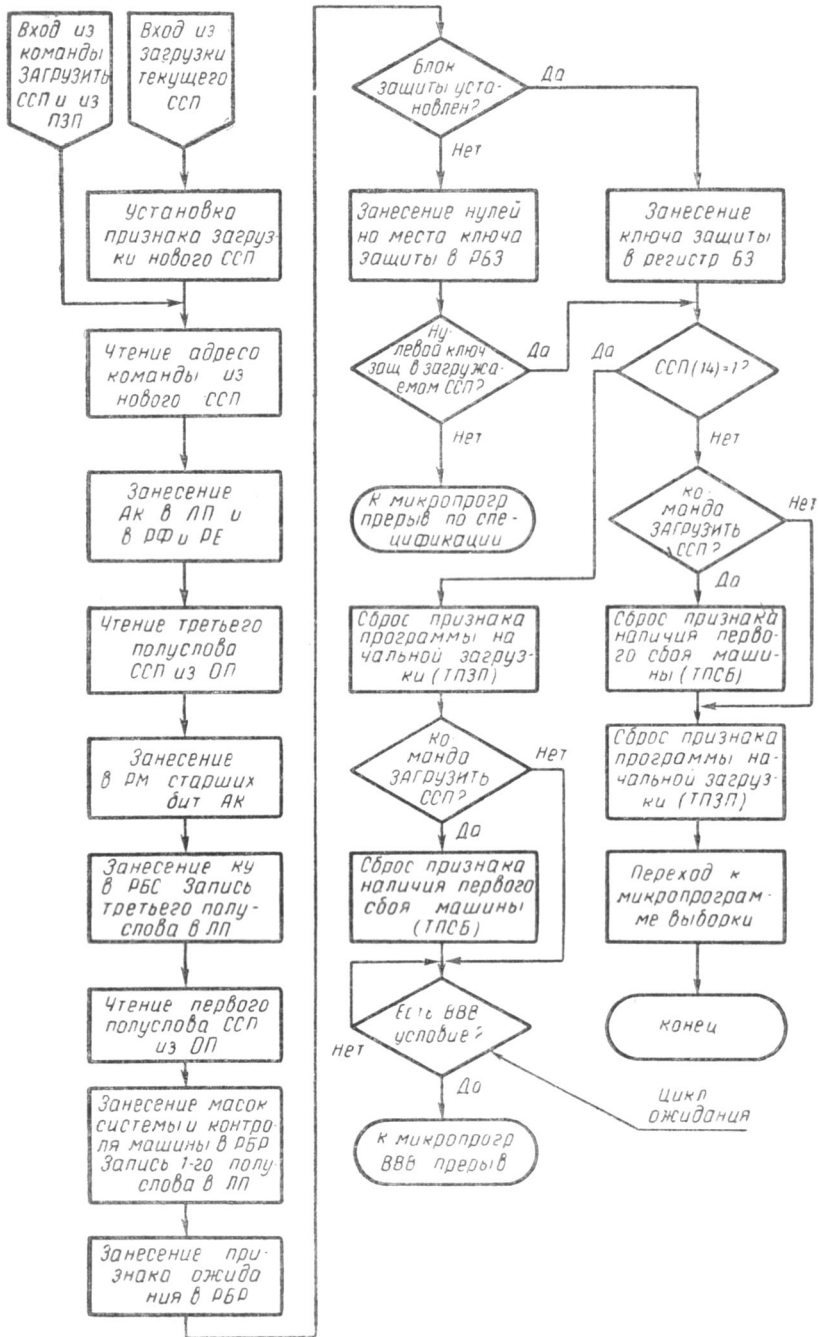


Рис. 7.14. Блок-схема микропрограммы загрузки нового SSP

Признак блока защиты (ТБЗ) определяет, имеется ли в процессоре дооборудование по защите памяти. Если оно есть (ТБЗ=1), то ключ защиты записывается в 4—7-й биты РБЗ. При отсутствии блока защиты анализируется ключ защиты в загружаемом ССП. В случае не нулевого ключа происходит переход к микропрограмме входа в программное прерывание (по спецификации). При нулевом ключе защиты продолжается процедура загрузки ССП.

Микропрограмма загрузки ССП содержит цикл ожидания, который вводится, если в новом ССП 14-й бит равен единице. Выход из этого цикла возможен только при наличии условия ВВВ (запрос на внешнее или введено-выводные прерывания, на остановленное состояние, на обновление таймера). Запрос на микропрограммную приостановку прерывает этот цикл лишь на время обслуживания каналов. Если признак ждущего состояния равен нулю, то происходит переход к микропрограмме выборки команды.

Микропрограмма загрузки нового ССП используется в команде ЗАГРУЗИТЬ ССП и при ПЗП (первоначальной загрузке программы) после выполнения ряда действий, указанных на блок-схеме рис. 7.15.

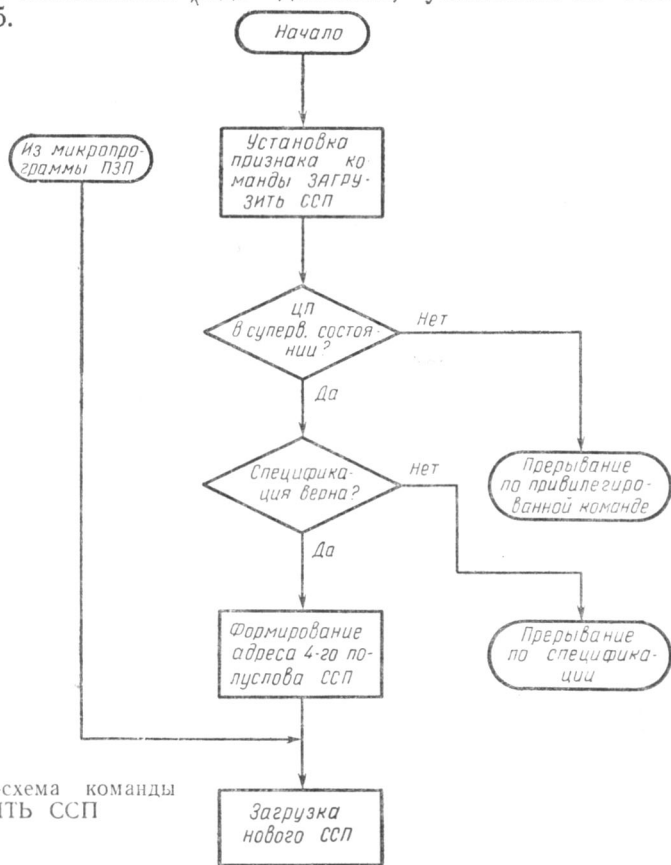


Рис. 7.15. Блок-схема команды ЗАГРУЗИТЬ ССП

Глава 8

СИСТЕМА КОНТРОЛЯ

8.1. Методы контроля

В процессоре ЕС-2020 применяется аппаратный контроль, осуществляемый непрерывно в течение всей его работы, основанный на методах контроля по модулю два и аппаратного дублирования.

Основным является метод контроля по модулю два, в частности контроль на нечет, осуществляемый побайтно, т. е. каждый байт данных сопровождается контрольным битом, дополняющим число единиц в битах байта до нечетного. Такой метод контроля позволяет выявить ошибки $(2l+1)$ -кратности ($l=0,1,2,\dots$), а также полную потерю информации байта.

Метод аппаратного дублирования используется для контроля работы арифметическо-логического блока (БА). Каждый бит информации представляется и обрабатывается парафазно: в прямой и в инверсной форме (двухпроводное дублирование). Результаты двухпроводной обработки сравниваются на выходе БА для каждого бита. Совпадение прямого и инверсного уровней хотя бы у одного бита выхода БА означает ошибку.

8.2. Формирование контрольных бит и выявление ошибок

Формирование контрольных бит, а также обнаружение двухпроводных сбоев и сбоев по нечету осуществляются стандартной схемой контроля (рис. 8.1), содержащей:

парафазную схему суммирования по модулю два («свертка») восьми двоичных бит байта информации $\pm И$ (0—7) (рис. 8.1,а). Результат свертки байта представляется в прямой и инверсной форме $+СМ|2|$ и $-СМ|2|$. При нечетном числе единиц в байте: $+СМ|2|=1$, а $-СМ|2|=0$, при четном: $+СМ|2|=0$, а $-СМ|2|=1$. Таким образом, получается парафазное значение

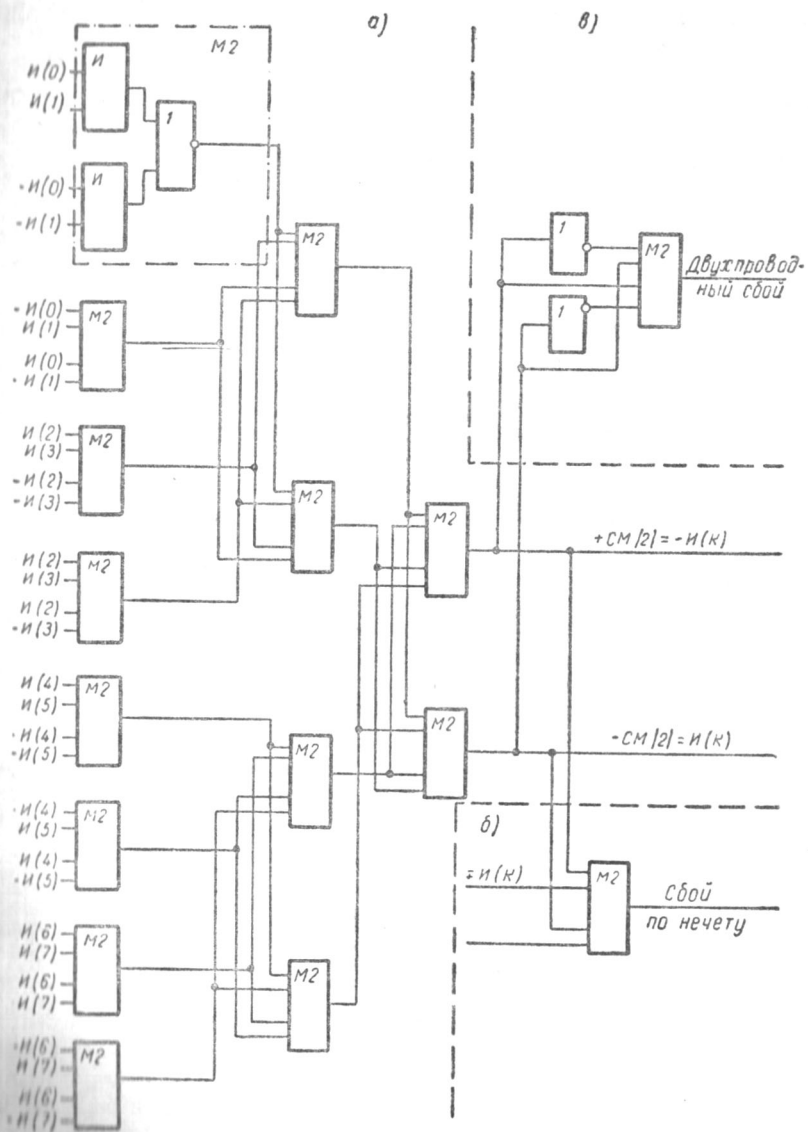


Рис. 81. Схема свертки ЛСВ на 1 байт: а—парафазная схема суммирования по модулю 2; б—схема контроля, предназначенная для сравнения; в—схема контроля, предназначенная для определения двухпроводной ошибки

контрольного бита байта информации ($\pm И(K)$), причем $-СМ|2| = И(K)$, а $+СМ|2| = -И(K)$;

вторую часть схемы контроля, предназначенную для сравнения контрольного бита ($\pm ИК$), пересылаемого с информацией, с контрольным битом, выработанным схемой свертки ($\pm И(K)$), и формирования сигнала СВОЙ ПО НЕЧЕТУ при несоответствии контрольных бит (рис. 8.1,б);

третью часть схемы контроля, предназначенную для определения двухпроводной ошибки в информации, подаваемой на вход схемы свертки, а также в самой схеме свертки (рис. 8.1,в).

Двухпроводная ошибка заключается в совпадении парафазных значений информации. Если такая ошибка возникла до схемы свертки, то эта схема транслирует ее на выход, что выражается в наличии одинаковых уровней на выходах $+СМ|2|$ и $-СМ|2|$, т. е. $+СМ|2| = -СМ|2|$. То же происходит и при ошибке в схеме свертки.

8.3. Распределение контрольных точек в процессоре

Для создания непрерывного контроля всего потока данных процессора предусмотрен следующий контроль*.

Контроль микропрограммного устройства управления. В микропрограммном устройстве управления на нечет контролируются адресный регистр ПП и информационный регистр ПП.

Для контроля адреса в каждой микрокоманде предусмотрен бит (РМК (62)), назначение которого—дополнение до нечетного числа единиц в адресе, по которому выбирается эта микрокоманда. После завершения чтения микрокоманды на выходе схемы контроля ЛСВ1 (рис. 8.2) появится сигнал сбоя, если прочиталось неадресованное слово. Так как занесение нового адреса в РАПП и запоминание сбоя в регистре ошибок выполняются одновременно (по ТИ2), сигнал сбоя РАПП запоминается в триггере по сигналу ЗАНЕСЕНИЕ.

Контроль информационного регистра ПП осуществляется схемой контроля ЛСВ2, которая выполняет суммирование по модулю два информации РМК (0—62) и сравнение результата с контрольным битом микрокоманды РМК (63). При сбое РМК по нечету схема контроля выдает сигнал СБРМК.

Контроль БА. В БА предусматривается: контроль на нечет входных регистров А и В; контроль на чет и двухпроводный контроль узла управляющих сигналов БА; двухпроводный контроль узлов обработки данных; формирование информации на выходе БА как с истинным, так и с неверным контрольным битом (для проверки работы схем контроля).

При контроле входных регистров определяются ошибки, возникшие при передаче и приеме информации по нечетности

* Наличие сбоя в какой-либо контрольной точке приводит к установке соответствующего бита регистра ошибок (РО).

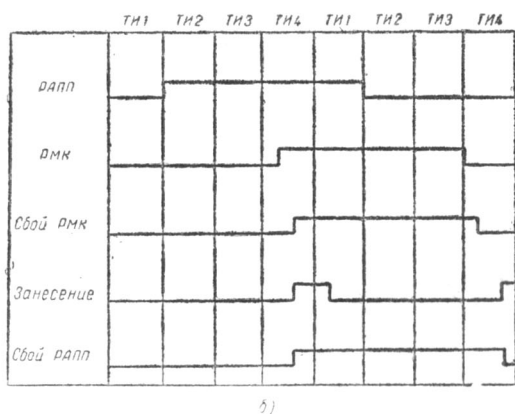
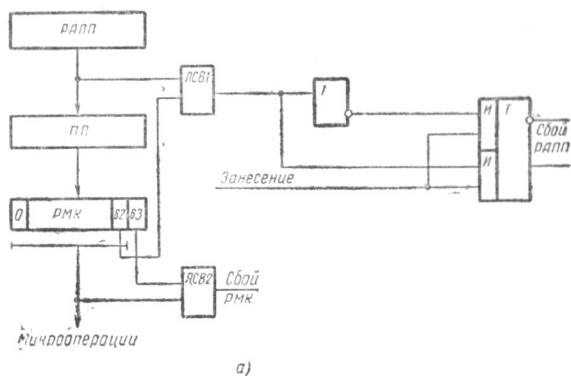


Рис. 8.2. Схема контроля микропрограммного устройства управления: а—формирование сбоев РАПП и РМК; б—временная диаграмма выработки сигналов сбоя постоянной памяти

(СБРАН и СБРВН — сбои РА и РВ по нечету) и двухпроводные (СБРАД, СБРВД — сбои РА и РВ двухпроводные).

Контроль работы узлов обработки информации основан на методе аппаратного дублирования. Путем сравнения парафазных сигналов каждого бита выходной информации БА такой метод позволяет выявить одиночную ошибку, возникшую в любом узле. Для определения ошибки информация с выхода БА в парафазной форме подается на стандартную схему контроля ЛСВ (рис. 8.3,а). Схема вырабатывает значение контрольного бита выходной информации $+С(K)$, $-С(K)$ и при наличии двухпроводной ошибки выдает сигнал СБСД (сбой выхода С двухпроводный). Для определения неверного контрольного бита выходной информации значение седьмого бита выходной информации ($\pm С(7)$) перед подачей на схему контроля может быть проинвертировано специальной схемой, управляющейся отдельной микрооперацией (НКР), если задан режим диагностики.

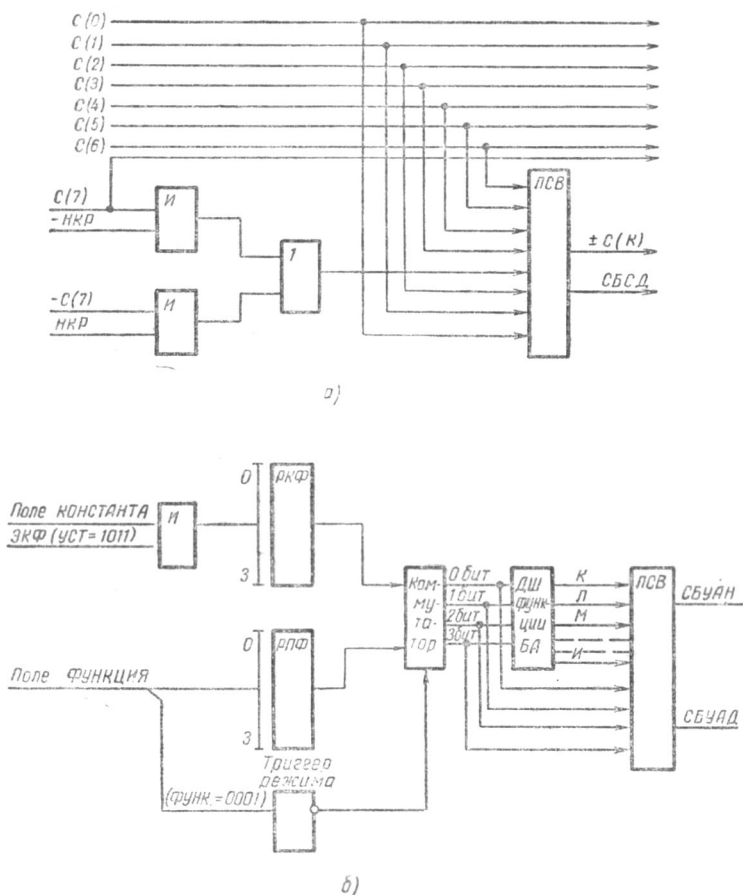


Рис. 8.3. Схема контроля БА: а—контроль выхода С; б—контроль узла управляющих сигналов

При контроле правильности выработки управляющих сигналов определяются ошибки четности и двухпроводного дублирования. Для определения ошибки четности введен дополнительный сигнал И, вырабатываемый таким образом, чтобы уравнивать четность числа единиц в коде операции БА и числа управляющих сигналов, необходимых для выполнения операции, заданной этим кодом. Лишь в случае кода 0001 это правило нарушается с целью возможности диагностики схемы контроля.

Сигнал сбоя управляющих сигналов по нечету (СБУАН) появляется при неравенстве нулю суммы по модулю два всех управляющих сигналов и разрядов кода операции (рис. 8.3, б), а сигнал двухпроводной ошибки (СБУАД) свидетельствует об одиночной ошибке в узле управляющих сигналов или в схемах контроля этого узла.

Контроль оперативной памяти и блока защиты. В оперативной памяти контролируются адресный (РМН) и информационный (РНЗ) регистры оперативной памяти.

Контроль на нечет адресного регистра осуществляется стандартной схемой контроля на два байта и дополнительной схемой свертки на два бита (для бит расширения адреса). При наличии сбойной информации на выходе РМН схема контроля выдает сигнал СБРМН.

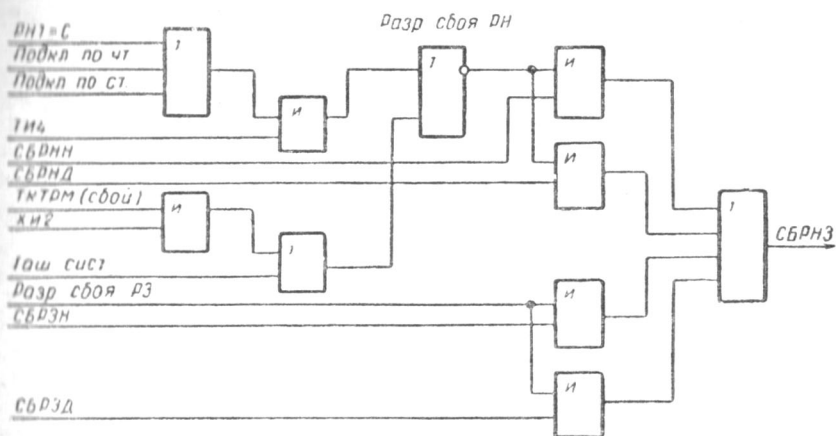


Рис. 8.4. Схема выработки сигнала СБРНЗ

Контроль информационного регистра памяти также выполняется стандартной схемой контроля на два байта, однако выдача сигнала сбоя дополнительно блокируется специальным триггером. На схеме рис. 8.4 показан триггер, разрешающий сбой РН (такой же триггер используется для РЗ). Триггер разрешает прохождение сигнала сбоя только при занесении информации в РНЗ из памяти или с выхода БА и запрещает выдачу сбоя сразу после включения питания (установка в нулевое положение сигналом ГАШ. СИСТ.), а также в холостом такте после фиксации сбоя машины. В последнем случае для того, чтобы не было повторных сбоев от одной и той же сбойной информации, запрещается прохождение сигнала сбоя.

Сбой РНЗ, зарегистрированный во время работы селекторных каналов с памятью, выдается в канал.

В блоке защиты контролируется правильность считывания из ЗУКП в информационный регистр (0—4-й биты РБЗ).

На рис. 8.5 приведена схема формирования сигнала сбоя РБЗ. При чтении ключа памяти из ЗУКП содержимое РБЗ (К, 0—4) проверяется на нечет схемой контроля ЛСВ1 и при сбое, возникшем в такте обращения к ОП, выдается сигнал СБРБЗ.

При передаче содержимого РБЗ на вход БА формирование контрольного бита РБЗ(К) осуществляется схемой ЛСВ2.

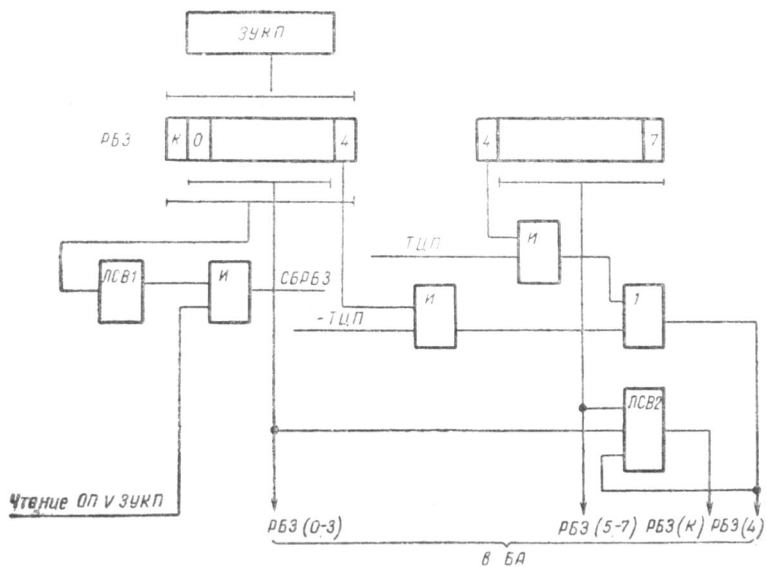


Рис. 8.5. Схема контроля блока защиты

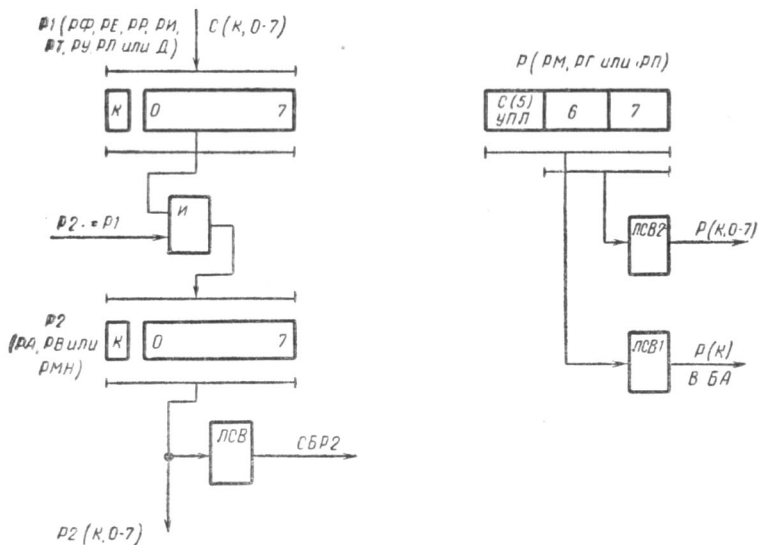


Рис. 8.6. Схема контроля адресных и общих регистров

Контроль регистров. В блоке регистров проверяется:

для регистров Ф, Е, Р, И, Т, У, Л или Д — правильность передачи информации с выхода БА через эти регистры в адресный регистр памяти и на входы БА. Эти регистры имеют триггеры контрольных бит байт информации, поступающей в них (рис. 8.6, а);

для регистров М, Г или П — правильность передачи с выхода регистра в РМН и на входы БА. У этих регистров есть схемы (рис. 8.6, б), формирующие контрольные биты для всего регистра (выдача в БА) и для двух младших бит регистров М, Г или П (выдача в ОП);

для регистров БК, БР, БС, БД или РО — правильность передачи с выхода регистра на входы БА. Контрольные биты этих регистров формируются схемами свертки на один байт (рис. 8.7).

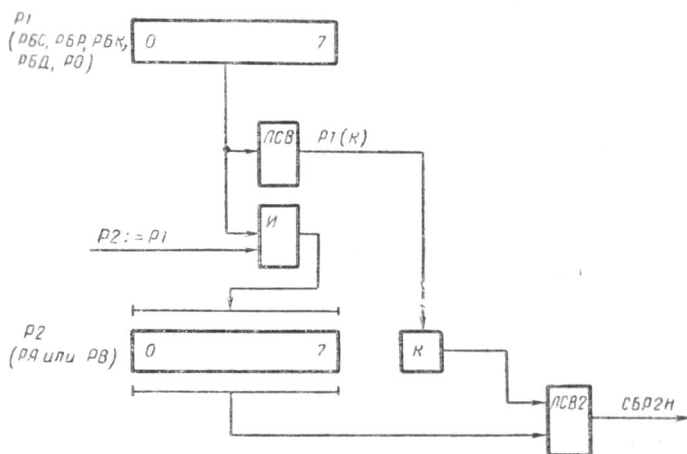


Рис. 8.7. Схема контроля служебных регистров (РБК, РБД, РБС, РБР, РО)

8.4. Обработка машинных сбояв

Если маска контроля машины (13-й бит текущего ССП) равна 1 и переключатель «Контроль» на пульте управления выключен (нулевое положение), то при обнаружении сбоя схемами контроля машины выполняются следующие действия:

запоминание сбоя в регистре ошибок;

останов тактовых импульсов (ТИ) и выработка холостых импульсов (ХИ). В такте ХИ в РАПП заносится начальный адрес микропрограммы входа в прерывание по машинной ошибке (адрес 0008);

пуск ТИ и выполнение микропрограммы входа в прерывание по машинной ошибке. Эта микропрограмма запоминает состояние процессора в момент сбоя (регистрация состояния), исправляет контрольные биты регистров ЦП и ячеек локальной памяти, а также производит смену ССП. Машинные ошибки, возникшие

при совместной работе ЦП с каналами, требуют дополнительного анализа, который должен выявить источник сбоя (ЦП или канал). Такой анализ выполняется в начале микропрограммы входа в прерывание по машинной ошибке. Если сбой обнаружен при работе канала, производится ветвление в микропрограмму проверки каналов;

смена ССП и обработка машинной ошибки супервизорной программой. Если в любой момент до окончания смены ССП появится вторая ошибка, то прекращается выработка главных и рабочих синхронимпульсов (тяжелый останов).

8.5. Регистр ошибок

Регистр ошибок имеет восемь бит (рис. 8.8):

0-й бит устанавливается по сигналу сбоя информационного регистра ЗУКП (регистра БЗ);

1-й бит устанавливается при сбое РА, если в РА заносилась информация;

2-й бит устанавливается при сбое РВ, если в РВ заносилась информация и триггер гашения системы (ТГС) сброшен. ТГС устанавливается в единичное положение в начале микропрограммы исправления контрольных бит регистров ЦП и ячеек локальной памяти. Это исправление выполняется передачей содержимого регистра или ячейки локальной памяти через БА, т. е. неверная информация, вызвавшая первую ошибку, может явиться причиной повторного сбоя и привести к тяжелому останову. Если происходила передача байта состояния БА на вход РВ, то сбой РВ по нечету блокируется, так как у байта состояния отсутствует контрольный бит;

3-й бит устанавливается при сбое РМН в такте обращения к памяти (в такте чтения, стирания или записи);

4-й бит устанавливается по сигналу сбоя РНЗ, если не было передачи в РНЗ из канала (отсутствует сигнал ДАН ИЗ КАНА) и триггер ТГС сброшен. Так как передача в РНЗ из канала и прием сигнала сбоя в РО выполняются по третьему синхронимпульсу такта, то в момент передачи может быть зарегистрирован ложный сбой РНЗ;

5-й бит устанавливается при сбое РМК, если триггер аппаратного гашения (ТАГ) находится в нулевом состоянии (ТАГ=1 в первом холостом такте, который выполняется после нажатия кнопки пульта «Гашение», «Загрузка» или «Питание вкл.»);

6-й бит устанавливается по сигналу сбоя РАПП, если сброшен триггер ТАГ;

7-й бит устанавливается при сбое управления БА или при двухпроводном сбое БА.

Установка триггеров регистра ошибок производится по третьему синхронимпульсу такта, в котором сбойная информация принимается в контролируемый регистр (РА, РВ, РБЗ, РМН), или следующего такта (РНЗ), а также по второму синхронимпульсу

вых операций, то при любом занесении информации в регистр ошибок будет сформирован сигнал сбоя процессора (СБМАШ) (рис. 8.9), который блокирует выработку синхроимпульсов ТИ и

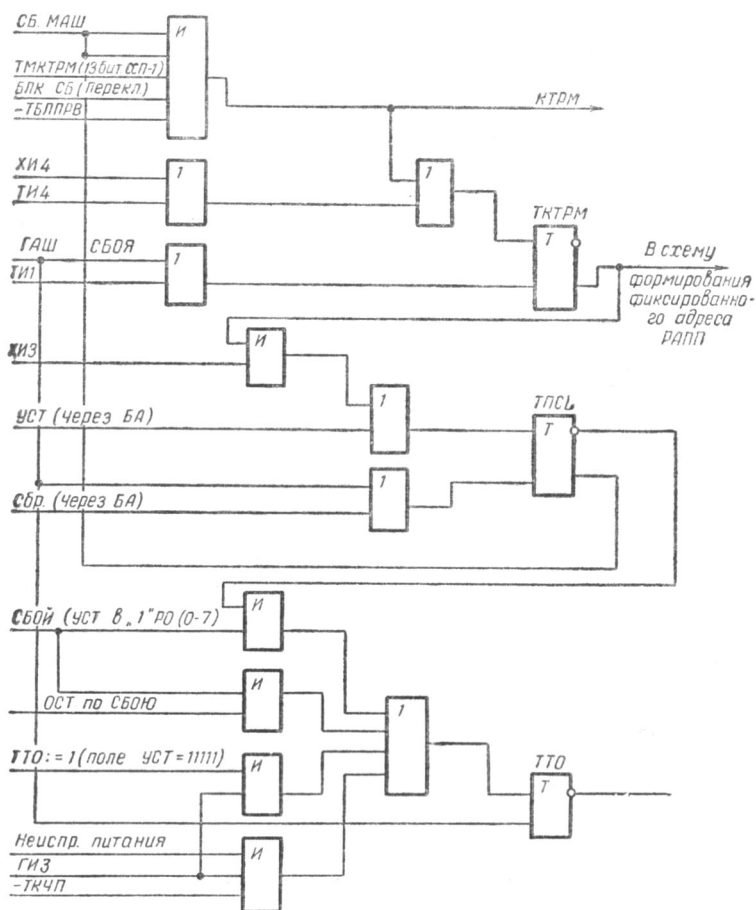


Рис. 8.9. Схема обработки сбоя машины

запускает схему формирования синхроимпульсов ХИ. По сигналу СБМАШ четвертым синхроимпульсом такта, в котором выполнялось занесение в РО, триггер контроля машины (ТКТРМ) устанавливается в единичное состояние. Далее следует холостой такт, в котором по импульсу ХИ2 и единичному выходу ТКТРМ в РАПП заносится начальный адрес микропрограммы входа в прерывание по контролю машины (008), а по импульсу ХИ3 устанавливается в единичное состояние триггер первого сбоя (ТПСБ). После установки ТПСБ повторный сбой приведет к останову ге-

нератора. ТПСБ блокирует выработку потенциала СБМАШ, т. е. после холостого такта вновь начинают вырабатываться тактовые импульсы ТИ. ТПСБ сбрасывается в конце микропрограммы загрузки нового ССП контроля машины (из ячейки 70 ОП). До этого момента любой сбой, снова зафиксированный в РО, будет расцениваться как неисправность машины и приведет к установке в единичное состояние триггера тяжелого останова (ТТО).

Временная диаграмма обработки сбоев приведена на рис. 8.10.

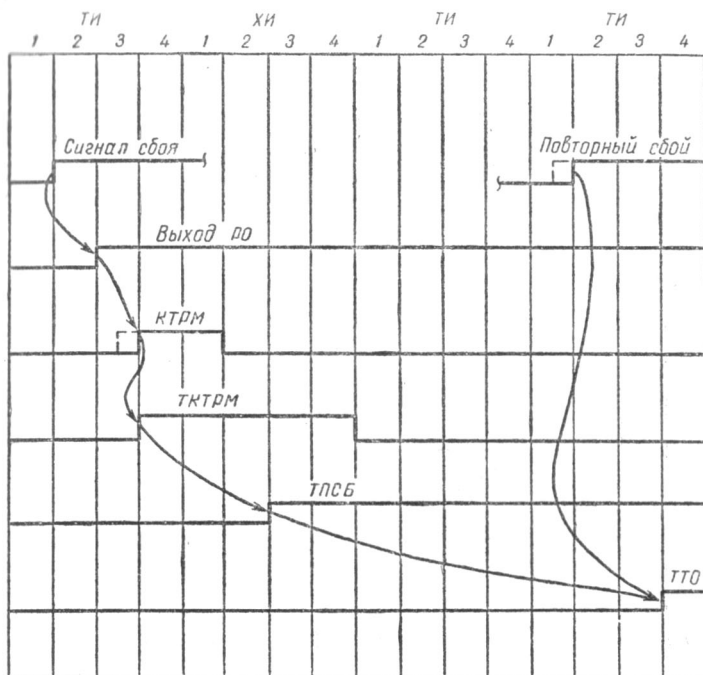


Рис. 8.10. Временная диаграмма обработки сбоев

ПУЛЬТ УПРАВЛЕНИЯ

9.1. Состав пульта управления

Пульт управления (ПУ) предназначен для индикации состояния, контроля и управления ЭВМ ЕС-1020. Блок-схема пульта приведена на рис. 9.1.

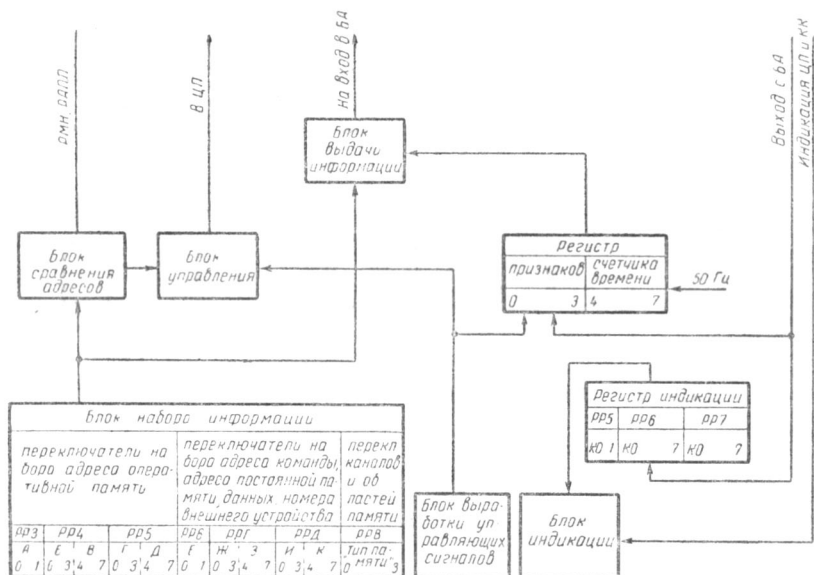


Рис. 9.1. Блок-схема пульта управления

Пульт управления состоит из блока набора информации, блока сравнения адресов, регистра признаков пультовых операций и счетчика времени, блока выработки управляющих сигналов, ре-

гистра индикации, блока индикации, блока выдачи информации и блока управления.

Блок набора информации содержит набор переключателей «Адрес памяти» (А, Б, В, Г, Д), набор переключателей «Адрес команды» (Е, Ж, З, И, К), а также переключатель «Тип памяти».

Блок сравнения адресов сравнивает адрес, установленный на переключателях «Адрес памяти», с содержимым адресного регистра постоянной (РАПП) или оперативной (РМН) памяти.

Регистр признаков пультовых операций и счетчика времени представляет собой восьмибитовый триггерный регистр (Р9), четыре бита которого (0—3) служат для запоминания признака типа операции, выполняемой с пульта, а остальные четыре бита (4—7) являются счетчиком сигналов временных интервалов, вырабатываемых с частотой сети (50 Гц) через 20 мс.

Блок выработки управляющих сигналов содержит кнопки и переключатели управления. При нажатии соответствующей кнопки управления устанавливается признак типа операции в регистре Р9 (0—3) и через блок управления выдается сигнал, который производит запуск процессора на выполнение указанной операции. Режим работы процессора задается переключателями управления.

Регистр индикации представляет собой 18-битовый триггерный регистр, в который может приниматься информация из оперативной (основной, мультиплексной и локальной) памяти, а также из регистров процессора и каналов, не имеющих постоянной индикации.

Блок индикации обеспечивает индикацию состояния регистров и отдельных управляющих триггеров, характеризующих работу процессора и каналов ввода-вывода. Элементы управления ламп индикации размещаются в соответствующих устройствах процессора и каналов.

Блок выдачи информации обеспечивает передачу информации из регистров пульта управления на входные шины БА.

Блок управления преобразует сигналы, поступающие с кнопок и переключателей пульта, в сигналы, необходимые для управления процессором.

9.2. Устройство пульта управления

Конструктивно пульт управления содержит панель управления, панель индикации и электронные схемы. На панели управления расположены переключатели и кнопки, на панели индикации — сигнальные лампочки. К электронным схемам относятся регистры и схемы управления. Размещение элементов на панелях пульта показано на рис. 9.2.

На панели управления расположены следующие элементы блока управляющих сигналов:

переключатели набора информации (табл. 9.1);

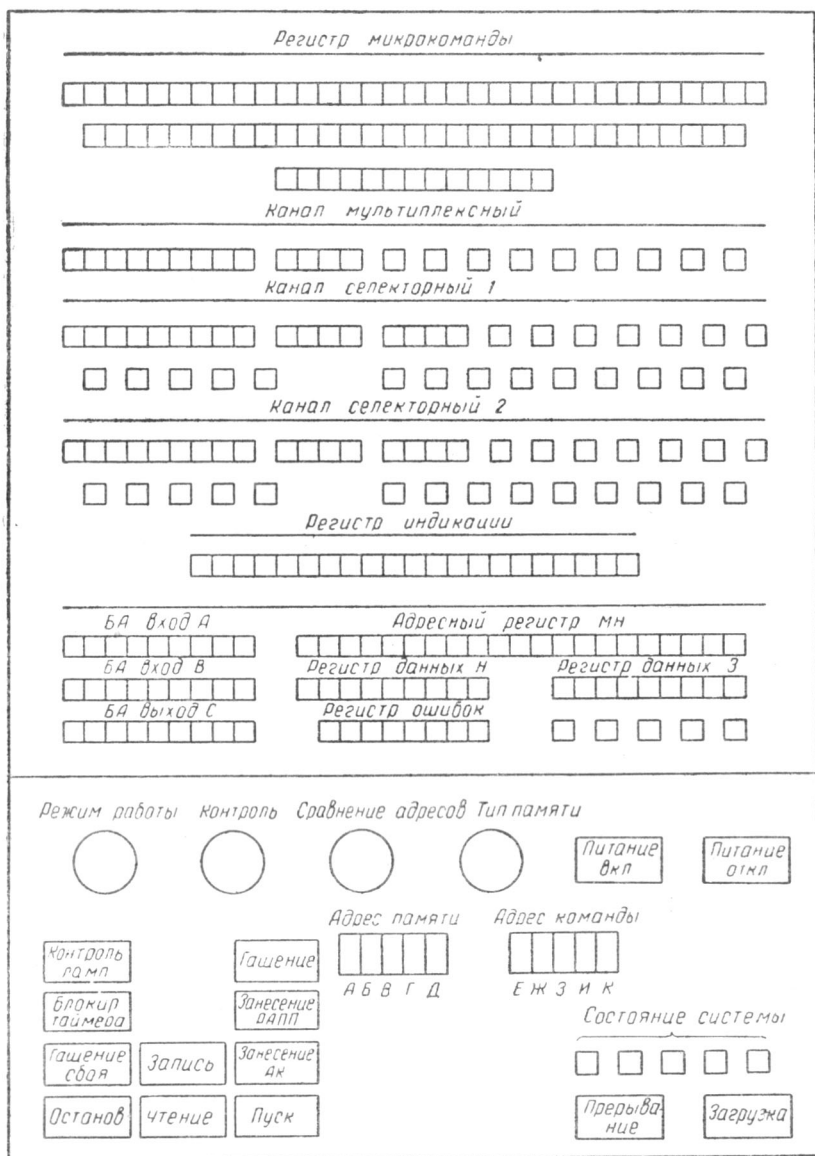


Рис. 9.2. Размещение элементов на панелях пульта управления

Название набора	Назначение
«Адрес памяти»	Задаёт адрес оперативной памяти, адрес для сравнения с адресом оперативной или постоянной памяти, номер регистра процессора или канала ввода-вывода Задаёт адрес текущей команды, адрес постоянной памяти, адрес внешнего устройства при вводе начальной программы
«Адрес команды»	

переключатели управления (табл. 9.2);

Таблица 9.2

Название переключателя	Назначение	Основные положения переключателя
«Тип памяти»	Указывает тип памяти, к которой производится обращение с пульта	ОП — к оперативной памяти, МП — к мультиплексной памяти, ЛП — к локальной памяти, ПЗ — к памяти ключей защиты, ЦП — к регистрам процессора, КМ — к регистрам мультиплексного канала, КС1 — к регистрам первого селекторного канала, КС2 — к регистрам второго селекторного канала
«Сравнение адресов»	Указывает, с адресом какой памяти будет производиться сравнение; определяет реакцию процессора на сравнение адресов	АВТОМАТ, ОСТАНОВ, ПЕРЕХОД
«Режим работы»	Определяет режим работы процессора	АВТОМАТ (основной режим работы процессора), КОМАНДА (выполняется одна команда при каждом нажатии кнопки «Пуск»), МИКРОКОМАНДА (выполняется одна микрокоманда при каждом нажатии кнопки «Пуск»), ОП (производится обращение к ряду ячеек памяти с пульта управления)
«Контроль»	Определяет реакцию процессора при обнаружении сбоя	АВТОМАТ, ОСТАНОВ, БЛОКИРОВКА

кнопки управления («Гашение», «Гашение сбоя», «Занесение АК», «Занесение РАПП», «Чтение», «Запись», «Прерывание», «Загрузка», «Пуск», «Останов», «Блокировка таймера», «Контроль ламп», «Питание вкл.», «Питание откл.»), назначение которых описано в разделе 9.3;

лампочки индикации состояния процессора (табл. 9.3).

Название	Указываемое состояние
«Загрузка» «Ожидание» «Контроль»	Выполняется программа первоначальной загрузки Процессор находится в состоянии ожидания Один из переключателей «Сравнение адресов», «Контроль» или «Режим работы» не находится в положении АВТОМАТ
«Ручная работа» «Система»	Процессор находится в состоянии останова или выполняется ручная операция Процессор выполняет любую микропрограмму

На панели индикации индицируется содержимое некоторых регистров и состояние отдельных управляющих триггеров:

а) постоянной памяти — содержимое регистра микрокоманды (РМК) и регистра адреса постоянной памяти (РАПП);

б) мультиплексного канала — состояние признаков интерфейса и контроля, а также содержимое регистра выходной информации мультиплексного канала (ШИН-К), через которые внешним устройствам передаются адреса устройств, коды команд и байты данных. Находящаяся на ШИН-К информация идентифицируется признаками интерфейса, индицируемыми на ПУ;

в) селекторных каналов — состояние выходных информационных шин селекторного канала (ШИН-К), содержимое регистра ключей защиты памяти канала, регистра команд, регистра состояния, регистра флажков и признаков интерфейса. На шинах канала ШИН-К индицируются как передаваемые внешним устройствам адреса устройств, коды команд, байты данных, так и принимаемые от них байты состояния и байты данных. Находящаяся на шинах канала ШИН-К информация идентифицируется с помощью признаков интерфейса, также индицируемых на ПУ;

г) регистра индикации, в который может быть выведено содержимое регистров процессора или каналов, не имеющих постоянной индикации, а также содержимое ячеек памяти;

д) процессора — адресного регистра оперативной памяти (РМН), информационных регистров оперативной памяти (РН и РЗ), входных регистров (РА и РВ) и выхода С БА, регистра ошибок (РО).

На панели индикации расположены четыре лампочки, характеризующие состояние некоторых управляющих триггеров процессора (табл. 9.4).

Микропрограммное управление пультом использует две микропрограммы: ПЗП — гашение системы и первоначальная загрузка программы и ОБСП — обслуживание пульта управления.

Для адресации в микропрограммах регистров пульта управления используются поля микрокоманд В и С, определяющие регистры, которые должны подключаться ко входу В и выходу С БА.

Название	Указываемое состояние
ТО ОСТАДР	Процессор находится в состоянии тяжелого останова Адрес, установленный на наборе «Адрес памяти», равен адресу постоянной или оперативной памяти (в зависимости от положения переключателя «Сравнение адресов»)
КС	Выполняется микропрограмма обслуживания селекторного канала (первого или второго)
КМ	Выполняется микропрограмма обслуживания мультиплексного канала

Если необходимо передать информацию из процессора в один из регистров индикации пульта управления, то в поле С микрокоманды указывается номер регистра, который должен быть подключен к выходу С БА. Предварительно с помощью поля УСТАНОВ в регистре РБС должен быть установлен признак пульта управления.

Все регистры пульта имеют условные обозначения, например, РРЗ, РР4, РР5, РРГ, РРД.

Если в поле С микрокоманды указано, что в один из регистров пульта управления (в РБС установлен признак пульта РБС (2)) должна быть принята информация с выхода С БА, то в дешифраторе поля С регистра микрокоманды вырабатывается соответствующий управляющий сигнал РР5: = С, РР6: = С и т. д.

Таблица 9.5

Название переключателей или регистра		Условное обозначение регистра	Управляющие сигналы	
			С	В
Набор «Адрес памяти»	А	РРЗ	—	РВ: = РРЗ
	Б и В	РР4	—	РВ: = РР4
	Г и Д	РР5	—	РВ: = РР5
Набор «Адрес команды»	Е	РР6	—	РВ: = РР6
	Ж и З	РРГ	—	РВ: = РРГ
	И и К	РРД	—	РВ: = РРД
«Тип памяти»		РРВ	—	РВ: = РРВ
Регистр признаков пультовых операций		РР9(0—3)	РР9: = С	РВ: = РР9
Регистр счетчика времени		РР9(4—7)	—	РВ: = РР9
Регистр индикации (РИНД)		РР5И РР6И РР7И	РР5: = С РР6: = С РР7: = С	— — —

Если в поле В микрокоманды указано, что содержимое одного из регистров пульта управления (РБС (2) = 1) должно быть передано на вход РВ БА, в дешифраторе поля В регистра микрокоманды вырабатывается соответствующий управляющий сигнал РВ: = РРЗ, РВ: = РР4 и т. д.

Перечень регистров пульта управления, их условные обозначения и сигналы, управляющие подключением этих регистров к входным и выходным шинам БА, приведены в табл. 9.5.

9.3. Выполнение пультовых операций

ПУ позволяет выполнять пультовые операции, перечень которых дан в табл. 9.6. При включении питания или выполнении микропрограммы гашения триггер останова устанавливается микропрограммно.

При любом обращении к процессору с пульта управления триггер останова в РБД сбрасывается и устанавливается соответствующим образом.

Таблица 9.6

Операция	Элемент ПУ, инициирующий операцию	Способ реализации операции
Гашение системы	Кнопка «Гашение»	Аппаратно-микропрограммный
Первоначальная загрузка программы (ПЗП)	Кнопка «Загрузка»	Микропрограммный
Занесение адреса команды в текущее ССП	Кнопка «Занесение АК»	Микропрограммный
Занесение адреса микрокоманды в РАПП	Кнопка «Занесение РАПП»	Аппаратный
Запись информации	Кнопка «Запись»	Микропрограммный
Чтение информации	Кнопка «Чтение»	Микропрограммный
Гашение сбоя	Кнопка «Гашение сбоя»	Аппаратный
Сравнение адресов	Переключатель «Сравнение адресов»	То же
Контроль работы процессора	Переключатель «Контроль»	»
Задание режима работы процессора	Переключатель «Режим работы»	»
Пуск процессора	Кнопка «Пуск»	»
Останов процессора	Кнопка «Останов»	»
Прерывание с ПУ	Кнопка «Прерывание»	»
Проверка ламп индикации	Кнопка «Контроль ламп»	»
Включение питания	Кнопка «Питание вкл.»	»
Отключение питания	Кнопка «Питание откл.»	»
Управление таймером	Кнопка «Блок таймера»	»

ющий триггер в регистре признаков (РР9(1—3)). После выполнения микропрограммы ОБСП производится останов.

При нажатии на кнопку «Пуск» процессор переходит к выполнению команды, адрес которой находится в РМФЕ.

В остановленном состоянии подсвечивается кнопка «Останов» и горит лампочка «Ручная работа».

ПРЕРЫВАНИЕ С ПУ осуществляется при нажатии кнопки «Прерывание», если процессор находится в автоматическом режиме и внешние прерывания не маскированы.

ПРОВЕРКА ЛАМП ИНДИКАЦИИ производится при нажатии кнопки «Контроль ламп», в результате чего все работоспособные лампочки индикации пульта подсвечиваются.

ВКЛЮЧЕНИЕ И ОТКЛЮЧЕНИЕ ПИТАНИЯ осуществляется нажатием кнопок «Питание вкл.» и «Питание откл.» соответственно. При включении питания автоматически запускается микропрограмма гашения системы.

ГАШЕНИЕ СИСТЕМЫ. Для выполнения этой операции следует установить все переключатели управления, кроме переключателя «Тип памяти», в положение АВТОМАТ, нажать кнопку «Гашение». В результате производится останов генератора тактовых импульсов и вырабатывается сигнал аппаратного гашения, по которому сбрасываются все регистры процессора, гасятся каналы и внешние устройства, аннулируются все запросы на прерывания, затем выполняется часть микропрограммы ПЗП (ГАШЕНИЕ), которая проверяет регистры процессора, корректирует контрольные биты ячеек памяти, стирает управляющую информацию каналов ввода-вывода, расположенную в локальной и мультиплексной памяти, и переводит систему в состояние останова.

После выполнения гашения системы на ПУ загорается лампочка «Ручная работа» и подсвечивается кнопка «Останов».

ПЕРВОНАЧАЛЬНАЯ ЗАГРУЗКА ПРОГРАММЫ (ПЗП) предназначена для ввода и запуска программ в том случае, когда содержимое оперативной памяти или слова состояния программы непригодно для дальнейшего использования.

Для выполнения этой операции необходимо выполнить следующие действия: на переключателях З, И, К набора «Адрес команды» установить номер канала и номер ВУ, с которого необходимо осуществить загрузку программы; установить все переключатели управления, кроме переключателя «Тип памяти», в положение АВТОМАТ; нажать кнопку «Загрузка». При этом осуществляется гашение системы и выполняется микропрограмма ПЗП, в результате чего в ОП с указанного на пульте управления ВУ вводится программа-загрузчик, осуществляющая дальнейший ввод программы в автоматическом режиме.

ЗАНЕСЕНИЕ АДРЕСА КОМАНДЫ В ТЕКУЩЕЕ ССП. Для выполнения этой операции необходимо выполнить следующие действия: перевести процессор в режим работы КОМАНДА; установить на переключателях набора «Адрес команды» требуемый адрес команды; установить переключатели «Сравнение адресов» и

«Контроль» в положение АВТОМАТ; нажать кнопку «Занесение АК». В результате этих действий будет выполнена часть микропрограммы ОБСП, осуществляющая передачу адреса команды, установленного на переключателях набора «Адрес команды», в текущее ССП и регистр индикации, после чего процессор переходит в состояние останова.

ЗАНЕСЕНИЕ АДРЕСА МИКРОКОМАНДЫ В РАПП. Для выполнения этой операции необходимо: установить переключатель «Режим работы» в положение МИКРОКОМАНДА; установить на переключателях набора «Адрес команды» требуемый адрес микрокоманды; нажать кнопку «Занесение РАПП». В результате адрес микрокоманды, установленный на переключателях «Адрес команды», передается в РАПП.

ЗАПИСЬ ИНФОРМАЦИИ. Эта операция позволяет вручную изменять содержимое ячеек памяти и регистров процессора. Для выполнения ее необходимо: перевести процессор в состояние останова; перевести переключатели «Режим работы» и «Сравнение адресов» в положение АВТОМАТ; установить адрес требуемой ячейки или номер регистра процессора на переключателях набора «Адрес памяти»; указать тип памяти на переключателе «Тип памяти» (см. табл. 9.2); задать записываемую информацию на переключателях набора «Адрес команды»; нажать кнопку «Запись». В результате этих действий будет выполнена микропрограмма обслуживания пульта ОБСП, которая запишет информацию в указанную ячейку памяти или регистр процессора.

Для записи информации в ряд последовательно расположенных ячеек памяти необходимо: передать начальный адрес памяти в РМФЕ (выполнить операцию занесения адреса команды); установить конечный адрес памяти на переключателях набора «Адрес памяти»; задать записываемую информацию на переключателях набора «Адрес команды»; установить переключатель «Сравнение адресов» в положение ОСТАНОВ или ПЕРЕХОД ОП, а переключатель «Режим работы» — в положение ОП; нажать кнопку «Запись».

В этом случае микропрограмма ОБСП запишет заданную на переключателях набора «Адрес команды» информацию в последовательные ячейки памяти (оперативной, локальной, мультиплексной, ключей защиты), начиная с адреса, переданного в РМФЕ. При записи информации в ячейку памяти, адрес которой установлен на переключателях «Адрес памяти», вырабатывается сигнал сравнения адресов, по которому процессор переходит в состояние останова, если переключатель «Сравнение адресов» находится в положении ОСТАНОВ, или операция записи повторяется, если этот переключатель установлен в положение ПЕРЕХОД ОП.

ЧТЕНИЕ ИНФОРМАЦИИ. Эта операция позволяет выводить в специальный регистр индикации РИНД содержимое любой ячейки памяти, регистров процессора и каналов, не имеющих постоянной индикации.

Для выполнения операции чтения производятся те же действия, что и при операции записи.

При нажатии кнопки «Чтение» запускается микропрограмма ОБСП, которая выполняет заданную операцию, затем процессор переводится в состояние останова.

В табл. 9.7—9.9 приведены положения переключателей, используемых при обмене информацией между пультом управления и ячейками различных типов памяти, а также между пультом управления и регистрами процессора и каналов.

ГАШЕНИЕ СБОЯ. Для выполнения этой операции необходимо нажать кнопку «Гашение сбоя». При этом осуществляются сброс триггеров регистра ошибок, индикация которого выведена на панель индикации, и сброс триггера тяжелого останова, после чего дальнейшая работа процессора может быть продолжена.

СРАВНЕНИЕ АДРЕСОВ. Переключатель «Сравнение адресов» позволяет сравнивать адрес, установленный на переключателях «Адрес памяти», с адресом постоянной или оперативной памяти. В зависимости от положения этого переключателя определяется реакция процессора на сравнение адресов.

Сравнение с адресом оперативной памяти. При этом сравнении возможны следующие режимы: останов ОП; останов по записи ОП; переход ОП.

Останов ОП. При сравнении адреса оперативной памяти с адресом, установленным на переключателях А—Д, формируется сигнал равенства адресов, по которому устанавливается в единичное состояние триггер останова в РБД, и по окончании выполнения текущей команды процессор переходит в состояние останова. При останове подсвечивается кнопка «Останов», горят лампочки «Ост. адр.» и «Ручная работа». Автоматическая работа процессора продолжается при нажатии кнопки «Пуск».

Останов по записи ОП. Останов центрального процессора происходит при сравнении адресов выполняемой микрокоманды, осуществляющей запись в ОП. В этом случае производится останов генератора тактовых импульсов, загорается лампочка «Ост. адр.», в информационных регистрах памяти РН, РЗ индицируется записываемая информация, а в адресном регистре РМН — ее адрес. Процессор продолжает работу при нажатии кнопки «Пуск».

Переход ОП. При сравнении адресов производится останов процессора. При работе процессора в автоматическом режиме по сигналу ПЕРЕХОД ОП выполняется занесение адреса команды текущего ССП, т. е. в РМФЕ передается адрес, установленный на наборе переключателей Е—К, и работа процессора продолжается, начиная с этого адреса. Если переключатель «Режим работы» установлен в положение КОМАНДА или ОП, то происходит закливание операции, выполняемой с пульта управления.

Сравнение с адресом постоянной памяти. При этом сравнении возможны следующие режимы: останов ПП; переход ПП; синхронный импульс ПП.

Ручная операция	Положение переключателя													Исполнительная кнопка	Место индикации	Примечание			
	"Time Memory"	"Сравнение адресов"	"Режим работы"	"Адрес памяти"				"Адрес команды"					Исполнительная кнопка						
				А	Б	В	Г	Д	Е	Ж	З	И					К		
Запись по определенному адресу	ОП	АВТОМАТ	АВТОМАТ	0-30-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	0-F	«Запись»	Адрес—РМН, данные—РИНД (0—7)			
	ЛП	АВТОМАТ	АВТОМАТ	—	—	—	0-F	0-F	—	—	—	—	0-F	0-F					
	МП	КОМАНДА	КОМАНДА	—	—	1-7	0-F	0-F	—	—	—	—	0-F	0-F					
	ПЗ			0-30-F	0,8	—	—	—	—	—	—	—	0-F	0-F					
Считывание по определенному адресу	ОП	АВТОМАТ	АВТОМАТ	0-30-F	0-F	0-F	0-F	0-F	—	—	—	—	—	—	«Чтение»				
	ЛП	АВТОМАТ	АВТОМАТ	—	—	—	0-F	0-F	—	—	—	—	—	—					
	МП	КОМАНДА	КОМАНДА	—	—	1-7	0-F	0-F	—	—	—	—	—	—					
	ПЗ			0-30-F	0,8	—	—	—	—	—	—	—	—	—					
Запись в ряд последовательно расположенных адресов	ОП	АВТОМАТ	АВТОМАТ	—	—	—	—	—	—	0-30-F	0-F	0-F	0-F	0-F	«Занесение АК»	РИНД (0—17)	Установка начального адреса		
	ЛП	АВТОМАТ	АВТОМАТ	—	—	—	—	—	—	—	—	—	0-F	0-F					
	МП	КОМАНДА	КОМАНДА	—	—	—	—	—	—	—	—	1-7	0-F	0-F					
	ПЗ			—	—	—	—	—	—	0-30-F	0,8	—	—	—					
	ОП	АВТОМАТ	ОП	0-30-F	0-F	0-F	0-F	0-F	—	—	—	—	0-F	0-F					

Ручная операция	Положение переключателей											Исполнительная кнопка	Место индикации	Примечание	
	ВНИМАНИЕ	Сравнение адресов*	Режим работы*	*Адрес памяти*											*Адрес команды*
				А	Б	В	Г	Д	Е	Ж	З				
	ЛП				—	—	0-F	0-F	—	—	—	—	0-F	0-F	Установка конечного адреса и запись —РМН, —данные— РИНД (0—7) и РН,РЗ
	МП				—	—	1—7	0-F	0-F	—	—	—	0-F	0-F	
	ПЗ				0-3	0-F	0,8	—	—	—	—	—	0-F	0,8	
Считывание из ряда последовательно расположенных адресов	ОП	АВТОМАТ	АВТОМАТ		—	—	—	—	—	0-3	0-F	0-F	0-F	0-F	Установка начального адреса РИНД (0—17)
	ЛП				—	—	—	—	—	—	—	—	0-F	0-F	
	МП				—	—	—	—	—	—	—	—	1-7	0-F	
ПЗ				—	—	—	—	—	—	0-3	0-F	0,8	—	—	
Считывание конечного адреса и считывание	ОП	АВТОМАТ	ОП		0-3	0-F	0-F	0-F	—	—	—	—	—	—	«Чтение» Адрес конечный— РМН, —данные— РИНД (0—17) и РН,РЗ
	ЛП				—	—	0-F	0-E	—	—	—	—	—	—	
	МП				—	—	1-7	0-F	0-F	—	—	—	—	—	
ПЗ				0-3	0-F	0,8	—	—	—	—	—	—	—	—	

Регистр процессора	Выполняемая операция	Положение переключателей											Исполнительная кнопка	Место индикации				
		Тип памяти*	Сравнение адресов*	Режим работы*	Адрес памяти				Адрес команды									
					А	Б	В	Г	Д	Е	Ж	З			И	К		
Адресный РМФЕ	Считывание	ЦП	АВТОМАТ	АВТОМАТ						0							«Чтение»	РИНД (0-17)
	Запись			КОМАНДА													«Запись»	
РГРИ	Считывание	То же	То же	То же						1							«Чтение»	То же
	Запись	»	»	»													«Запись»	»
РПТУ	Считывание	»	»	»						2							«Чтение»	»
	Запись	»	»	»													«Запись»	»
Универсального назначения	Считывание																	
	Запись																	
РЛ	Считывание	»	»	»						3							«Чтение»	РИНД (0-7)
	Запись	»	»	»													«Запись»	
РД	Считывание	»	»	»						4							«Чтение»	То же
	Запись	»	»	»													«Запись»	»

Регистр процессора	Выполняемая операция	Положение переключателей										Исполнительная кнопка	Место назначения					
		Тип памяти*	Сравнение адресов*	Режим работы*	Адрес памяти*									Адрес команды*				
					А	Б	В	Г	Д	Е	Ж				З	И	К	
Служебный РБК	Считывание	ЦП	АВТОМАТ	АВТОМАТ													Чтение	РИНД (0-7)
	Запись			КОМАНДА					5								0-F0-F	
РБД	Считывание	То же	То же	То же														То же
	Запись	»	»	»					6								0-F0-F	»
РБС	Считывание	»	»	»														»
	Запись	»	»	»					7								0-F0-F	»
РБР	Считывание	»	»	»														»
	Запись	»	»	»					8								0-F0-F	»
Информационный РБЭ	Считывание	—	—	—														»
	Запись								9								0-F 0,8	»

Таблица 9.9

Регистр канала	Канал	Положение переключателей						Исполнительная кнопка	Место индикации	
		Тип памяти*	Сравнение адресов*	Режим работы*	Адрес памяти*					
					А	Б	В	Г	Д	
Регистр входной информации РЗ	Мультиплексный Селекторный 1 Селекторный 2	КМ КС1 КС2	АВТОМАТ	АВТОМАТ КОМАНДА	—	—	—	—	0	„Чтение“ РИНД (0—7)
Регистр управления Р4	Мультиплексный Селекторный 1 Селекторный 2	КМ КС1 КС2	То же	То же	—	—	—	—	1	То же
Регистр ошибок РБ	Мультиплексный Селекторный 1 Селекторный 2	КМ КС1 КС2	»	»	—	—	—	—	2	»
Регистр счетчика байтов Р5, Р6	Селекторный 1 Селекторный 2	КС1 КС2	»	»	—	—	—	—	3	РИНД (0—15)
Регистр состояния РВ	Селекторный 1 Селекторный 2	КС1 КС2	АВТОМАТ	АВТОМАТ КОМАНДА	—	—	—	—	4	„Чтение“ РИНД (0—7)
Регистр канальных признаков КП	Селекторный 1 Селекторный 2	КС1 КС2	То же	То же	—	—	—	—	5	То же
Регистр адреса РЯ	Селекторный 1 Селекторный 2	КС1 КС2	»	»	—	—	—	—	6	» РМН

Останов ПП. При равенстве адреса, установленного на переключателях набора «Адрес памяти» (Б, В, Г, Д), и содержимого адресного регистра постоянной памяти вырабатывается сигнал равенства адресов, по которому отключается генератор тактовых импульсов. При этом на ПУ индицируются микрокоманда, которая должна быть выполнена при продолжении работы процессора в любом режиме, и ее адрес. Процессор продолжает работу при нажатии кнопки «Пуск».

Переход ПП. При совпадении адреса постоянной памяти и адреса, заданного переключателями Б — Д, адрес, установленный на переключателях Ж — К, передается в адресный регистр постоянной памяти и продолжается работа по данному адресу.

Синхроимпульс ПП. При совпадении адреса, установленного на наборе Б, В, Г, Д, с адресом постоянной памяти вырабатывается синхроимпульс. Режим выработки синхроимпульса на работу процессора не влияет и используется при наладке и проверке работы процессора.

КОНТРОЛЬ РАБОТЫ ПРОЦЕССОРА. На пульте управления имеется переключатель «Контроль», который задает реакцию процессора на сбой в зависимости от его положения: АВТОМАТ, ОСТАНОВ, БЛОКИРОВКА.

При возникновении сбоя в процессоре происходит установка в единичное состояние триггера первого сбоя в РБД, и процессор переходит к обработке сбоя, если переключатель «Контроль» находится в положении АВТОМАТ.

Если переключатель «Контроль» находится в положении ОСТАНОВ, то машинные сбои не обрабатываются, а производится отключение генератора тактовых импульсов с индикацией ошибки в регистре ошибок процессора или на индикаторах сбоя каналов.

Если переключатель «Контроль» установлен в положение БЛОКИРОВКА, процессор продолжает работать, не реагируя на сбой.

ПУСК-ОСТАНОВ ПРОЦЕССОРА. Для запуска процесса выполнения операций тем способом, который задан переключателем «Режим работы» (положения АВТОМАТ, КОМАНДА, МИКРОКОМАНДА), служит кнопка «Пуск».

Переход процессора в состояние останова возможен после включения питания, гашения системы, останова по адресу, окончания выполнения команды, если переключатель «Режим работы» установлен в положение КОМАНДА, и после нажатия кнопки «Останов».

Процессор переходит в состояние останова, если выполнение текущей команды закончено и запросов на прерывание нет. При останове процессора в счетчике адреса команд (РМФЕ) и регистре индикации находится адрес следующей команды, начиная с которой при нажатии кнопки «Пуск» будет продолжена дальнейшая обработка.

Процессор также останавливается, если он не способен закончить выполнение текущей команды или обслуживающей микропрограммы из-за машинного сбоя.

Останов производится следующим образом: по окончании выполнения каждой команды осуществляется анализ триггера останова. При единичном его состоянии происходит заикливание микрокоманды, анализирующей триггер останова. Единичное состояние этого триггера устанавливается при работе процессора в режиме КОМАНДА, сравнении адреса, установленного на наборе «Адрес памяти», с адресом оперативной памяти или нажатии кнопки «Останов».

УПРАВЛЕНИЕ ТАЙМЕРОМ. Таймер (счетчик временных интервалов) представляет собой одно полноразрядное слово, расположенное в оперативной памяти по адресам 50—53/16.

Интервальный таймер может регистрировать время выполнения программы и вырабатывать сигналы для управления ходом выполнения программы по истечении заданного промежутка времени.

Управление таймером производится микропрограммно. Таймер рассматривается как целое число и обрабатывается по правилам арифметики с фиксированной запятой. Полный цикл таймера составляет 15,5 ч.

Для подсчета временных интервалов служит четырехбитовый двоичный счетчик P9 (4—7), значение которого увеличивается на единицу с частотой сети 50 Гц с возможностью запоминания до 15 интервалов счета.

Если содержимое счетчика равно 15 и вырабатывает еще один запрос, счетчик сбрасывается в нулевое положение. При неравенстве содержимого счетчика нулю устанавливается запрос на обновление таймера.

В конце выполнения каждой команды производится обслуживание всех запросов на прерывание по внешним причинам или по вводу-выводу, если они имеются, а также запроса на обновление таймера. Так как таймер имеет самый низкий приоритет, то запрос на его обновление не всегда может быть удовлетворен в течение 20 мс, поэтому счетчик может накапливать до 15 интервалов времени по 20 мс ($T_{сч} = 15 \times 20 \text{ мс} = 300 \text{ мс}$). Если значение счетчика интервалов времени не будет учтено в течение 300 мс, то для интервального таймера оно будет потеряно.

Независимо от содержимого счетчика таймер обновляется при переходе процессора в состояние ожидания или останова. В остановленном состоянии процессора таймер не обновляется.

Обновление таймера заключается в считывании содержимого таймера из оперативной памяти (ячеек 50—53) и вычитания из него содержимого счетчика (P9 (4—7)), умноженного на шесть. Результат вычитания записывается в ячейки таймера, и процессор переходит к выполнению прерванной программы. Если при вычитании значение таймера стало отрицательным, то устанавливается запрос на внешнее прерывание по таймеру.

Интервальный таймер может быть заблокирован, для этого служит кнопка «Блок. таймера» на пульте управления. Если эта кнопка подсвечивается, таймер заблокирован.

Микропрограмма ОБСП вначале передает содержимое Р9 (4—7), умноженное на шесть, в регистр процессора РД и сбрасывает Р9 (4—7) в нулевое положение. Из оперативной памяти считывается младшее полуслово таймера и из него вычитается содержимое регистра РД. Затем модифицированное значение младшего полуслова таймера записывается в ячейки 52—53 оперативной памяти и контролируется возможность образования переноса при вычитании (признака «занять» из старшего полуслова таймера).

Если при вычитании из младшего полуслова таймера переноса не возникло, процессор переходит к выполнению прерванной программы. Если перенос был, то модифицируется старшее полуслово таймера путем повторного входа в программу изменения таймера и вычитания нуля из содержимого ячеек 50—51 оперативной памяти. После этого значение таймера записывается в ячейки 50—51, и производится анализ на образование переноса. Если при модификации старшего полуслова таймера переноса не возникло, то происходит переход, как и в предыдущем случае, к прерванной программе.

Перенос после повторного вычитания означает, что значение таймера изменилось от положительного к отрицательному. В этом случае микропрограмма устанавливает признак внешнего прерывания по таймеру и управление передается соответствующей программе.

Глава 10

МУЛЬТИСИСТЕМНАЯ РАБОТА

В ЭВМ ЕС-1020 для организации системы из двух машин, входящих в состав Единой системы, предусмотрены следующие средства: две команды прямого управления (ПРЯМОЕ ЧТЕНИЕ, ПРЯМАЯ ЗАПИСЬ); разделенные устройства управления; адаптер канал — канал.

10.1. Команды прямого управления

Команды прямого управления являются средством связи непосредственно между двумя процессорами, осуществляемой с помощью внешних прерываний и команд ПРЯМОЕ ЧТЕНИЕ и ПРЯМАЯ ЗАПИСЬ. Эти команды предназначены в основном для передачи управляющей информации.

Связь по линиям интерфейса прямого управления осуществляется в обоих направлениях по функционально разделенным линиям интерфейса.

В процессоре ЕС-2020 имеется восемь шин для передачи информации Вых-Ш0 — Вых-Ш7, восемь шин для приема информации Вх-Ш0 — Вх-Ш7, восемь выходных шин синхросигналов для передачи внешнего прерывания ЛС-0 — ЛС-7, восемь входных шин синхросигналов для приема внешних прерываний другого процессора ВС-0 — ВС-7 и четыре шины управления Вых-ЛЧТ, Вх-ЛЧТ, Вх-ЛЗП, Вых-ЛЗП.

Выходная шина сигнала записи Вых-ЛЗП одного процессора соединяется с входной шиной сигнала записи Вх-ЛЗП другого процессора. Сигнал записи не позволяет процессору считывать данные с входных шин Вх-Ш0 — Вх-Ш7 до тех пор, пока эти данные не станут действительны или не будут заменены новыми.

Назначение выходной шины сигнала чтения (Вых-ЛЧТ) — передать сигнал о выполнении команды ПРЯМОЕ ЧТЕНИЕ. Эта шина соединяется с входной шиной чтения (Вх-ЛЧТ) другого процессора.

На рис. 10.1 показано соединение входных и выходных шин двух процессоров.

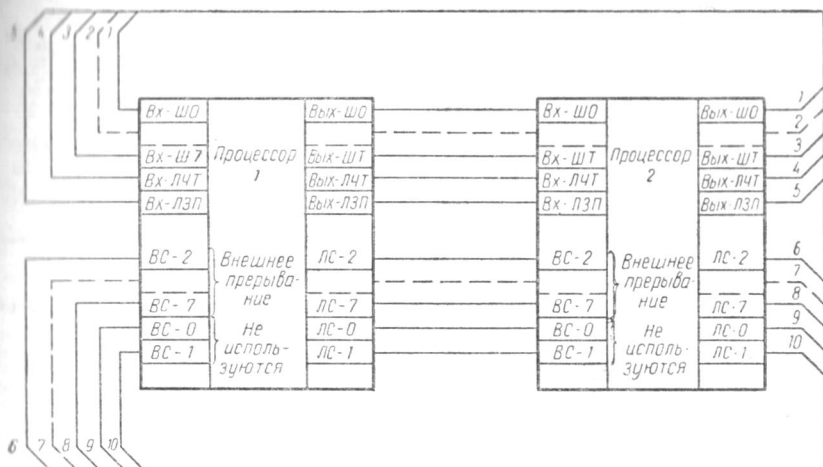


Рис. 10.1. Схема соединения входных и выходных шин двух процессоров

Команда **ПРЯМОЕ ЧТЕНИЕ** имеет формат *RR*. Структура ее приведена на рис. 10.2. Команда осуществляет запись данных, поступающих из другого процессора на шины ВХ-Ш0—ВХ-Ш7, в ячейку памяти, адрес которой определен в команде.

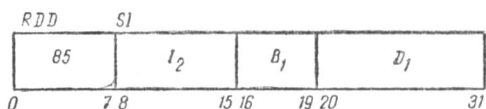


Рис. 10.2. Структура команды ПРЯМОЕ ЧТЕНИЕ

Данные на ВХ-Ш0—ВХ-Ш7 не могут считаться действительными на время действия сигнала на входной шине сигнала записи. Биты контроля на четность с данными не передаются. При записи данных в память процессор вырабатывает бит контроля на четность. До того как данные будут приняты, команда **ПРЯМОЕ ЧТЕНИЕ** передает содержимое поля I_2 команды (биты 8—15) на выходные шины синхросигналов (ЛС-0—ЛС-7), являющихся сигналами запроса внешнего прерывания в процессоре, передавшем байт данных. Одновременно с синхросигналами подается девятый сигнал на выходную шину чтения (ВЫХ-ЛЧТ).

Команда **ПРЯМАЯ ЗАПИСЬ** имеет формат *RR*. Структура команды приведена на рис. 10.3.

Команда выбирает байт данных из ячейки памяти, адрес которой определен в команде, и передает его на выходные шины передачи информации (ВЫХ-Ш0—ВЫХ-Ш7) в виде потенциаль-

ного сигнала. Повторением команды эти сигналы могут изменяться через различные интервалы времени.

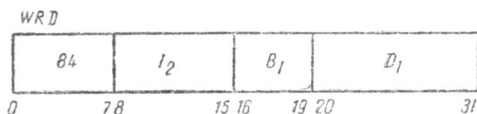


Рис. 10.3. Структура команды ПРЯМАЯ ЗАПИСЬ

Команда ПРЯМАЯ ЗАПИСЬ вызывает передачу содержимого поля I_2 команды (биты 8—15) на выходные шины синхросигналов (ЛС-0 — ЛС-7). Одновременно на выходную шину сигнала записи подается девятый сигнал. Этот сигнал во времени перекрывает изменение сигналов на выходных шинах передачи информации.

Синхросигналы на шинах ЛС-0 — ЛС-7 используются для организации внешних прерываний в процессоре, принимающем информацию.

10.2. Разделенные устройства управления ВУ

Устройства управления ВУ, доступные для нескольких каналов одного или разных процессоров, называются разделенными. Такие устройства управления могут быть использованы для связи накопителей на магнитных лентах и магнитных дисках, разделенных между двумя процессорами, работающими в мультисистеме.

При подключении разделенного УВУ с накопителями на магнитных лентах к двум каналам УВУ с каждым из них работает последовательно. Во время работы с одним каналом УВУ недоступно для другого канала.

Для взаимосвязанных переключений каналов предусмотрены команды РЕЗЕРВИРОВАНИЕ И РАЗЪЕДИНЕНИЕ, относящиеся к основным командам и являющиеся модификациями команды УТОЧНИТЬ СОСТОЯНИЕ.

10.3 Адаптер канал—канал

В ЕС ЭВМ имеется специальное устройство — адаптер канал — канал (АКК), позволяющее выполнять и синхронизировать операции между двумя каналами.

АКК может быть использован как в мультипроцессорных, так и в однопроцессорных системах. В мультисистемных (многопроцессорных системах) он позволяет устанавливать быструю связь между каналами двух машин ЕС ЭВМ, в односистемных (однопроцессорных системах) — с помощью каналов этой системы перемещать блоки данных из одной области в другую область ОП.

АКК работает в монопольном режиме и передает данные со скоростью, меньшей, чем у связанного с ним скоростного канала.

Он может связывать мультиплексные и (или) селекторные каналы. В некоторых моделях ЕС ЭВМ, используемых в однопроцессорных системах, АКК предназначен для связи только селекторных каналов.

ЛИТЕРАТУРА

1. Вычислительная система IBM/360. Принципы работы. Пер. с англ. под ред. Штаркмана В. С. М., «Советское радио», 1969.
2. Джермейн К. Программирование на IBM/360. М., «Мир», 1971.
3. Дроздов Е. А., Пятибратов А. П. Основы построения и функционирования вычислительных систем. М., «Энергия», 1973.
4. Единая система ЭВМ. Под общей редакцией А. М. Ларионова. М., «Статистика», 1974.
5. Каган Б. М., Каневский Н. М. Цифровые вычислительные машины и системы. М., «Энергия», 1973.
6. Карцев М. А. Арифметические устройства. М., Физматгиз, 1970.
7. Майоров С. А., Новиков Г. И. Структура цифровых вычислительных машин. М., «Машиностроение», 1970.
8. Мультипроцессорные вычислительные системы. Под ред. Хетагурова Я. А. М., «Энергия», 1971.
9. Путинцев Н. Д. Аппаратный контроль управляющих цифровых вычислительных машин. М., «Советское радио», 1966.
10. Селлерс Ф. Методы обнаружения ошибок в работе ЭЦВМ, М., «Мир», 1972.
11. Флорес А. Организация вычислительных машин. М., «Мир», 1972.
12. Электронная вычислительная машина ЕС-1020. Под общ. ред. А. М. Ларионова. М., «Статистика», 1975.

ОГЛАВЛЕНИЕ

Глава 1. Основные сведения о процессоре ЭВМ ЕС-1020	
1.1. Назначение и состав процессора	3
1.2. Выполнение команды	6
1.3. Выполнение программы	10
Глава 2. Блок синхронизации	
2.1. Назначение и состав блока	11
2.2. Узел формирования основных и рабочих тактовых импульсов	12
2.3. Узел распределения синхронимпульсов	15
2.4. Общая временная диаграмма работы процессора	16
Глава 3. Блок микропрограммного управления	
3.1. Состав блока	17
3.2. Структура микрокоманды	18
3.3. Дешифраторы полей микрокоманды	26
3.4. Триггеры состояний	27
3.5. Способы организации адреса микрокоманды	28
3.6. Разработка микропрограмм	33
Глава 4. Блок регистров	
4.1. Назначение и состав блока	36
4.2. Адресные регистры	37
4.3. Регистры общего пользования	40
4.4. Формирование адреса оперативной памяти	41
4.5. Служебные регистры	42
Глава 5. Оперативная память и блок защиты основной памяти	
5.1. Структура оперативной памяти	51
5.2. Блок связи с оперативной памятью	53
5.3. Назначение и принцип работы блока защиты памяти	57
5.4. Состав и режимы работы блока защиты	59
Глава 6. Арифметическо-логический блок	
6.1. Назначение и принцип работы	62
6.2. Структура блока	67
6.3. Описание узлов блока	71
6.4. Выполнение операций в блоке	85
6.5. Обработка чисел в блоке	93
Глава 7. Система прерываний	
7.1. Понятие о супервизоре и структура слова состояния программы	97
7.2. Действия при прерывании и возврат к прерванной программе	99
7.3. Приоритет прерываний	104
7.4. Виды прерываний	106
7.5. Запрещение прерываний посредством маскирования	117
7.6. Состояния процессора	118
7.7. Размещение текущего ССП	120
7.8. Микропрограммы смены ССП	121
Глава 8. Система контроля	
8.1. Методы контроля	126
8.2. Формирование контрольных бит и выявление ошибок	126
8.3. Распределение контрольных точек в процессоре	128
8.4. Обработка машинных сбоев	133
8.5. Регистр ошибок	134
8.6. Временная диаграмма обработки сбоя машины	135
Глава 9. Пульт управления	
9.1. Состав пульта управления	138
9.2. Устройство пульта управления	139
9.3. Выполнение пультовых операций	144
Глава 10. Мультисистемная работа	
10.1. Команды прямого управления	156
10.2. Разделенные устройства управления ВУ	158
10.3. Адаптер канал — канал	158
Литература	159