

ЦИФРОВАЯ ЭЛЕКТРОННАЯ
ВЫЧИСЛИТЕЛЬНАЯ МАШИНА
ЕС-1030

Курс лекций

Выпуск 3

ПРОЦЕССОР ЭВМ ЕС-1030

Цифровая электронная вычислительная машина ЕС-1030.
Курс лекций. Выпуск 3. Процессор ЭВМ ЕС-1030.

Курс лекций содержит описание общих принципов построения Единой системы электронных вычислительных машин и основных устройств ЭВМ ЕС-1030. Курс состоит из пяти выпусков. Материал изложен в соответствии с программой дисциплины «Техника и операционные системы ЕС ЭВМ». В основу курса лекций положено техническое описание ЭВМ ЕС-1030». При изучении курса необходимо пользоваться альбомом схем «Цифровая ЭВМ ЕС-1030».

Настоящий выпуск содержит главу 6, в которой рассмотрен процессор ЭВМ ЕС-1030. Параграфы 1—3, 5—9 написаны В. Н. Карповым, параграфы 4, 10 — А. В. Серостановым. Таблиц 9.



Глава 6

ПРОЦЕССОР ЭВМ ЕС-1030

6.1. Назначение и технические характеристики процессора

Процессор — это центральная часть ЭВМ ЕС-1030, которая организует работу машины в целом.

Основные функции процессора:

1. Выполнение стандартного набора команд ЕС ЭВМ, а также команд десятичной арифметики и команд с плавающей запятой.
2. Переключение программных состояний машины.
3. Организация обращений к оперативной памяти (ОП) для записи или считывания информации.
4. Приоритетное обслуживание запросов на прерывания.
5. Выдача управляющей информации каналам для управления обменом информацией ОП с ВУ.
6. Управление работой машинных часов (таймера), используемых для подсчета времени выполнения отдельных программ, времени нахождения машины в состоянии «Счет» и для других целей.

7. Автоматический контроль информации при ее передачах.

8. Фиксация факта отказа в отдельных устройствах машины.

В процессоре имеются средства для обеспечения работы ЭВМ ЕС-1030 в составе вычислительного комплекса, а также предусмотрена возможность установки дополнительного оборудования, обеспечивающего работу машины в режиме динамического распределения памяти.

Управление процессором аппаратно-микропрограммное.

При решении задач на ЭВМ ЕС-1030 вычисления или другая обработка информации выполняются в процессоре по программам, предварительно введенным в оперативную память машины.

Технические данные процессора ЭВМ ЕС-1030

1. Диапазон представления обрабатываемых чисел — от 10^{-78} до 10^{75} .
2. Время выполнения основных арифметических операций (табл. 6.1.1.).

3. Число уровней приоритетов прерываний — 5.
 4. Количество одновременно выполняемых программ — до 15.
 5. Емкость местной оперативной памяти — 64 36-разрядных слова.

6. Емкость долговременного запоминающего устройства — 4096 72-разрядных слов.

7. Емкость памяти ключей защиты — 256 9-разрядных слов.

8. Потребляемая мощность — не более 2,5 кВА.

Конструктивно процессор в машине ЕС-1030 выполнен в виде отдельной стойки, которая имеет шифр ЕС-2030.

Т а б л и ц а 6.1.1

Операции	С фиксированной запятой, мкс	С плавающей запятой, мкс	Десятичная арифметика, мкс
Сложение	7,42—11,47	11,47—23,62	39,47—163,59
Умножение	36—41	39	50,17—198,73
Деление	98,5—102,57	50,17—198,73	201,12

6.2. Структурная схема и краткая характеристика основных блоков процессора

В состав процессора входят следующие блоки и устройства (рис. 6.2.1).

1. БАЛ — блок арифметический логический.

2. Устройство микропрограммного управления (УМУ), состоящее из двух элементов:

БМУ — блока микропрограммного управления,

ДЗУ — долговременного запоминающего устройства.

3. БУР — блок управляющих регистров.

4. БООП — блок обращения к оперативной памяти.

5. МП — местная память.

6. БСИ — блок синхронизации.

7. ПКЗ — память ключей защиты.

Для целей контроля и диагностики состояния машины в составе процессора имеются: блок контроля (БК) и блок диагностики (БД).

Для обеспечения работы машины в составе многомашинного вычислительного комплекса процессор имеет блок прямого управления — БПРУ.

Расположение блоков и устройств процессора в рамках А, В, С стойки ЕС-2030 приведено на рис. 6.2.2.

Рассмотрим состав и назначение отдельных блоков процессора, а также информационные связи и связи управления, используемые при выполнении основных команд.

БАЛ — блок арифметический логический. Предназначен для выполнения всех арифметических и логических операций.

Результаты операций записываются, как правило, в оперативную или в местную память по адресу первого операнда команды.

В состав БАЛ входят регистры Р1—Р8, счетчики РС1—РС6, а также сумматоры, дешифраторы и одиночные триггеры.

Обмен информацией с ОП осуществляется через регистр коммутатора процессора (РКП).

Выходным регистром блока БАЛ при записи информации в оперативную память является регистр Р3. Прием информации из оперативной памяти осуществляется на регистр Р5.

С местной памятью связь осуществляется только через регистр Р5.

На регистр Р5 поступает также адрес внешнего устройства (ВУ) с переключателей пульта управления. Этот адрес из блока БАЛ передается в стойку каналов для указания ВУ, используемого для ввода программы первоначальной загрузки. Кроме того, в стойку каналов адреса ВУ передаются при выполнении команд ввода — вывода.

Управляющие сигналы в блок БАЛ поступают из блока микропрограммного управления. Устройство микропрограммного управления предназначено для реализации микропрограммного принципа управления работой процессора.

Долговременное запоминающее устройство (ДЗУ), входящее в состав УМУ, предназначено для хранения микропрограмм, по которым организуются все вычисления в процессоре. Долговременное запоминающее устройство состоит из регистра адреса РАДЗУ, регистра информации РИДЗУ и накопителя, в котором хранятся микропрограммы всех машинных команд и служебные микропрограммы.

Каждая микропрограмма состоит из набора микрокоманд. Одна микрокоманда представляет собой 72-разрядное слово, записанное по определенному адресу блока ДЗУ. Служебные микропрограммы предназначены для выполнения таких действий, как «сброс» системы в исходное состояние, обработка прерываний, смена ССП и т. п.

Блок микропрограммного управления (БМУ) организует считывание микрокоманд из накопителя ДЗУ с частотой 1,5 мГц и выдает управляющую информацию в другие блоки.

Считанная из накопителя ДЗУ микрокоманда сначала помещается в регистр информации РИДЗУ, а затем передается в дополнительный регистр РИДЗУД, находящийся в блоке БМУ. Группы разрядов (поля) микрокоманды (рис. 6.2.3), находящейся в регистрах РИДЗУ и РИДЗУД, дешифрируются, в результате чего образуются исполнительные сигналы для выполнения элемен-

тарных операций. Элементарная операция — это, например, за-сылка информации в какой-либо регистр, установка триггера в то или иное состояние, анализ его состояния и т. п.

Дешифраторы элементарных операций для различных полей микрокоманды располагаются в тех блоках, где выполняются данные операции. Это блоки БМУ, БАЛ, БУР и БД.

Адрес следующей микрокоманды указывается в предыдущей микрокоманде или формируется в блоке БМУ в результате анализа задаваемых условий.

БУР — блок управляющих регистров.

В блок БУР входят четыре основных управляющих регистра:

1. РССП — регистр слова состояния программы.
2. РВО — регистр выполняемой операции.
3. РФП — регистр фиксации прерываний.
4. РОЦ — регистр ошибок процессора.

Эти регистры фиксируют состояние машины и выдают управляющую информацию в процессе выполнения программ и команд.

В регистре РССП хранится текущее слово состояния программы.

На регистр РВО принимается код выполняемой команды и хранится там до ее завершения.

Регистр фиксации прерываний (РФП) служит для запоминания запросов на прерывания и используется схемой приоритетов для определения очередности их обработки.

Регистр ошибок процессора (РОЦ) предназначен для фиксации и хранения причин программных прерываний, например в случаях переполнения в операциях с фиксированной запятой, переполнения порядка в операциях с плавающей запятой, ошибок адресации и др.

Состояние триггеров, регистров, входящих в блок БУР, передается в блок БМУ для анализа и определения последующих действий при выполнении программы. Например, если в регистре ошибок процессора РОЦ один из триггеров установится в единичное состояние, то блок микропрограммного управления вызовет микропрограмму обработки ошибки, зафиксированной в регистре РОЦ.

После считывания очередной команды рабочей программы содержимое регистра РВО передается в регистр адреса РАДЗУ для образования начального адреса вызываемой микропрограммы.

БООП — блок обращения к оперативной памяти предназначен для приоритетного обслуживания запросов к оперативной памяти со стороны процессора, каналов и машинных часов (таймера).

В состав блока входят следующие основные элементы:

1. Регистр адресного слова памяти — РАСП, служащий для размещения в нем адресов инструкций или данных.

2. Регистр запросов, предназначенный для запоминания запросов от процессора, каналов и таймера.

3. Схема выработки приоритетов запросов.

Кроме перечисленных в состав БООП входит и ряд других элементов, используемых при формировании управляющих сигналов.

Для организации обращения к оперативной памяти устанавливается в «1» соответствующий запрашиваемому устройству триггер регистра запросов.

При поступлении одновременно нескольких запросов проверяется приоритет запрашивающих устройств и одному из них посылается сигнал, разрешающий обращение к оперативной памяти.

Сигнал «Оч=0» разрешает обращение к ОП каналам.

Сигнал «Оч=2» разрешает обращение к ОП таймеру.

Сигнал «Оч=4» разрешает обращение к ОП процессору.

Сигналы перечислены в порядке их приоритета.

В регистр РАСП из блока БУР или из блока арифметического логического или от каналов передаются адреса команд либо данных, по которым осуществляется чтение или запись информации.

МП — местная память, предназначена для хранения регистровых операндов, а также служебной информации, часто используемой в процессе вычислений, например значений индексов, базы адресов, счетчиков. В состав местной памяти входят: 16 регистров общего назначения (РОН); 4 регистра с плавающей запятой (РПЗ) и служебные регистры.

В регистры адреса и информации местной памяти (РАМП и РИМП) передачи осуществляются из регистра Р5 блока БАЛ.

ПУ — пульт управления, представляет собой самостоятельный блок, предназначенный для выбора режимов работы и индикации состояний ЭВМ ЕС-1030.

Для обеспечения различных наладочных режимов работы блок пульта управления имеет связи с блоками БМУ, ДЗУ и оперативной памятью. С блока ПУ можно набирать и засылать адреса в регистры РАСП (блока БООП) и РАДЗУ (блока ДЗУ), а также заносить или считывать информацию по этим адресам. Набор адресов и информации осуществляется с помощью тумблеров.

БСИ — блок синхронизации. Предназначен для синхронизации работы блоков процессора. Состоит из генератора синхроимпульсов и узла распределения, обеспечивающего разветвление синхроимпульсов по блокам и устройствам процессора.

Тактовая частота работы блоков процессора примерно равна 1,5 МГц. За время одного такта $T \simeq 670$ нс считывается (из долговременного запоминающего устройства) и выполняется одна микрокоманда.

ПКЗ — память ключей защиты, предназначена для хранения ключей защиты, назначаемых программам перед их размещением в оперативной памяти.

В ЭВМ ЕС-1030 действует система защиты памяти по записи и чтению. С этой целью при каждом обращении к оперативной памяти в блоке БООП сравнивается ключ защиты программы и ключ памяти, считанный из ПКЗ. При обращении к ОП из процессора ключ защиты программы поступает из регистра слова состояния программы (РССП), находящегося в блоке БУР, а при обращении к ОП со стороны каналов — из регистра слова состояния канала. Ключ памяти, поступающий в БООП из памяти ключей защиты, считывается автоматически при каждом обращении к ОП. Для этого при занесении адреса в регистр РАОП одновременно часть разрядов этого адреса заносится в регистр адреса ПКЗ. При совпадении ключа памяти с ключом защиты программы считывание из оперативной памяти или запись в нее разрешается, а при несовпадении — запрещается. При указании нулевого ключа защиты разрешается обращение в любую область памяти. Запись ключей памяти осуществляется с помощью специальных команд.

Блоки контроля (БК) и диагностики (БД) предназначены для контроля обрабатываемой информации и для определения неисправностей, возникающих в процессе выполнения программ.

В процессоре реализованы система контроля по четности, обеспечивающая проверку передач информации между блоками и устройствами, а также система контроля арифметических и логических операций, выполняемых в блоке БАЛ.

В блок контроля входит целый ряд схем аппаратного контроля, замыкающихся на регистр машинных отказов (РМО). Установка в единичное состояние хотя бы одного триггера регистра РМО говорит о наличии сбойной ситуации или ошибки, возникшей при выполнении вычислений.

Блок диагностики (БД) предназначен для записи в ОП состояния регистров процессора в момент возникновения сбойной ситуации и последующего их анализа с целью локализации неисправности при помощи тестовых программ.

Блок прямого управления БПРУ предназначен для обмена информацией и управляющими сигналами с другой ЭВМ или специальным устройством. Байт информации и синхронизирующие сигналы, подлежащие передаче, поступают в блок БПРУ из блока БАЛ и по интерфейсу прямого управления передаются в другую ЭВМ в аналогичный блок прямого управления. Информация, поступающая в блок БПРУ извне, передается в БАЛ для последующей записи в оперативную память.

Управляющие триггеры процессора

ТРС — триггер рабочего состояния.

Этот триггер предназначен для фиксации режимов работы машины «Останов» или «Работа» и при этом устанавливается в «1» или «0» вручную с пульта управления.

TRC-1 соответствует режиму «Останов».
TRC-0 соответствует режиму «Работа».

ТЗП — триггер записи.

ТЗП управляет режимами «Запись» и «Чтение» оперативной памяти.

ТЗП-1 соответствует режиму «Чтение».

ТЗП-0 соответствует режиму «Запись».

ТРКП — триггер регистра коммутатора процессора.

Единичное состояние триггера разрешает выдачу информации с регистра коммутатора процессора в регистр РИОП при записи информации, а нулевое — запрещает.

ТДАН — триггер данных.

Устанавливается в «1» при обращении в память за инструкциями (командами) и в «0» при обращении в ОП за операндами или данными.

ТДБ — триггер длины байтов.

Единичное состояние триггера (ТДБ-1) говорит о том, что обрабатываются операнды длиной в двойное слово (8 байтов). Если триггер ТДБ-0, то обрабатываются операнды длиной в одно слово (4 байта).

ТФАЦ — триггер фиксированных адресов процессора.

Устанавливается в «1» при обращении к ОП по фиксированному адресу со стороны процессора, например при записи «старого» или считывании «нового» ССП.

ТФАК — триггер фиксированных адресов канала.

Устанавливается в «1» при обращении к ОП по фиксированному адресу со стороны каналов, например при записи слова состояния канала — ССК.

ТИВ — триггер интервалов времени.

Устанавливается в «1» с частотой сети 50 Гц (через каждые 20 мс) и этим инициирует вызов микропрограммы корректировки машинных часов — таймера. После вызова микропрограммы корректировки сбрасывается в «0».

Состояние перечисленных управляющих триггеров и ряда других триггеров анализируется процессором при выполнении микрокоманд, и на основании этого анализа организуются ветвления (переходы) в микропрограммах.

6.3. Взаимодействие блоков процессора при выполнении команд

Как было отмечено, вычисления в ЭВМ осуществляются по заранее составленным программам, представленным в виде последовательностей машинных команд, записанных в оперативную память машины. Выполнение всех машинных команд, а также таких операций, как «сброс» машины в исходное состояние, обработка прерываний, корректировка машинных часов — таймера, и другие действия осуществляются в процессоре по микропрограммам, помещенным в долговременное запоминающее устройство ДЗУ.

Каждая микропрограмма, представляющая собой некоторую последовательность микрокоманд, определяет очередность выработки управляющих сигналов, необходимых для выполнения данной команды или действия.

Выполнение машинных команд в процессоре организуется в два этапа. Первый этап — начальная выборка команды — состоит в извлечении из оперативной памяти самой команды и размещении отдельных ее компонентов: кода операции, адресов операндов и их значений в регистрах блоков БАЛ и БУР.

Второй этап — непосредственное выполнение команды — заключается в выполнении конкретных действий, определяемых кодом команды, над операндами.

В соответствии с этим в процессоре выполняются сначала микропрограмма начальной выборки, затем микропрограмма конкретной команды. Алгоритмы микропрограмм начальной выборки команд форматов *RR*, *RX*, *RS*, *SI*, *SS* и микропрограммы некоторых команд представлены в альбоме схем.

Укрупненный алгоритм микропрограммы начальной выборки («Выборка») изображен на рис. 6.3.1.

Рассмотрим основные фазы выполнения микропрограммы «Выборка» по структурной схеме процессора и алгоритму, изображенным на рис. 6.2.1 и 6.3.1 (на структурной схеме процессора и в алгоритме начало отдельных фаз отмечено цифрами в кружках).

1-я фаза. Проверка режимов работы машины.

В регистр адреса ДЗУ кнопкой пульта управления ПУСК или автоматически, после выполнения очередной команды рабочей программы, заносится адрес первой микрокоманды (1.0.15) микропрограммы ВЫБОРКА.

После этого из накопителя ДЗУ в регистры информации РИДЗУ и РИДЗУД помещается первая микрокоманда. Затем осуществляется ее дешифрация и вырабатываются элементарные операции, обеспечивающие анализ режимов работы машины.

1. Анализируется состояние триггера ТРС, определяющее программные состояния «Останов» или «Работа».

Если триггер ТРС находится в единичном состоянии, то это соответствует режиму «Останов», и поэтому осуществляется за-

цикливание первой микрокоманды микропрограммы ВЫБОРКА.

2. Если триггер ТРС сброшен в «0» (переключатель режимов работы стоит в положении АВТОМ), то анализируется состояние 14 разряда регистра РССП.

При значениях РССП[14]=1 и триггера ТРС=0 процессор находится в режимах «Работа» и «Ожидание» одновременно. В этом случае организуется переход к микрокоманде по адресу 1.0.1, которая выполняет проверку наличия запросов на прерывания в регистре фиксации прерываний (РФП) блока БУР и, если их нет, передает управление микрокоманде по адресу 1.0.15. Так организуется цикл, обеспечивающий ожидание запросов на прерывания. При появлении запроса на прерывание, например прерывания от ввода — вывода, вызывается микропрограмма обработки прерывания, которая обеспечивает запоминание «старого» и вызов «нового» слова состояния программы и сброс РССП[14] в «0» (переход из режима «Ожидание» в режим «Счет»), после чего управление вновь передается микропрограмме ВЫБОРКА.

Если РССП[14]=0 и триггер ТРС=0, то процессор находится в режимах «Работа» и «Счет». Поскольку в режиме «Счет» выполнение команд рабочей программы разрешено, то осуществляется переход ко 2-й фазе ВЫБОРКИ.

2-я фаза. Чтение команды из ОП. В этой фазе выполняются следующие действия:

1. Засылка адреса команды рабочей программы из регистра РССП[40/63] через блок БООП в регистр адреса оперативной памяти — РАОП:

РССП [40/63] → РАСП → РАОП.

2. Чтение команды из ОП и засылка ее в блоки БАЛ и БУР:
РИОП → Р5. (передача всей или части команды в Р5).

Поскольку команды формата SS занимают в памяти полтора слова, то для чтения всей команды возможно повторное обращение к ОП, так как считывание из ОП осуществляется словами.

После чтения команды проверяются возможные ошибки чтения из оперативной памяти и, если они есть, вызывается программа обработки этих ошибок.

При отсутствии ошибок по числовому коду в 1-м и 2-м разрядах кода операции расшифровывается формат команды и определяется ветвь микропрограммы ВЫБОРКА, обеспечивающая выборку операндов данного формата.

3-я фаза. Чтение операндов из ОП и местной памяти (МП) и размещение их в регистрах блока БАЛ.

Адреса операндов А1 и А2, сформированные в сумматоре блока БАЛ, передаются в регистр адресного слова памяти — РАСП блока БООП, а с него — в регистр РАОП оперативной памяти и ОП запускается в режим «Чтение». Считанные из накопителя в

регистр РИОП операнды передаются в блок БАЛ. За регистровыми операндами обращение организуется в местную память.

4-я фаза. Передача управления микропрограмме выполняемой команды.

Адресом микропрограммы выбираемой команды служит код операции команды, находящийся в регистре выполняемой операции — РВО.

Этот код операции засылается в регистр адреса — РАДЗУ долговременного запоминающего устройства:

РВО → РАДЗУ.

Например, коду операции 1А (сложение с фиксированной запятой) соответствует адрес 0.1.10 долговременного запоминающего устройства. С этого адреса начинается микропрограмма — **сложение с фиксированной запятой.**

Крайний левый нуль добавляется в соответствующие разряды регистра РАДЗУ при занесении адреса из регистра РВО, а 1 и А — это шестнадцатиричные цифры, которым соответствуют десятичные 1 и 10.

Выполнение команды в блоках процессора

Команды рабочей программы выполняются по микропрограммам, алгоритмы которых, как правило, включают в себя следующие фазы:

1. Передача операндов на входы сумматора блока БАЛ и получение на его выходе конечного или промежуточного результата вычислений.

2. Проверка ошибок, обнаруженных в процессе обработки и передачи операндов; наличие ошибок фиксируется в регистре ошибок процессора (РОЦ) блока управляющих регистров.

а) При обнаружении ошибок (РОЦ $\neq 0$) выставляется запрос на прерывание и осуществляется переход на микропрограмму обработки ошибок.

б) Если ошибок нет (РОЦ = 0), то организуются завершение выполнения операции, выработка кода условия и занесение его в РССП [34, 35].

3. Запись результата в ОП или МП по адресу 1 операнда, если это требуется при выполнении данной команды.

4. Проверка ошибок оперативной памяти.

5. Проверка на необходимость корректировки машинных часов — таймера.

В зависимости от выполняемой команды могут осуществляться и другие проверки и действия, не названные здесь.

Алгоритм обработки прерываний

Укрупненный алгоритм микропрограмм обработки прерываний с пояснением отдельных его этапов изображен на рис. 6.3.2.

Перед началом выполнения микропрограммы **ВЫБОРКА** всегда анализируется регистр фиксации запросов на прерывания (РФП). Если хотя бы один из триггеров регистра РФП хранит запрос на прерывание, то организуется переход на соответствующую запросу микропрограмму прерывания и последующий переход на программу обработки этого запроса. При обнаружении ошибок выход на прерывание возможен как из микропрограммы **ВЫБОРКА**, так и из любой микропрограммы выполняемой команды.

Обработка запросов от каналов

Параллельно с выполнением команд рабочей программы процессор обеспечивает обращение к оперативной памяти со стороны селекторных и мультимплексного каналов, запросы которых через общий канал поступают в блок БООП.

Если в процессе выполнения команды в регистр запросов (РЗПР) блока БООП придет запрос на обращение к ОП со стороны каналов (на рис. 6.2.1 сигналы «Запрос от канала» и «адр от кан.»), то блок БООП принимает в регистр РАСП адрес ОП от канала и выдает ему разрешение на обращение к ОП (сигнал Оч-0) и одновременно запускает оперативную память на запись или чтение информации. В режиме «Запись» информация в регистр РИОП поступает из общего канала.

Если в момент прихода запроса на обращение к ОП со стороны каналов в оперативной памяти осуществляется запись или считывание информации по запросу процессора или таймера, то выдача сигнала разрешения обращения к ОП каналу (**Оч=0**) будет задержанс до завершения работы ОП.

После восприятия процессором от канала сигнала на прерывание по вводу — выводу и выхода на прерывание в канал передается сигнал «Блокировка», которым каналу сообщается о том, что запрос на прерывание процессором принят, одновременно этим же сигналом блокируется работа процессора на время завершения каналом действия по прерыванию (занесение кода прерывания в РССП[16/31], формирование и запись в ОП слова состояния канала).

6.4. Блок арифметическо-логический

Назначение и состав блока

Арифметическо-логический блок (БАЛ) процессора ЭВМ ЕС-1030 предназначен для выполнения арифметических и логических операций и операций над адресами.

Операции в БАЛ выполняются над числами, представленными в двоичной и двоично-десятичной системах счисления. Обработка чисел производится либо в режиме с фиксированной запятой, либо в режиме с плавающей запятой.

БАЛ условно можно разделить на два узла: арифметическо-логический узел, в котором обрабатывается информация в виде полуслов, слов и двойных слов, и узел обработки байтов.

Структурная схема блока изображена в альбоме схем на рис. 6.4.1.

Рассмотрим назначение основных элементов блока.

Регистр P1.

P1 — 32-разрядный одновходовый запоминающий регистр, предназначенный для размещения второго слагаемого, вычитаемого, делителя и одного из операндов в логических операциях.

Регистр P1 принимает информацию из регистра P5. Управление приемом в P1 осуществляется отдельными элементарными операциями, позволяющими принимать информацию в определенные группы разрядов.

Информация с выхода регистра P1 может поступать на входной регистр сумматора РА в прямом коде (прямо) или в обратном коде (инверсно); старшие 16 разрядов информации из регистра P1 поступают в младшие 16 разрядов регистра РА. Каждый байт информации с выхода регистра P1 может быть передан на вход регистра РС (рис. 6.4.1).

Информация с выхода нулевого разряда регистра P1 поступает на триггер знака второго операнда (ТЗН2) прямо и инверсно. Регистр P1 собран на ТЭЗ ЕС-Т000/0001 по четыре разряда в каждом ТЭЗ и расположен в раме А стойки процессора по адресам: P1 [0/3] — 1E07, P1 [4/7] — 1E10, P1 [8/11] — 1H07, P1 [12/15] — 1H10, P1 [16/19] — 2E11, P1 [20/23] — 2E14, P1 [24/27] — 2H12, P1 [28/31] — 2A14.

Регистр P2.

P2 — 32-разрядный одновходовый запоминающий регистр, предназначенный для размещения второго слагаемого, младшего слова второго операнда (если он двойной длины), вычитаемого и делителя.

Регистр P2 принимает информацию из регистра P5 двумя группами: P2[0/7] := P5[0/7] и P2[8/31] := P5[8/31].

Информация с входов регистра P2 поступает на входной регистр сумматора РВ.

Регистр P2 собран на ТЭЗ ЕС-Т000/0001 по четыре разряда в каждом ТЭЗ и расположен в раме А стойки процессора по тем же адресам, по которым расположен регистр P1.

Регистр P3.

P3 — 32-разрядный одновходовый запоминающий регистр, предназначенный для размещения первого слагаемого, уменьшаемого, делимого, одного из операндов в логических операциях и операциях преобразования и сдвигов. Кроме того, регистр P3 исполь-

зуется для записи информации из БАЛ в оперативную память через регистр коммутатора процессора (РКП) с помощью элементарной операции $РКП := РЗ$.

Регистр РЗ принимает информацию из регистра Р5. Информация с выходов регистра РЗ поступает на входной регистр сумматора РВ прямо и инверсно; старшие 16 разрядов информации из регистра РЗ поступают в младшие 16 разрядов регистра РВ. Каждый байт информации с выходов регистра РЗ может поступать на вход регистра РД (рис. 6.4.1).

Информация с выхода нулевого разряда регистра РЗ передается на триггер знака первого операнда (ТЗН1). Младший байт информации из регистра РЗ поступает на регистр информации памяти ключей защиты. Информация с выходов РЗ[16/19] поступает в регистр адреса местной памяти.

Регистр РЗ собран на ТЭЗ ЕС-Т000/0001 по четыре разряда в каждом ТЭЗ и расположен в раме А стойки процессора по тем же адресам, по которым расположены регистры Р1, Р2.

Регистр Р4.

Р4 — 32-разрядный многовходовый сдвигающий регистр с возможностью сдвига информации на два разряда вправо или четыре разряда влево за один такт.

Регистр Р4 предназначен для размещения первого слагаемого или младшего слова первого операнда (если он двойной длины), уменьшаемого, частного, множителя и адреса второго операнда.

Регистр Р4 принимает информацию из регистров Р5, Р4, четыре младших разряда регистра Р4 принимают информацию из регистра Р6, а два старших разряда регистра Р4 принимают информацию из двух старших разрядов регистра Р6.

Информация с выходов регистра Р4 поступает на входной регистр сумматора РВ, с выходов старших четырех разрядов регистра Р4 информация поступает на четыре младших разряда регистра Р5, младший байт информации с регистра Р4 поступает на регистр Р8, нулевой разряд регистра Р4 соединен со входом триггера Т30.

Для сдвига информации вправо на два разряда или влево на четыре разряда выходы регистра Р4 соответствующим образом соединены со входами регистра Р4" (рисунки 6.4.2 и 6.4.3). Сдвиг информации осуществляется путем передачи информации из регистра Р4 на регистр Р4" с соответствующим сдвигом и обратной прямой передачей из регистра Р4" на регистр Р4.

Регистр Р4 собран на ТЭЗ ЕС-Т000/0002, ЕС-2030/00013 и расположен в раме А стойки процессора по адресам: Р4 [0/1] — 1Е09, Р4 [2/7] — 1Е08, Р4 [8/15], — 1Н08, Р4 [16/23] — 2Е13, Р4 [24/27] — 2Н13, Р4 [28/31] — 2Н07.

Регистр P4''.

P4'' — 32-разрядный двухходовый запоминающий вспомогательный регистр, предназначенный для осуществления совместно с регистром P4 сдвига информации вправо на два разряда или влево на четыре разряда.

Регистр P4'' собран на ТЭЗ ЕС-T000/0002 и расположен в раме А стойки процессора по адресам: P4'' [0/7] — 1E04, P4'' [8/15] — 1H04, P4'' [16/23] — 2E17, P4'' [23/31] — 2H17.

Регистр PA.

PA — 32-разрядный многоходовый запоминающий входной регистр сумматора, предназначенный для хранения второго слагаемого. Регистр PA принимает информацию из регистра P1 прямо и инверсно. Регистр PA имеет входы для приема в каждую тетраду двоичной цифры шесть и цифры десять и вход общей установки в ноль.

Выходы PA непосредственно связаны со входами комбинационной схемы сумматора.

Регистр PA собран на ТЭЗ ЕС—2030/0005 по четыре разряда в каждом и расположен в раме А стойки процессора по адресам: PA [0/3] — 1E06, PA [4/7] — 1E11, PA [8/11] — 1H06, PA [12/15] — 1H11, PA [16/19] — 2E09, PA [20/23] — 2E15, PA [24/27] — 2H10, PA [28/31] — 2H15.

Регистр PB.

PB — 32-разрядный многоходовый запоминающий входной регистр сумматора, предназначенный для хранения первого слагаемого.

PB принимает информацию из регистров P2, P4 и из регистра P3 — прямо и инверсно. В него также могут передаваться константы из ДЗУ.

Выходы регистра PB связаны непосредственно со входами комбинационной схемы сумматора. Регистр PB собран на ТЭЗ ЕС-2030/0001, за исключением младших восьми разрядов, собранных на ТЭЗ ЕС-2030/0005.

Регистр PB размещен в раме А стойки процессора по адресам: PB [0/3] — 1E05, PB [4/7] — 1E12, PB [8/11] — 1H05, PB [12/15] — 1H12, PB [16/19] — 2E10, PB [20/23] — 2E16, PB [24/27] — 2H11, PB [18/31] — 2H16.

Сумматор.

Комбинационный сумматор (СМ) (рис. 6.4.1) служит для формирования суммы двух чисел, хранимых на входных регистрах PA и PB. Полученная сумма хранится на выходном регистре P5.

Сумматор состоит из схемы формирования полусуммы и схемы ускоренного переноса.

Сумматор 32-разрядный разбит на восемь групп (рис. 6.4.4). Кроме того, сумматор имеет четыре дополнительных разряда, названные сумматором дополнительным (СМД). Сумматор дополнительный используется в операциях над десятичными числами и операциях с плавающей запятой. В качестве входных регистров для сумматора дополнительного используются регистры Р6 и РХ, а в качестве выходного — регистр Р6".

Числа, подлежащие обработке, поступают на входные регистры РА и РВ.

В таблице 6.4.1 показан процесс суммирования двух одnorазрядных чисел A и B .

Т а б л и ц а 6.4.1

Числа		Перенос из предыд. раз- ряда ПЕР ($n-1$)	Результат	
A	B		сумма S	перенос в следующий разряд ПЕР (n)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
0	1	1	0	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Логические выражения для n -го разряда суммы (S) и переноса в n -й разряд ПЕР (n), написанные по единичным значениям A , B и ПЕР (n), имеют следующий вид:

$$S = \bar{A} \cdot \bar{B} \cdot \text{ПЕР } (n-1) \vee \bar{A} \cdot B \cdot \overline{\text{ПЕР}}(n-1) \vee A \bar{B} \cdot \overline{\text{ПЕР}}(n-1) \vee$$

$$AB \text{ ПЕР } (n-1) = (AB \vee \bar{A} \bar{B}) \cdot \text{ПЕР } (n-1) \vee$$

$$(\bar{A} B \vee A \bar{B}) \cdot \overline{\text{ПЕР}}(n-1)$$

$$\text{ПЕР } (n) = \bar{A} \cdot B \cdot \text{ПЕР } (n-1) \vee A \bar{B} \text{ ПЕР } (n-1) \vee A \cdot B \cdot \overline{\text{ПЕР}}(n-1) \vee$$

$$\bar{A} \cdot B \cdot \text{ПЕР } (n-1) \vee A \cdot B \cdot \overline{\text{ПЕР}}(n-1) \vee (\bar{A} B \vee A \bar{B}) \vee$$

$$AB \overline{\text{ПЕР}}(n-1).$$

На рис. 6.4.5 показана комбинационная схема одного разряда сумматора, реализующая указанное логическое выражение для получения суммы.

На вход микросхем 1ЛР551 поданы прямые и инверсные значения чисел. Полученные на выходах схем сигналы соответствуют выражениям $AB \vee \bar{A} \bar{B}$ и $\bar{A} B \vee A \bar{B}$. Эти сигналы поступают на схему 1ЛР551 вместе с прямым и инверсным значениями сигнала, соответствующего переносу из предыдущего разряда ПЕР ($n-1$) и $\overline{\text{ПЕР}}(n-1)$. В результате получается сигнал, соот-

ветствующий инверсному значению суммы ($\bar{S}$), который после инвертирования поступает на соответствующий разряд регистра Р5 в качестве суммы входных сигналов А и В.

В альбоме на рис. 6.4.6 показана упрощенная схема группы сумматора на четыре разряда (24, 25, 26, 27), на рис. 6.4.7 показана функциональная схема тех же четырех разрядов сумматора, а на рис. 6.4.8 — схема образования переносов.

Регистр Р5.

Р5 — 32-разрядный многовходовый запоминающий регистр, предназначенный для запоминания конечного или промежуточного результата (суммы, частного, произведения, остатка), а также информации, подлежащей сдвигу или записи в местную память.

Р5 принимает информацию из следующих источников: СМ, Р6, РС1, Р7, СМДЕС, ПУ, ОП, РССП, МП.

Информация с выходов регистра Р5 поступает на регистры Р1, Р2, Р3, Р4, Р8, Р7, РССП, РВО, РИМП и счетчики РС2, РС3, РС4, РС5, РС6.

Регистр Р5 собран на ТЭЗ ЕС-Т001/0004 и ЕС-2030/0002, первый из которых основной, а второй — расширитель.

Регистр Р5 расположен в раме А стойки процессора в панелях 1С и 2С.

Регистры Р6 и Р6".

Р6 — четырехразрядный многовходовый сдвигающий регистр, предназначенный для накопления цифры частного и произведения, для размещения в нем цифры в операциях сдвига и нормализации.

Регистр Р6 принимает информацию из сумматора, регистра Р6" и триггера Т10.

Информация с выходов регистра Р6 поступает на регистры Р4, Р5, Р6", сумматор дополнительный, триггеры Т10, Т20, Т30.

Регистр Р6 собран на ТЭЗ ЕС-2030/0015, расположенном в раме А стойки процессора по адресу 1Д01.

Регистр Р6" — четырехразрядный многовходовый запоминающий регистр, используемый в качестве вспомогательного вместе с регистром Р6. Он принимает информацию из регистра Р6 и сумматора дополнительного.

Информация с выходов регистра Р6" поступает на регистр Р6 и триггер Т30.

Триггеры Т10, Т20, Т30 используются при выполнении различных операций.

Регистр Р6" собран на ТЭЗ ЕС-2030/0013, расположенном в раме А стойки процессора по адресу 1Д03:

Регистр РХ.

РХ — четырехразрядный запоминающий вспомогательный регистр, предназначенный для преобразования в дополнительный код младшей тетрады сумматора.

Информация с выходов регистра РХ поступает на входы комбинационной схемы сумматора дополнительного.

Регистр РХ собран на ТЭЗ ЕС-2030/0013, расположенном в раме А стойки процессора по адресу 1Д02:

Регистры РС и РД.

РС и РД — восьмиразрядные многоходовые регистры, предназначенные для хранения одного байта информации или порядка в операциях с плавающей запятой (рис. 6.4.9).

Регистр РС принимает информацию из регистров РС1, РС3 и Р1 побайтно.

Регистр РД принимает информацию из регистров РС1, РС4 и Р3 побайтно.

Информация с выходов регистров РС и РД поступает на входы комбинационной схемы узла обработки байтов (ГОВ) и на входы комбинационной схемы сумматора десятичного. Регистр РС собран на ТЭЗ ЕС-2030/0005 по четыре разряда в каждом и расположен в раме А стойки процессора по адресам: РС [0/3] — 1Д06, РС [4/7] — 1Д07. Регистр РД на ТЭЗ ЕС-2030/0005 по четыре разряда в каждом и расположен в раме А стойки процессора по адресам: РД [0/3] — 1Д08, РД [4/7] — 1Д09.

Сумматор десятичный (СМ ДЕС).

СМ ДЕС — восьмиразрядный сумматор комбинационного типа, предназначенный для выполнения десятичных операций — сложения, вычитания, сравнения и сложения с очисткой.

Входными регистрами сумматора десятичного служат регистры РС и РД. Информация с выходов СМ ДЕС поступает на любой байт регистра Р5.

Формирование суммы происходит в сумматоре, состоящем из схемы получения полусуммы и схемы образования и распространения переноса.

Двоичная сумма схемно корректируется для получения десятичной суммы.

Информация из регистра РС на вход сумматора может подаваться инверсно или с избытком 6, для чего на выходе регистра РС собрана схема предварительной корректировки.

Если знаки обоих операндов с учетом выполняемой операции одинаковые (триггер вариантов установлен в «1»), то в каждую тетраду первого операнда добавляется цифра 6. Если знаки операндов разные (триггер вариантов установлен в «0»), то один из операндов инвертируется.

Регистр Р7.

Р7 — восьмиразрядный многоходовый регистр, предназначенный для хранения результата обработки одного байта информации.

Регистр Р7 принимает информацию из узла обработки байтов и регистров Р5, РИПКЗ.

Информация с выходов регистра Р7 поступает на регистры Р5 и РС1. Регистр Р7 собран на двух ТЭЗ ЕС-2030/0005 и расположен в раме А по адресам: Р7[0/3] — 1С10 и Р7[4/7] — 1С11.

Регистр Р8.

Р8 — восьмиразрядный многоходовый регистр, предназначенный для промежуточного хранения адресов первого и второго операндов, знаков десятичных операндов, а также заполняющего знака в операции РЕДАКТИРОВАТЬ.

Регистр Р8 принимает информацию из регистров Р4, Р5, РС1.

Информация с выходов регистра Р8 поступает в регистры РАМП, РС1, и триггеры Т10 и Т20.

Регистр Р8 собран на двух ТЭЗ ЕС-2030/0013 и расположен в раме А стойки процессора по адресам: Р8[0/3] — 1С05 и Р8[4/7] — 1С06.

Регистр счета адреса (РСА).

РСА — четырехразрядный двухходовый регистр, используемый вместе с регистром адреса местной памяти (РАМП) для формирования адреса местной памяти. Для чего регистр РСА принимает информацию из регистра РЗ ($РСА := РЗ[16/19]$). РСА может работать в счетном режиме ($РСА := +1$), что используется для изменения адреса местной памяти. Сформированный на регистре РСА адрес передается в регистр РАМП ($РАМП[2/5] := РСА$).

Регистр РСА собран на ТЭЗ ЕС-2030/0016, расположенном в раме А стойки процессора по адресу 1В01.

Регистры РСА и РАМП на схеме не показаны.

Регистры РС1 и РС1'.

РС1 — восьмиразрядный двоичный счетчик, используемый для подсчета циклов в операции деления с фиксированной запятой; в операциях групповой загрузки и групповой записи в оперативную память; при выравнивании порядков и нормализации операндов в операциях с плавающей запятой; для подсчета длин операндов в логических операциях и операциях с десятичными операндами.

В регистре РС1 имеется возможность складывать и вычитать единицу, вычитать четыре. Регистр РС1 принимает информацию из регистров Р7 и Р8.

РС1[0/3] и РС1[4/7] могут считать независимо друг от друга как два четырехразрядных счетчика. В этом режиме РС1 используется в операциях десятичной арифметики, групповых операциях загрузки и записи в память.

Регистр РС1 как восьмиразрядный счетчик используется в операциях с плавающей запятой при действиях с порядками.

Информация из регистра PC1 поступает на регистры P5, P8, PC, PД, РАМП, PC1".

PC1" — восьмиразрядный одновходовый регистр, повторяющий содержимое регистра PC1 (в альбоме на схеме не показан). Передача из PC1 в PC1" осуществляется автоматически в каждом такте, независимо был прием в PC1 или нет.

Регистр PC1 собран на двух ТЭЗ ЕС-2030/0010 по четыре разряда в каждом и расположен в раме А стойки процессора по адресам: PC1[0/3] — 1C07, PC1[4/7] — 1C08.

Регистр PC1" собран на ТЭЗ ЕС-T000/0001 и расположен в раме А стойки процессора по адресу 1B09.

Регистры PC2 и PC2".

PC2 — четырехразрядный двоичный счетчик, предназначенный для подсчета циклов в операциях умножения и деления.

В регистре PC2 осуществляется сложение и вычитание единицы.

Регистр PC2 принимает информацию из P5[28/31] и константу из РИДЗУ [48/51].

На выходах регистра PC2 собрана схема анализов: $PC2 \neq 0$, $PC2 = 1000$, $PC2 \neq 1111$.

PC2" — четырехразрядный запоминающий одновходовый вспомогательный регистр, повторяющий содержимое PC2 (в альбоме на схеме не показан).

Передача $PC2'' := PC2$ производится автоматически в каждом такте.

Регистр PC2 собран на ТЭЗ ЕС-2030/0010 и расположен в раме В стойки процессора по адресу 1E01.

Регистр PC2" собран на ТЭЗ ЕС-T000/0001 и расположен в раме В стойки процессора по адресу 1E03.

Регистры счета PC3, PC4, PC5, PC6.

Вспомогательные регистры PC3'', PC4'', PC5'', PC6''.

PC3, PC4, PC5, PC6 — двухразрядные двоичные счетчики, которые осуществляют сложение и вычитание единицы.

PC3 принимает информацию из P5[30/31] и PC6, PC4 принимает информацию из P5[30/31] и PC5, PC5 принимает информацию из P5[30/31], PC6 принимает информацию из P5[30/31] и PC5.

Информация поступает с выходов PC3 на PC[4/5], с выходов PC4 — на PД [4/5] и на РАДЗУ [9/10], с выходов PC5 — на PC4 и PC6, с выходов PC6 на PC3.

Регистры PC3—PC6 собраны на ТЭЗ ЕС-2030/0010 и расположены в раме В стойки процессора по адресам: PC3, PC4 — 1E0C, PC5, PC6 — 1E04.

Регистры PC3"—PC6" — запоминающие двухразрядные, одновходовые регистры (в альбоме на схеме не показаны), повторяющие содержимое PC3—PC6 соответственно. Передача из PC3—PC6 в PC3"—PC6" производится автоматически в каждом полутакте C1.

Регистры РС3"—РС6" собраны на ТЭЗ Т000/0001 и расположены в раме В стойки процессора по адресу 1Е03.

Узел обработки байтов (ГОб).

Для обработки байтов используется комбинационная схема и схема сравнения на один байт. В качестве входных регистров использованы регистры РС и РД, а в качестве выходного — регистр Р7.

Узел обработки байтов предназначен для:

1. Выборки нужного байта из 32-разрядного слова или для изменения любого байта в слове.
2. Реализации логических операций умножения, логических операций сложения.
3. Замещения четырех разрядов (старших или младших) одного байта четырьмя разрядами другого.
4. Выполнения поразрядного сравнения двух байтов.

Информация из узла обработки байтов через Р7 поступает в Р5 и РС1.

На рисунке 6.4.10 приведена схема ГОб для одного разряда.

На входы 07—10 схемы подведены нулевые и единичные выходы триггеров разрядов регистров РС и РД. Сигналы на выходах 1 схемы ГОб ((ТЭЗ ЕС-2030/0012 РТ06, 13, 17, 19) соответствуют функции поразрядного логического произведения при любых значениях одноименных разрядов регистров РС и РД. Сигналы на выходах 1 схемы ГОб устанавливаются в полутакте С1 в результате приема информации в регистры РС и РД или очистки одного из них.

В полутакте С2 при наличии разрешающего сигнала $P7 := PC \wedge PD$ информация с выходов схемы ГОб поступает на регистр Р7.

Сигналы на выходах 3 (ТЭЗ ЕС-2030/0012, РТ5, 29, 39, 45) соответствуют сумме по модулю 2 содержимого РС и РД.

Выходы 3 схемы также подведены на соответствующие входы регистра Р7. При наличии разрешающих сигналов $P7[0/3] := PC[0/3] \oplus PD[0/3]$ и $P7[4/7] := PC[4/7] \oplus PD[4/7]$ сумма по модулю 2 содержимого разрядов 0—3, 4—7 или 0—7 регистров РС и РД записывается в регистр Р7. По этой же цепи осуществляется передача содержимого одного из регистров РС и РД в регистр Р7. При этом второй регистр должен быть очищен. Выходы 3 разрядов 0—3 схемы ГОб подведены на входы Р7[4/7], а выходы 3 разрядов [4/7] схемы ГОб подведены на входы Р7[0/3].

Сигналы на выходах 3 схемы ГОб устанавливаются в полутакте С1 в результате приема информации в РС или РД или очистки одного из них. Соответствующие передачи в Р7 возможны в следующем полутакте С2.

Информация с выходов 2 схемы ГОб используется в схемах контроля и схеме сравнения.

Комбинационная схема сравнения на восемь разрядов осуществляет сравнение содержимого регистров РС и РД и выдает два сигнала: $PC > PD$ и $PC \neq PD$.

Эти сигналы можно опрашивать в полутакте С1 после приема информации в РС и РД.

Кроме того, имеются схемы сравнения для разрядов 0—3 регистров РС и РД, которые вырабатывают сигналы $PD[0/3] > PC[0/3]$ и $PD[0/3] = PC[0/3]$, и аналогичная схема сравнения для разрядов [4—7] этих же регистров, которая вырабатывает сигналы $PD[4/7] > PC[4/7]$ и $PD[4/7] = PC[4/7]$.

Операции, выполняемые в арифметическо-логическом блоке.

Арифметические и логические операции, выполняемые блоком, делятся на четыре класса: операции с фиксированной запятой, операции над десятичными числами, операции с плавающей запятой и логические операции.

Все команды, реализуемые в арифметическо-логическом блоке, выполняются по соответствующим микропрограммам.

Команды с фиксированной запятой.

Команды с фиксированной запятой выполняют операции над операндами с фиксированной запятой. Эти команды могут быть трех форматов: RR , RX и RS .

Команды с фиксированной запятой выполняют операции загрузки, сложения, вычитания, умножения, деления, записи в память, перевода чисел из десятичной системы счисления в двоичную и из двоичной системы в десятичную, сдвига, сравнения.

При выполнении команд с фиксированной запятой первый операнд всегда находится в одном из шестнадцати регистров общего назначения (РОН), второй операнд может находиться в РОН или в ОП.

Результат операции во всех командах помещается на место первого операнда. Исключение составляет команда записи в оперативную память и преобразования двоичной информации в десятичную, в которых результат помещается на место второго операнда.

При выполнении большинства команд с фиксированной запятой устанавливается признак результата, который может быть использован для выбора пути при выполнении последующих команд условного перехода. Признак результата устанавливается равным одному из четырех значений: 0, 1, 2 или 3. Для большинства операций признак результата 0, 1 и 2 указывает, что результат операции соответственно равен нулю, меньше нуля и больше нуля. В операциях сравнения признак результата 0, 1, 2 указывает, что первый операнд соответственно равен второму, меньше и больше его. Во всех операциях признак результата равен 3, указывает на то, что произошло переполнение.

При выполнении команд с фиксированной запятой могут возникать программные прерывания, по которым выполнение «текущей» программы прекращается. Прерывание программы вызывают следующие причины: защита памяти, адресация, спецификация, данные, переполнения, некорректность деления.

Прерывание по данным вызывает неправильный код знака или цифры в десятичном операнде при выполнении команды ПРЕОБРАЗОВАТЬ В ДВОИЧНУЮ.

Ниже рассматривается порядок выполнения некоторых команд.

Команды сложения AR (1A), A (5A).

Операции сложения с фиксированной запятой могут выполняться командами двух форматов *RR* и *RX*.

По командам *AR* и *A* производится суммирование первого и второго операндов.

После выборки из оперативной памяти первый операнд помещается в регистр *P3*, а второй — в регистр *P1*, код операции — в регистр *РАДЗУ*. Сложение производится в сумматоре после передачи операндов на входные регистры *РА* и *PВ*.

Результат с выхода сумматора поступает на выходной регистр *P5*, откуда он переписывается по адресу первого операнда.

Пример: сложить два числа:

$$\begin{array}{r} (+18) 0.000 100 10 \text{ PВ} \\ + (+90) 0.010 110 10 \text{ РА} \\ \hline (+108) (0.011) 011 00 \text{ СМ} \end{array}$$

В сложении участвуют все тридцать два разряда каждого операнда. В зависимости от результата устанавливается код условия. Если переносы из знакового разряда и старшего разряда поля целого числа одновременно или присутствуют или отсутствуют, то это значит, что сложение выполнено нормально.

Наличие переноса только в одном из указанных разрядов говорит о переполнении, которое может вызвать прерывание.

Пример: сложить два числа:

$$\begin{array}{r} (+11) 0.1011 \text{ PВ} \\ + (+6) 0.0110 \text{ РА} \\ \hline 1.0001 \text{ СМ} \end{array} \quad \text{— переполнение, (есть перенос только в зна-} \\ \text{ковый разряд).}$$

Команда вычитания SR (1B), S (5B).

Операции вычитания могут выполняться командами двух форматов *RR* и *RX*.

По командам *SR* и *S* второй операнд вычитается из первого. После выборки уменьшаемое находится в регистре *PВ*, а вычитаемое — в регистре *P1*, код операции — в регистре *РАДЗУ*. Вычитание представляет собой сложение первого операнда с дополнительным кодом второго.

Операция выполняется аналогично операции сложения, но в начале операции производится преобразование вычитаемого в дополнительный код ($PA := -P1$, $CM[31] := +1$).

В конце операции вырабатываются признак результата.

Пример: выполнить вычитание 87 (0 1010111) из 90 (0 1011010).

Преобразуем вычитаемое в дополнительный код

$$\begin{array}{r}
 + \quad 10101 \ 000 \\
 \quad \quad \quad 1 \\
 \hline
 10101 \ 001
 \end{array}
 \quad
 \begin{array}{r}
 + \ (+90) \\
 + \ (87) \text{ доп} \\
 \hline
 (+3)
 \end{array}
 \quad
 \begin{array}{r}
 0.1011010 \ PB \\
 + \ 1.0101001 \ PA \\
 \hline
 0.0000011 \ CM
 \end{array}$$

Команды сравнения CR (19), C (59).

Операции сравнения могут выполняться командами двух форматов *RR* и *RX*.

Сравнение операндов выполняется алгебраическим вычитанием второго операнда из первого. Сравнимые операнды рассматриваются как целые 32-разрядные числа со знаком.

Операнды в регистрах общего назначения и в оперативной памяти не изменяются. После выборки первый операнд находится в регистре *P3*, а второй — в регистре *P1*, код операции — в регистре *PAДЗУ*.

При выполнении операции производится преобразование второго операнда в дополнительный код ($PA := -P1$; $CM[31] := +1$). Выполнение операции производится в сумматоре.

Результатом выполнения команды является код условия, помещаемый в регистр *PCСП*.

Команды арифметического сдвига SLA (8B), SRA (8A).

Операции сдвига выполняются командой формата *SS*. Сдвигается целая часть первого операнда на количество разрядов, определяемое шестью младшими разрядами адреса второго операнда, находящимися в *P4[26/31]*. В этих операциях разряды адреса второго операнда для адресации данных не используются.

Знак операнда не сдвигается. Он сохраняется в нулевом разряде регистра *P6*. После выборки сдвигаемый операнд размещается в регистре *P1*, код операции — в *PAДЗУ*.

Число сдвигов заносится в счетчик *PC1*.

После каждого сдвига на один разряд из содержимого счетчика *PC1* вычитается единица и выполняется анализ содержимого счетчика на нуль. При равенстве содержимого счетчика *PC1* нулю операция заканчивается, сдвинутый операнд записывается по первому адресу команды.

При выполнении команды сдвига влево *SLA* в освободившиеся младшие разряды заносятся нули. В этой команде может возникнуть переполнение, которое выявляется анализом $P1[0/1] = 01 \vee 10$, производящимися после каждого сдвига на один разряд. При выполнении условия анализа фиксируется переполнение.

При выполнении команды сдвига вправо *SRA* в освободившиеся старшие разряды заносится значение знакового разряда.

Команды умножения *MR (1C)* и *M (5C)*.

Операция умножения выполняется командами форматов *RR* и *RX*.

По команде умножения первый (множимое) и второй (множитель) операнды перемножаются, а их произведение помещается на адресу первого операнда.

Оба операнда представляют собой 32-разрядные числа со знаком. Произведение может иметь длину до 64 разрядов. Поэтому оно помещается в два регистра общего назначения: в регистр, указанный в команде и регистр с номером на единицу больше. Так как произведение замещает множимое, то регистр общего назначения для множимого должен иметь четный номер, а само множимое берется из нечетного регистра.

Знак произведения определяется алгебраически в соответствии со знаками операндов.

После выборки множитель находится на регистре *P4*, а множимое — на регистре *P1*, код операции — в *РАДЗУ*.

Произведение размещается на регистрах *P3* и *P4*.

Перед началом операции производится анализ операндов на нуль. При наличии операнда, равного нулю, операция заканчивается, а результату присваивается нулевое значение.

В ЭВМ *ЕС-1030* используется метод одновременного умножения множимого на два разряда множителя, начиная с младших разрядов, с накоплением суммы частичных произведений.

В команде предусмотрено 16 сдвигов множителя в *P4*. Количество сдвигов хранится в счетчике *РС2*.

Для анализа двух цифр множителя используются разряды регистра *P4* [30, 31]. При этом могут иметь место следующие комбинации: 00, 01, 10, 11. В зависимости от этих комбинаций выполняются различные действия над суммой частичных произведений и множимым.

Два очередных разряда множителя — 00.

Сумма частичных произведений сдвигается вправо на два разряда.

Два очередных разряда множителя — 01.

К сумме частичных произведений прибавляется множимое и результат сдвигается вправо на два разряда.

Два очередных разряда множителя — 10.

Сумма частичных произведений сдвигается вправо на один разряд, после чего прибавляется множимое и результат сдвигается вправо на один разряд.

Два очередных разряда множителя — 11.

При такой комбинации к сумме частичных произведений необходимо прибавить множимое, умноженное на 3. Но в машине удобнее выполнять умножение на числа, кратные целой степени

двух. Поэтому учитывая, что $3=4-1$, сначала из суммы частичных произведений вычитают множимое, затем результат сдвигают вправо на два разряда, что соответствует сдвигу множимого на два разряда влево, т.е. умножение множимого на 4, что фиксируется в триггере вариантов, установкой его в единицу.

Состояние триггера вариантов учитывается при анализе очередных двух разрядов множителя.

Перед началом выполнения операции умножения проверяются два старших разряда множимого. Если первый разряд множимого равен нулю, то умножение выполняется по описанному алгоритму. Если первый разряд множимого единица, то множимое сдвигается на один разряд вправо с сохранением младшего разряда множимого в триггере Т40. После этого производится умножение.

Если сдвига множимого вправо не было, то произведение записывается по адресу первого операнда. Если сдвиг вправо множимого был, то произведение сдвигается влево на один разряд и, если триггер Т40=1, суммируется со множимым (с учетом знака множителя). Если Т40=0, то суммирования нет и сразу производится запись произведения по адресу первого операнда.

Команды деления DR (1D), D (5D).

Операции деления выполняются командами форматов *RR* и *RX*.

По команде деления первый операнд (делимое) делится на второй операнд (делитель) и частное с остатком помещается по адресу делимого.

Делимое представляет собой двойное слово со знаком и располагается в двух соседних регистрах с четным и нечетным номерами. Делитель представляет собой целое со знаком длиной в слово. Знак частного определяется по алгебраическим правилам. Остаток имеет знак делимого.

После выборки делитель находится в регистре Р1, а делимое — в регистре Р3 (старшее слово) и Р4 (младшее слово). Знаки операндов находятся в триггерах знака ТЗН1 (знак делимого) и ТЗН2 (знак делителя).

Перед выполнением операции операнды анализируются на равенство нулю. Если делитель равен нулю, то операция не выполняется и фиксируется прерывание. Если делитель не равен нулю, а делимое равно нулю, то по адресу первого операнда записывается частное, равное нулю, и остаток, равный нулю.

Если делимое и делитель отличны от нуля, то выполняется операция деления по методу без восстановления остатка.

Перед выполнением операции деления производится анализ старшего разряда целой части делимого. Если он значащий, то возникает прерывание программы по переполнению. Если переполнение не может быть, то деление выполняется.

Чтобы получить 32 разряда частного предусмотрено 32 цикла деления. Счет циклов деления выполняется в РС1.

После получения 32 разрядов частного и остатка в регистрах P4 и P3 соответственно выполняется запись частного и остатка по адресу первого операнда. Остаток записывается по четному адресу, а частное — по нечетному.

Логические команды.

Логические команды предназначены для логической обработки данных. Операнды этих команд представляют собой группу байтов. Они могут входить в состав команды, могут находиться в оперативной памяти или регистрах общего назначения. Операнды могут иметь переменную длину до 256 байтов или фиксированную длину в слово, двойное слово.

Логические команды могут быть любого формата (таблица 6.4.2). Логическими командами выполняются операции пере-

Таблица 6.4.2

Формат команды Название операций	<i>RR</i>		<i>RX</i>		<i>SJ</i>		<i>SS</i>	
	Код операции		Код операции		Код операции		Код операции	
	мнемо- ниче- ский	машин- ный	мнемо- ниче- ский	машин- ный	мнемо- ниче- ский	машин- ный	мнемо- ниче- ский	машин- ный
Логическое умноже- ние И	<i>NR</i>	14	<i>N</i>	54	<i>NJ</i>	94		
Логическое сложе- ние ИЛИ	<i>OR</i>	16	0	56	<i>OJ</i>	96		
Поразрядное сложе- ние	<i>XR</i>	17	<i>X</i>	57	<i>XJ</i>	97		
Сравнение кодов	<i>CLR</i>	15	<i>CL</i>	55	<i>CLT</i>	95	<i>CLC</i>	<i>D5</i>

сылки логического умножения и логического сложения, поразрядного суммирования, перекодировки, редактирования, сравнения и сдвига.

При выполнении большинства логических команд устанавливается признак результата. Могут возникать прерывания по следующим причинам: защита памяти, спецификация, адресация и данные.

Выполнение команд начинается с выборки команды из оперативной памяти и выборки операндов. После выборки один из операндов размещается в регистре P1, а другой в регистре P3 (при коротких операндах).

Ниже рассматривается методика выполнения некоторых операций.

Группа команд «Логическое умножение» NR, N, NJ.

Эти команды выполняют операцию поразрядного логического умножения операндов по следующему правилу:

$$\begin{aligned}0 \wedge 0 &= 0 \\0 \wedge 1 &= 0 \\1 \wedge 0 &= 0 \\1 \wedge 1 &= 1.\end{aligned}$$

Группа команд «Логическое сложение» OR, O, OJ.

Эти команды выполняют операцию поразрядного логического сложения операндов по следующему правилу:

$$\begin{aligned}0 \vee 0 &= 0 \\0 \vee 1 &= 1 \\1 \vee 0 &= 1 \\1 \vee 1 &= 1.\end{aligned}$$

Группа команд «Поразрядное сложение» XR, X, XJ.

Эти команды выполняют операцию сложения по модулю два по следующему правилу:

$$\begin{aligned}0 \oplus 0 &= 0 \\0 \oplus 1 &= 1 \\1 \oplus 0 &= 1 \\1 \oplus 1 &= 0.\end{aligned}$$

Все команды обрабатывают операнды побайтно в узле обработки байтов, для чего операнды из регистров Р1 и Р3 побайтно передаются на регистры РС и РД соответственно.

С выходов регистров РС и РД байты поступают на входы комбинационной схемы узла обработки байтов. Полученный байт результата передается в регистр Р7 и через регистр Р5 записывается в разряды 0—7 регистра Р3. Аналогично получают остальные байты результата и содержимое Р3 записывается по адресу первого операнда. Одновременно выполняется анализ результата с целью выработки кода условия.

Примеры:

$$\begin{array}{r}1. \quad \begin{array}{r} \vee \quad \text{РС } 0 \ 011 \ 0 \ 101 \\ \quad \text{РД } 1 \ 001 \ 1 \ 011 \\ \hline \text{Р7 } 0 \ 001 \ 0 \ 001 \end{array}\end{array}$$

$$\text{РС} \wedge \text{РД} = 0001 \ 0001$$

$$\begin{array}{r}2. \quad \begin{array}{r} \vee \quad \text{РС } 0011 \ 0101 \\ \quad \text{РД } 1001 \ 1011 \\ \hline \text{Р7 } 1011 \ 1111 \end{array}\end{array}$$

$$\text{РС} \vee \text{РД} = 10111111$$

3.

$$\begin{array}{r} \oplus \text{ РС } 0011\ 0101 \\ \oplus \text{ РД } 1001\ 1011 \\ \hline \text{Р7 } 1010\ 1110 \end{array}$$

$$\text{РС} \oplus \text{РД} = 1010\ 1110$$

Группа команд «Сравнение кодов» CLR, CL, CLT.

Операция сравнения осуществляет поразрядное сравнение двух операндов равной длины и устанавливает соответствующий код условия. Операнды рассматриваются как текстовая или буквенно-цифровая информация.

Сравниваемые операнды после их выборки из ОП размещаются в регистрах РЗ и Р1. Сравнение производится побайтно в схемах сравнения узла обработки байтов. При обнаружении неравенства операция заканчивается установлением кода условия «1» — если первый операнд меньше второго, «2» — если первый операнд больше второго.

В случае равенства первых байтов операндов сравниваются следующие байты. Если все байты операндов равны, то вырабатывается код условия, равный нулю, что соответствует равенству операндов.

Десятичные команды.

Команды десятичной арифметики выполняют арифметические операции над десятичными операндами. Все десятичные команды имеют формат SS. Команды десятичной арифметики выполняют следующие операции: сложение, вычитание, умножение, деление, сравнение и преобразование формата данных. Команды выполняют операции над данными переменной длины от 1 до 16 байтов. Адрес всегда задается самым левым байтом области, в которой находится операнд. Результат операций помещают в память на место первого операнда.

Десятичные команды обрабатывают данные только в упакованном виде, кроме команды РАСК (УПАКОВАТЬ), у которой один операнд должен быть десятичным числом с зоной. Операнды могут начинаться с любого байта.

При выполнении команд сложения, вычитания и сравнения устанавливается признак результата. Для операции сравнения признак результата 0, 1, 2 указывает на то, что первый операнд соответственно равен, меньше или больше второго. Для остальных операций признак результата 0, 1, 2 или 3 устанавливается в тех случаях, когда результат операции соответственно равен нулю, меньше нуля, больше нуля или произошло переполнение.

При выполнении команд десятичной арифметики могут возникать программные прерывания. Прерывания программы вызывают следующие причины: защита памяти, адресация, спецификация, данные, десятичное переполнение и некорректность десятичного деления.

Команда сложение десятичное AP (FA).

Первый операнд складывается со вторым, результат помещается по адресу первого операнда, вырабатывается код условия. Операнды обрабатываются побайтно справа налево.

Адреса правых байтов первого и второго операндов после выборки из ОП помещаются соответственно в регистры Р4 и Р2, два младших разряда адресов операндов, определяющих положение байтов в слове, размещаются в регистрах РС3 и РС4. Число байтов в первом операнде записывается в 0—3 разрядах, а число байтов во втором операнде — в 4—7 разрядах регистра РС1. Первый операнд помещается в регистр Р3, а второй — в регистр Р1.

По анализу РС4 и РС3 соответствующие байты операндов из регистров Р1 и Р3 пересылаются в регистры РС и РД соответственно и суммируются в сумматоре десятичном. Перенос из старшего разряда сумматора запоминается в триггере Т30 и учитывается при сложении следующих байтов операндов.

Знак первого операнда запоминается в триггере Т10, а второй — в триггере Т20.

По анализу состояния этих триггеров триггер вариантов (ТВАР) устанавливается в нуль, если знаки разные, или в единицу, если знаки одинаковые.

При ТВАР=1 сложение операндов производится в прямом коде, и второй операнд из регистра РС на вход сумматора десятичного подается с избытком 6 в каждой тетраде, отображающей десятичную цифру.

При ТВАР=0 производится сложение первого операнда с дополнительным кодом второго. Второй операнд из регистра РС поступает на сумматор десятичный инверсно, т.е. в обратном коде, а в младший разряд самого правого байта добавляется +1.

Использование обратного кода второго операнда обеспечивает занесение в каждую тетраду избытка 6, используемого при организации переносов в десятичных разрядах.

Полученный байт результата корректируется и передается по анализу РС3 через Р5 в соответствующий байт регистра Р3. Корректировка выполняется схемно на выходе сумматора десятичного.

Если из данной десятичной цифры (из тетрады) нет переноса, к ней прибавляется число 10 (двоичное число 1010) без распространения переноса из старшего разряда тетрады. Это действие соответствует вычитанию избытка 6, занесенного перед началом операции. Полученному результату присваивается знак первого операнда. Результат операции, накапливаемый в регистре Р3, по анализу содержимого регистра РС6, в котором подсчитывается количество обработанных байтов, записывают в ОП. После суммирования одного байта длины операндов и адрес первого операнда изменяются на единицу. Адрес второго операнда уменьшается на четыре после выборки очередного слова из ОП.

Если результат получается отрицательный, то он после записи выбирается из ОП, переводится в прямой код и снова записывается в ОП.

Команда десятичное вычитание SR (FB).

По этой команде второй операнд вычитается из первого, результат записывается по адресу первого операнда.

Вычитание выполняется по той же микропрограмме, что и сложение, но перед выполнением операции знак второго операнда инвертируется.

Команда десятичное сравнение CP (F9).

Первый операнд сравнивается со вторым и вырабатывается код условия.

Адрес крайнего правого байта первого операнда помещается в регистр P4, а второго операнда — в регистр P2, разряды, определяющие положение байта в слове, помещаются соответственно в регистры PC3 и PC4.

Число байтов L1 и L2 помещаются в PC1 [0/3] и PC1 [4/7]. Слово первого операнда после выборки из ОП помещается в регистр P3, а второго — в регистр P1. Сравнение выполняется в сумматоре десятичном.

После обработки каждого байта из содержимого PC3 и PC4 вычитается единица, а после обработки каждого слова из адресов вычитается 4 и из ОП выбирается следующее слово.

Операция заканчивается при сравнении всех байтов операндов. Если результат равен нулю, код условия устанавливается нуль. Если первый операнд больше второго, код условия устанавливается 2, если первый операнд меньше второго, код условия устанавливается 1.

Результат в оперативную память не записывается.

Команда умножение десятичное MP (C).

Умножение производится командой формата SS.

Умножение любого числа Z на любую цифру можно представить следующим образом:

$$\begin{aligned} z \times 1 &= z, \\ z \times 2 &= z + z = 2z, \\ z \times 3 &= 2z + z = 3z, \\ z \times 4 &= 3z + z = 4z, \\ z \times 5 &= 4z + z = 5z, \\ z \times 6 &= 4z + 2z = 6z, \\ z \times 7 &= 4z + 2z + z = 7z, \\ z \times 8 &= 4z + 4z = 8z, \\ z \times 9 &= 4z + 4z + z = 9z. \end{aligned}$$

Умножение числа на два выполняется сдвигом этого числа на один разряд влево.

Пример:

$$\begin{array}{r} \times 97 \\ 36 \\ \hline + 582 \\ 291 \\ \hline 3492 \end{array}$$

Запишем множимое 97 в двоичном эквиваленте

0000 1001 0111.

Удвоим это число

0001 0010 1110.

Для корректировки результата в тетрады, из которых был перенос или цифра больше 9, добавим 6

$$\begin{array}{r} 0001\ 0010\ 1110 \\ + 0000\ 0110\ 0110 \\ \hline 0001\ 1001\ 0100 \\ 1\quad\quad 9\quad 4 \end{array}$$

Полученное число 194 — это удвоенное множимое. Выполняя аналогичные действия, получим учетверенное множимое:

$$\begin{array}{r} 0001\ 1001\ 0100 \\ 0011\ 0010\ 1000 \\ \quad 0110 \\ \hline 0011\ 1000\ 1000 \\ 3\quad\quad 8\quad\quad 8 \end{array}$$

Полученные удвоенное и учетверенное множимое запоминаться.

Для того, чтобы получить умножение множимого (97) на младшую цифру множителя (6), произведем сложение полученных чисел 388 и 194.

Для выполнения сложения производим корректировку первого слагаемого:

$$\begin{array}{r} 0011\ 1000\ 1000 \\ + 0110\ 0110\ 0110 \\ \hline 1001\ 1110\ 1110 \end{array} \text{ — скорректированное число 388.}$$

$$\begin{array}{r} 1001\ 1110\ 1110 \\ + 0001\ 1001\ 0100 \\ \hline 1011\ 1000\ 0010 \end{array}$$

Произведем корректировку полученной суммы с запретом переноса

$$\begin{array}{r}
 1011 \ 1000 \ 0010 \\
 1010 \\
 \hline
 0101 \ 1000 \ 0010 \\
 5 \quad 8 \quad 2
 \end{array}$$

Число 582 — это частичное произведение (97×6).

Аналогично выполняя умножение множимого на цифру 3, получим второе частичное произведение 291 (97×3).

Производя сложение сдвинутого вправо на 4 разряда первого частичного произведения со вторым, получаем результат:

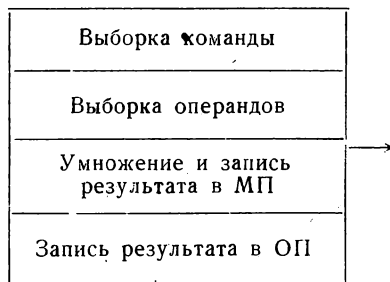
$$\begin{array}{r}
 0000 \ 0101 \ 1000 \\
 + 0110 \ 0110 \ 0110 \\
 \hline
 0110 \ 1011 \ 1110 \text{ — корректировка первого слагаемого,} \\
 \quad 0110 \ 1011 \ 1110 \\
 + \quad 0010 \ 1001 \ 0001 \\
 \hline
 \quad 1001 \ 0100 \ 1111 \\
 + 1001 \ 0100 \ 1111 \\
 + 1010 \quad \quad 1010 \\
 \hline
 0011 \ 0100 \ 1001 \text{ — корректировка результата.}
 \end{array}$$

К полученному результату «пристраиваем» младшую цифру первого частичного произведения

$$\begin{array}{r}
 0011 \ 0100 \ 1001 \ 0010 \\
 3 \quad 4 \quad 9 \quad 2
 \end{array}$$

Знак произведения определяют алгебраически.

Процесс десятичного умножения в машине можно разделить на несколько стадий.



После выборки команды из ОП выбираются операнды и помещаются в МП по определенным адресам. Операнды при этом выравниваются по правым границам. Старшее слово выбирается

из ОП первым и размещается в регистре общего назначения номер 32, младшее слово размещается в РОН номер 33. Множимое размещается в регистрах общего назначения как показано в таблице 6.4.3.

Таблица 6.4.3

Номер РОН	Содержимое
36 39 42 45	Множимое
37 40 43 46	Удвоенное множимое
38 41 44 47	Учетверенное множимое

Перед выполнением умножения проверяются условия: 1) $L1 > L2$, 2) $L2 \leq 7$.

Множимое должно содержать столько нулей в старших разрядах, сколько байтов во множителе.

Умножение выполняется в прямом коде способом последовательного прибавления множимого столько раз, сколько единиц содержится в очередной цифре множителя.

Анализ ведется начиная с младших цифр. За цикл обрабатывается одна цифра множителя.

Слово множителя считывается из 32 регистра общего назначения в регистр Р5 и передается в регистр Р4. В зависимости от значения младшей цифры множителя из МП считывается в регистр Р3 или учетверенное множимое, или удвоенное множимое, или множимое. Считанное множимое складывается с частичным произведением, находящимся в регистре Р2, и корректируется.

Произведение накапливается в регистре Р4. В конце операции произведение записывается в МП по адресам 51—54. Адрес первого операнда, записанный ранее в МП по адресу 48, считывается в регистр Р5 и пересылается в регистр Р2. Результат по одному слову считывается из МП и записывается в регистр Р3. Из регистра Р3 результат записывается в ОП.

Деление десятичное ДР (ФД).

Делимое (первый операнд) делится на делитель. Частное и остаток записываются по адресу делимого.

Деление выполняется по способу с восстановлением остатка. В результате деления получаем частное и остаток со своими знаками. Знак частного определяется алгебраически после сравнения знаков операндов в регистрах РС и РД и запоминается в триггере Т10.

Остаток имеет знак делимого, который сохраняется в триггере Т20.

Обязательно должны выполняться следующие условия:

1. Старшая цифра делимого должна быть нулем.

2. Делитель не должен быть больше двух слов.
3. Делимое должно быть больше делителя ($L1 > L2$).

Деление десятичное выполняется с предварительной выборкой операндов в регистры общего назначения и выравниванием их по левым границам (т. е. по старшим разрядам).

Частное и остаток в местной памяти помещаются по адресу первого операнда. Остаток находится в старших разрядах, а частное — в младших. После окончания деления производится обмен местами частного и остатка и запись их в оперативную память.

Процесс деления можно разделить на следующие стадии:

Выборка команды формата SS стандартным образом
Выборка операндов в МП. Их выравнивание и проверка условий
Деление Запись частного и остатка в МП
Запись результата из МП в ОП

Деление выполняется начиная со старших разрядов, поэтому операнды выравниваются по левым границам. Первым из ОП выбирается делимое.

Адрес выбираемого операнда размещается в регистре P2. Операнд из ОП выбирается по одному слову. Делимое после выборки из ОП запоминается в МП по адресам 32—35. Запись в МП начинается с адреса 32. Для записи в следующий адрес к РСА прибавляется единица.

После выборки делимого выбирается делитель и записывается в МП по адресам 36 и 37.

Деление выполняется методом последовательного вычитания с восстановлением остатка.

Команды с плавающей запятой.

Команды с плавающей запятой выполняют операции над числами с плавающей запятой, которые могут быть двух форматов фиксированной длины: короткого (длиной в слово) и длинного (длиной в двойное слово). Эти форматы имеют одинаковые порядки, но отличаются длиной дробной части (мантиссы).

Команды с плавающей запятой выполняют операции загрузки, сложения, вычитания, сравнения, умножения, деления, записи в память. Они могут быть двух форматов RR и RX.

При выполнении этих команд, кроме оперативной памяти, используются четыре регистра с плавающей запятой, которым присвоены номера 0, 2, 4, 6.

При выполнении большинства операций с плавающей запятой устанавливается признак результата 0, 1, 2, который указывает на то, что результат соответственно равен нулю, меньше нуля и больше нуля. Признак результата, равный 3, указывает на то, что произошло переполнение порядка.

При выполнении команд с плавающей запятой могут возникать прерывания программы по следующим причинам: защита памяти, адресация, спецификация, переполнение порядка, исчезновение порядка, потеря значимости, некорректность деления.

Команды сложения с нормализацией ADR (2A), AD (6A), AER (3A), AE (7A).

При выполнении команд сложения операнды складываются и нормализованная сумма помещается по адресу первого операнда.

Сложение выполняется в три этапа:

1. Выравнивание характеристик операндов.
2. Сложение мантисс.
3. Нормализация результата.

При сложении коротких операндов (операции *AER*, *AE*) после выборки первый операнд помещается в регистре P3, а второй — в P1. При сложении длинных операндов (операции *ADR*, *AD*) первый операнд помещается в регистры P3 и P4, а второй операнд — в регистры P1 и P2. Характеристики операндов сравниваются в узле обработки байтов, для чего они размещаются в регистрах PД и РС. Характеристика первого операнда, кроме того, размещается в счетчике PC1.

Если характеристики не равны, они выравниваются. Для этого мантисса операнда с меньшей характеристикой сдвигается вправо на один разряд и к содержимому PC1, в котором хранится порядок, прибавляется единица, если сдвигался первый операнд, или вычитается единица, если сдвигался второй операнд. Затем содержимое PC1 вновь посылается в PД и сравнивается с содержимым РС. Сдвиги продолжают до тех пор, пока содержимое PД и РС не станет одинаковым. В качестве порядка результата берется больший порядок, который помещается в PC1.

Максимальное количество сдвигов при выравнивании порядков коротких операндов равно семи, а длинных операндов — пятнадцати. Если после максимального числа сдвигов порядки не выравнены, то операция заканчивается. В этом случае в качестве результата берется операнд с большим порядком. После установления признака результата производится запись результата в местную память.

Счет количества сдвигов ведется в регистре счета PC2. После выравнивания характеристик производится анализ знаков операндов, которые находятся в TЗН1 и TЗН2 и затем сложение мантисс. Если знаки одинаковые, мантиссы складываются как положительные числа. Если знаки операндов разные, то операнд с отрицательным знаком переводится в дополнительный код.

Если при сложении мантисс возникает переполнение, то промежуточная сумма сдвигается вправо на один разряд, а характеристика увеличивается на единицу.

Если это вызывает переполнение характеристики, о чем свидетельствует $PC1[0]=1$, то вырабатывается код переполнения порядка и происходит программное прерывание.

Помежуточная сумма при сложении коротких операндов состоит из семи шестнадцатеричных цифр и возможного переноса, а при сложении длинных операндов — из пятнадцати шестнадцатеричных цифр и возможного переноса. Младшая цифра появляется при сдвиге одной из мантисс вправо. Она хранится в регистре Р6.

После сложения промежуточная сумма сдвигается влево до получения нормализованной мантиссы, освобождающиеся при этом младшие разряды заполняются нулями, характеристика уменьшается на количество сдвигов. Так как нормализация производится для шестнадцатеричных цифр, то три старших двоичных разряда нормализованной суммы могут быть нулями. Если складываются равные числа с разными знаками, то результат операции будет нулем (потеря значимости), что может вызывать прерывание программы. Знак суммы определяется алгебраически. Знак суммы с нулевой мантиссой всегда положителен.

Команды вычитания с нормализацией SDR (2B), SD (6B), SE (7B), SER (3B).

При выполнении команд вычитания второй операнд вычитается из первого операнда и нормализованная разность помещается по адресу первого операнда. Команды вычитания выполняются по той же микропрограмме, что и команды сложения, но перед выполнением знак второго операнда изменяется на противоположный ($T3H2 := -P1[0]$). Знак разности определяется так же, как и при сложении. Знак разности с нулевой конечной мантиссой всегда положителен.

6.5. Блок синхронизации

Блок синхронизации — БСИ предназначен для выработки и распределения синхроимпульсов по блокам процессора.

Процессор работает с тактовой частотой 1,5 МГц. Под тактом работы процессора понимается время, в течение которого считывается и выполняется одна микрокоманда. Один такт работы процессора, равный $T \approx 750$ нс, делится на два полутакта. В соответствии с этим блоком БСИ вырабатываются две серии: (C1 и C2) синхроимпульсов, сдвинутых друг относительно друга. Серия C1 соответствует первому полутакту, а C2 — второму.

Исполнительные импульсы элементарных операций формируются на выходах дешифраторов полей микрокоманды. Момент появления соответствующего импульса (импульсов) определяется синхроимпульсом C1 или C2, стробирующим данный дешифратор.

В состав блока синхронизации входят:

- узел формирования тактовых синхроимпульсов;
- узел распределения тактовых синхроимпульсов.

Узел формирования тактовых синхроимпульсов (генератор импульсов)

Узел формирования тактовых синхроимпульсов смонтирован в специальном типовом элементе замены (ТЭЗ 2030/0101) территориально расположенном в стойке каналов по конструктивному адресу 02С 2А—09. Смысловое значение шифра адреса следующее:

- 02 — стойка каналов;
- С — рама «С» стойки каналов;
- 2А — координата панели на раме;
- 09 — порядковый номер или место ТЭЗа на панели.

Генератор собран на типовых микросхемах 1ЛР551 (Т150) и транзисторах 2Т 306В. В его состав входят также линии задержки, в качестве которых используются отрезки коаксиального кабеля, и другие вспомогательные элементы (резисторы, конденсаторы, диоды).

Упрощенная схема генератора импульсов и временная диаграмма, иллюстрирующая формирование двух серий синхроимпульсов С1 и С2, изображены на рис. 6.5.1 и 6.5.2 соответственно.

Принцип работы

При включении питания за счет наличия обратной связи (с выхода усилителя УС1 через линии задержки Л2 и Л1 и элемент У1 на вход УС1) генератор возбуждается и генерирует прямоугольные импульсы с частотой следования, определяемой, в основном, параметрами линий задержки Л1 и Л2.

В элементе У2, представляющем собой схему 2И—ИЛИ—НЕ, осуществляется сложение двух последовательностей импульсов. Эти импульсы, подаваемые на входы 1 и 2, имеют скважность 0,5 и сдвинуты друг относительно друга за счет задержки в линиях Л1 и Л2 (см. временную диаграмму рис. 6.5.2). В результате сложения на выходе микросхемы У2 и усилителя УС2 образуются синхроимпульсы необходимой скважности.

Сформированные тактовые синхроимпульсы через элементы У3 и УС3 подаются на микросхему У4 и через линию задержки Л3 на микросхему У5. С выходов этих микросхем снимаются две серии синхроимпульсов С1 и С2, сдвинутые на величину задержки в линии Л3.

Временные параметры генерируемых синхроимпульсов можно менять, изменяя задержку в линиях задержки Л1, Л2, Л3. Для этой цели подбирается путем перепаек необходимая длина коаксиальных кабелей, пять отрезков которых находятся в блоке пульта управления.

Внешние синхросигналы с названием «ССВ СИСТ ЭВМ» (рис. 6.5.1) подаются на контакт 36 только в том случае, если вы-

числительная машина входит в состав многомашинного комплекса и синхронизация всего комплекса осуществляется от генератора импульсов второй ЭВМ. В этом случае обратная связь в генераторе импульсов первой ЭВМ разрывается, т. е. элементы У1, У2, УС1, УС2 выключаются (выпадают подходящие к ним проходы).

К контактам 04, 05, 15, 27, 37 разъема ТЭЗ 2030/0101 (констр. адр. 02С 2А—09) может быть подключен осциллограф для измерения временных параметров синхроимпульсов, вырабатываемых генератором.

Узел распределения синхроимпульсов

Синхроимпульсы серии С1 и С2, поступающие в стойку процессора из стойки каналов, усиливаются и распределяются узлом распределения по блокам процессора.

Узел распределения состоит из 13 однотипных ТЭЗ ЕС-2030/0103, размещенных в различных панелях рам А, В, С процессора, в местах, приближенных к потребителям синхроимпульсов.

Часть схемы ТЭЗа ЕС-2030/0103 приведена на рис. 6.5.3.

На входы схемы параллельно на контакты 36, 7 (14, 18, 19, 24, 37, 30) поступают синхроимпульсы серий С1 и С2. Усиленные синхроимпульсы восстановленной формы снимаются с выходов схемы — контакты 6, 25 (8, 4, 22, 20, 23, 26). В круглых скобках указаны номера контактов отсутствующей на рис. 6.5.3 части принципиальной схемы ТЭЗа.

Резисторы и диод, подключенные через конденсатор к эмиттеру микросхемы (Т150), образуют цепочку, которая улучшает форму выходного сигнала.

Для синхронизации работы блоков процессора используются синхроимпульсы, снимаемые с выходов ТЭЗов, расположенных по конструктивным адресам:

1. В раме „А“ 1С—12, 1В—14, 2А—05,
2С—0С, 1И—09, 2Н—20.
2. В раме „С“ 2В213, 2С—20, 1Д—10,
2Е—10, 1В—13.
3. В раме „В“ 1Е—08, 2Н—03.

Как правило, в каждом распределительном ТЭЗе ЕС-2030/0103 используется для синхронизации от 2 до 7 выходов, остальные (6—1) не используются и являются резервными.

Схемы ТЭЗ ЕС-2030/0101 и ЕС-2030/0103 приведены на рис. 6.5.4 и 6.5.5.

6.6. Блок микропрограммного управления

Назначение и состав БМУ

В состав устройства микропрограммного управления процессора входит долговременное запоминающее устройство (ДЗУ) со

своими информационным (РИДЗУ) и адресным (РАДЗУ) регистрами, блок микропрограммного управления (БМУ) и дешифраторы элементарных операций, расположенные в блоках процессора.

ДЗУ служит для хранения микропрограмм команд, микропрограмм пультовых операций и микропрограмм процедур прерывания и диагностики.

БМУ предназначен для полного или частичного управления блоками процессора, которое он осуществляет, выполняя следующую последовательность действий:

1. Формирование адреса очередной микрокоманды в регистре РАДЗУ.

2. Считывание микрокоманд из накопителя ДЗУ в регистр РИДЗУ.

3. Дешифрация отдельных частей микрокоманды и выработка управляющих сигналов для выполнения элементарных операций в блоках процессора.

Микрокоманды из ДЗУ считываются через каждый такт работы машины, который равен $T \approx 670$ нс, однако элементарные операции в процессоре выполняются с вдвое большей частотой, так как микрокоманда содержит в себе закодированные элементарные операции, выполняемые в полутаках $C1$ и $C2$.

В состав блока БМУ (рис. 6.6.1) входят регистр информации дополнительный (РИДЗУД), дешифраторы элементарных операций (ДШЭЛ ОП-1), (ДШЭЛ ОП-2), дешифратор для анализов (ДШ АН) и схема формирования адреса в 8—11 разрядах РАДЗУ (ВРА [8], ВРА [9], ВРА [10], ВРА [11]).

Структура микрокоманды

Информация, управляющая работой цифровой ЭВМ в течение одного машинного такта, называется микрокомандой.

В ЭВМ ЕС-1030 принят способ кодирования микрокоманды, называемый способом управляющих полей, который заключается в следующем:

72 — битовая микрокоманда (рис. 6.6.2) разбивается на поля. Каждое поле определяет до 2^n элементарных операций (n — число бит в поле). За один машинный такт выполняется только одна элементарная операция из данного поля.

Последовательность микрокоманд, реализующих какую-либо целевую функцию (например, сброс системы, выполнение команды и др.), называется микропрограммой. При организации микрокоманд в микропрограммы каждая микрокоманда должна определять адрес следующей. В ЕС-1030 принят произвольный способ организации адресов, заключающийся в том, что каждая микрокоманда содержит поля, определяющие условия формирования адреса следующей микрокоманды.

72-разрядный регистр информации (РИДЗУ), в который помещается микрокоманда, разделен на поля следующим образом (рис. 6.6.2.):

0		1	2	7 8			13 14		19 20			25 26		31
К		У		Г			В		Б			А		Н
32	33	37	38	42	43	46	47	48	51	52	55	56	63	
К		Д		Ж		И		С	М		Л		адрес следующей микрокоманды	
64	65	69	70	71										
К		Е		Р		П								

Рис. 6.6.2

Регистр РИДЗУ имеет три контрольных разряда (обозначенных на рис. 6.6.2 буквой К):

РИДЗУ [0] — для контроля РИДЗУ [1/31],
 РИДЗУ [32] — для контроля РИДЗУ [33/63],
 РИДЗУ [64] — для контроля РИДЗУ [65/71].

В формировании сигналов элементарных операций принимают участие все поля, кроме полей «С», «У», «К» и поля, выделенного для указания адреса следующей микрокоманды. Всего с помощью этих полей можно закодировать 612 различных элементарных операций. В течение одного такта могут выполняться до 13 элементарных операций. Это определяется числом используемых в микрокоманде полей.

Дешифрация полей микрокоманды

С помощью указанных на рис. 6.6.2 полей микрокоманда может выполнять следующие четыре основных действия:

1. Формирование адреса следующей микрокоманды.
2. Выполнение операций в блоке БАЛ.
3. Обращение к ОП и МП.
4. Установку управляющих триггеров и регистров.

„1“ в поле „С“ (47 разряд) означает, что в полях „М“ и „Л“ записана константа, которую нужно передать в какой-либо регистр. При этом дешифраторы элементарных операций этих полей блокируются, а в каком-либо другом поле записывается элементарная операция приема константы в регистр. Например, в микрокоманде записано: С1, Л6, Ж1 (в поля „С“ и „Ж“ занесены „1“, а в поле „Л“ — шестерка). Это означает, что константу 6 из поля „Л“ нужно передать в регистр РАДЗУ [0/3].

Дешифратор поля „Ж“ для кода „1“ выдает управляющий сигнал РАДЗУ: = „КОНСТ“, который обеспечивает передачу константы из поля „Л“ в 0—3 разряды регистра РАДЗУ.

Элементарные операции, записанные (закодированные) в полях „В“, „Г“, „Е“, дешифрируются двояко в зависимости от состояния поля „У“. Для каждого кода этих полей имеется два элемента дешифратора: один выдает управляющий сигнал при $У=0$, другой — при $У=1$.

В случаях, когда $У=1$, выполняются элементарные операции диагностики неисправностей машины.

Формирование адреса следующей микрокоманды

В поле, занимающем разряды с 56 по 63, записывается адрес следующей микрокоманды. После считывания микрокоманды из ДЗУ этот адрес в полутакте С1 передается в регистр РАДЗУ [4/11]. Содержимое РАДЗУ [0/3] (номер страницы ДЗУ) изменяется с помощью элементарной операции „РАДЗУ := КОНСТ“, о которой говорилось выше.

Для осуществления условного перехода при выполнении микрокоманд 8—11 разряды РАДЗУ связаны с дешифраторами полей Г, В, Б, А т. е.

РАДЗУ [8]	связан с дешифратором поля „Г“
РАДЗУ [9]	— „—“ — „В“
РАДЗУ [10]	— „—“ — „Б“
РАДЗУ [11]	— „—“ — „А“

Если в каком-либо из этих полей микрокоманды стоит код элементарной операции анализа (проверка состояний триггеров или регистров), то соответствующий разряд регистра РАДЗУ устанавливается по результату анализа и не принимает информацию из соответствующего ему разряда регистра РИДЗУ.

Например, после считывания команды формата *RR* из ОП необходимо проанализировать в какой половине прочитанного слова находятся два байта команды (команда формата *RR* имеет длину в два байта). Для этого проверяется состояние 62-го разряда регистра РССП. Если РССП[62]=0, то команда начинается с начала слова, а если РССП[62]=1, то команда занимает последние два байта в слове, т. е. начинается с полуслова и в этом случае необходимо перейти на другую ветвь микропрограммы ВЫБОРКА.

Для выполнения анализа на равенство „1“ 62-го разряда регистра РССП в поле „А“ микрокоманды должен находиться код 61 (А61). При выполнении условия — РССП[62]=1 в 11 разряд регистра адреса РАДЗУ заносится „1“, в противном случае в 11 разряде остается „0“.

Аналогичным образом осуществляются другие проверки в соответствии с кодами полей Б, В, Г.

Если условие анализа выполняется по соответствующему коду

в поле „А“, то „1“ заносится в РАДЗУ [11],
 в поле „Б“, то „1“ заносится в РАДЗУ [10],
 в поле „В“, то „1“ заносится в РАДЗУ [9],
 в поле „Г“, то „1“ заносится в РАДЗУ [8].

Адрес очередной микрокоманды может увеличиваться на 1, 2, 4, 8 или на их сумму в любой комбинации.

Например, в поле 56—63 указан адрес очередной микрокоманды 2.7.0. Тогда при выполнении соответствующих условий анализа только

по полю „А“	адрес станет равным	2.7.1,
по полю „Б“	адрес	— „— 2.7.2,
по полю „В“	адрес	— „— 2.7.4,
по полю „Г“	адрес	— „— 2.7.8.

Если ни одно условие анализа не выполняется, адрес очередной микрокоманды останется 2.7.0.

При выполнении условий анализов сразу в нескольких полях А, Б, В, Г можно получить ветвления в микропрограмме до 16 направлений (адресов). Такими адресами для нашего примера будут:

2.7.0, 2.7.1, 2.7.2, 2.7.3, 2.7.4, 2.7.5, 2.7.6, 2.7.7, 2.7.8, 2.7.9, 2.7.10, 2.7.11, 2.7.12, 2.7.13, 2.7.14, 2.7.15.

Поля микрокоманды А, Б, В, Г могут использоваться как для проверки условий выполнения анализов, так и для выработки обычных сигналов управления. Определить назначение поля можно по хранимому в нем коду.

Если в поле „А“ код больше 15, то это код анализа;

— „—	„Б“	— „— 15,	— „— ;
— „—	„В“	— „— 47,	— „— ;
— „—	„Г“	— „— 31,	— „— .

Это необходимо иметь в виду при анализе микрокоманд при выполнении их в „одионочном“ режиме работы с пульта управления.

Поскольку такт работы машины разбит на два полутакта С1 и С2, то для выполнения элементарных операций в полутакте С2 создан дополнительный регистр, дублирующий поля В, Н, Ж, И, М, Л, Е, У, С (рис. 6.6.1).

Информация из регистра РИДЗУ поступает в регистр РИДЗУД в полутакте С1 и сохраняется до следующего приема. РИДЗУД — 36-разрядный регистр, собран на трех ТЭЗ типа Т000/0001, занимающих места 2Е15, 2Е16 и 2Е17 в раме „В“ стойки процессора.

На выходах РИДЗУ собраны дешифраторы элементарных операций, выполняющихся в полутакте С1, а на выходах регистра РИДЗУД — в полутакте С2. Эти дешифраторы элементарных операций находятся как в блоке БМУ, так и в блоках БАЛ и БУР.

Дешифраторы собраны на ТЭЗ типа 2030/0070, 2030/0071, 2031/0072 (рис. 6.6.6).

Информация в регистр адреса РАДЗУ засылается из регистров: РИДЗУ [56/63], РВО [0/7], РИОП [19/30], из регистра адреса пульта управления РАПУ [12/23] и от схем формирования фиксированных адресов (например, от схем, обеспечивающих занесение адреса от кнопок пульта управления). Одновременно с приемом адреса в регистр РАДЗУ вырабатывается сигнал „ЧТЕНИЕ“, поступающий в блок управления ДЗУ из блока БМУ.

В таблице 6.6.1 приведены все случаи передач адреса в 0—11 разряды регистра РАДЗУ (правая колонка).

Таблица указывает адреса микропрограмм, хранящихся в ДЗУ. Например, при нажатии кнопки СБРОС СИСТЕМЫ в регистр РАДЗУ занесется адрес 14.0.2 или в двоично-шестнадцатичной системе счисления:

1110 0000 0010

Это начальный адрес микропрограммы СБРОСА СИСТЕМЫ (рис. 6.9.1).

На рис. 6.6.4 изображена схема модификации адреса РАДЗУ [11], соответствующая элементу ВРА [11] структурной схемы блока микропрограммного управления (рис. 6.6.1).

Модификация 11 разряда регистра РАДЗУ будет осуществляться сигналами, снимаемыми с выходов микросхем И—ИЛИ—НЕ (IV—V) при отсутствии сигналов передачи в РАДЗУ информации из регистров РВО, РИОП и засылки в РАДЗУ фиксированных адресов (в этом случае на всех входах схемы „И“ (1) будут единичные сигналы).

При наличии сигнала анализа 62-го разряда регистра РССП („—РССП [62]=1“) модификация РАДЗУ [11] будет зависеть от состояния разряда РССП [62]. Она выполняется с помощью верхней схемы „И“ микросхемы IV. А при отсутствии сигнала анализа будет зависеть от значения разряда РИДЗУ [63] и выполняться с помощью нижней схемы „И“ (IV).

Дешифраторы полей микрокоманды и схемы модификации адреса РАДЗУ

Дешифраторы отдельных полей микрокоманды устройства микропрограммного управления установлены непосредственно в местах действия управляющих сигналов. Это сделано с целью сокращения количества управляющих проводов. Например, в случае установки дешифраторов 6-разрядного поля А непосредственно в блоке БМУ из этого блока потребовалось бы выводить $2^6=64$ провода. А при принятой методике требуется только 6 проводов.

Дешифраторы выполнены на интегральных микросхемах типа Т130 и имеют однотипную, состоящую из совокупности элементов „И“, принципиальную схему. Схема одного такого элемента изо-

бражена на рис. 6.6.3. На рисунке изображен элемент дешифратора поля А, обеспечивающий выработку управляющего сигнала для передачи содержимого регистра РВО в регистр РАДЗУ (—РАДЗУ:=РВО) при наличии в поле „А“ кода 15.

Элемент выдает управляющий сигнал (уровень „0“) только в том случае, если в поле „А“ будет код 001111 и отсутствует сигнал блокировка (т.е. на всех входах будут высокие уровни „1“).

Другие элементы дешифратора поля „А“, а также дешифраторы других полей строятся аналогично.

Сигналы блокировки дешифраторов

На входы всех дешифраторов помимо значений соответствующих полей регистров РИДЗУ или РИДЗУД поступают сигналы блокировки, вырабатываемые в процессоре. Эти сигналы называются:

БЛОК 1 — БЛОК 9 и БЛПУ ДШ ЭЛ ОП.

В таблице 6.6.2 приведены условия возникновения и назначения указанных сигналов.

Примечание:

1. Сигнал „БЛПУ ДШ ЭЛ ОП“ (блокировка пульта управления для дешифраторов элементарных операций) вырабатывается при нажатии кнопок, включении тумблеров или изменении положения переключателей пульта управления.

2. Сигнал „БЛПУ ДШ АН“ (блокировка пульта управления для дешифраторов анализов) вырабатывается в блоке пульта управления при наличии сигналов БЛПУ ЭЛ ОП $\vee$ „ЦИКЛ“ (на ПУ тумблер „ЦИКЛ“ — „ТАКТ“ включен в положение „ЦИКЛ“).

3. „ОДИН“, „ЦИКЛ“, „ТАКТ“ означают, что переключатель режимов работы и тумблер „ЦИКЛ/ТАКТ“ находятся соответственно в положениях „ОДИН“, „ЦИКЛ“ или „ТАКТ“.

4. Сигнал ПР ФИКС АДР Д (прием фиксированного адреса диагностики) вырабатывается в блоке БМУ.

5. ТЧП (триггер часов процессора) находится в блоке диагностики.

6. Нажатием на пульте управления кнопки НАЧ блокировка дешифраторов, вызванная условиями, заключенными в фигурные скобки (см. табл. 6.6.2) снимается на время $C1 + C2$, если переключатель режимов ДЗУ находится в положении „C1, C2“ или на время C1 или C2, если переключатель находится соответственно в положениях „C1“ или „C2“. (Этот режим используется при наладке машины).

7. БЛ ДШ означает, что соответствующий тумблер находится в положении блокировки дешифраторов (т.е. не в исходном положении).

Временная диаграмма работы блока микропрограммного управления изображена на рис. 6.6.5.

Таблица 6.6.1

Фикс. адреса формируемые от кнопок ПУ							РАДЗУ: = РИОП [19/30]	РАДЗУ: = адрес диагн.				РАДЗУ: = РВО [0/7]	РАДЗУ: = РАПУ	Конст. из поля Л (при С=1)	Разряды РАДЗУ [0/11]	
Вкл. пита- ния	Сброс системы	Ввод	Повто- рить РССП	Устано- вить РССП	Запись	Считы- вание		1	2	3	4					
14	14	14	14	14	14	14	[19/22]	F	F	F	F	0				Адрес с тумблерного набора ПУ
				1	2	4	[23/26]	C	C	C	C	[0/3]			РИДЗУ [56/59]	4/7
			1				[27]	1	1	0	0	[4]			РИДЗУ [60]	8
		1					[28]	1	0	1	0	[5]			РИДЗУ [61]	9
	1						[29]	0	0	0	0	[6]			РИДЗУ [62]	11
1							[30]	0	0	0	0	[7]			РИДЗУ [63]	11

Идентификатор сигнала	Условия выработки сигнала	Блок, в котором вырабатывается	Блокируемые поля (дешифраторы)	Полуэквивалентные элементов операций
Блок. 1	БЛУ ДШ. ЭЛ. ОП. $\vee$ (РИД ЗУ $[1]=0$)	БМУ, БУР	В, Г, Е [при $У=1$]	C1
Блок. 2	БЛУ ДШ. ЭЛ. ОП. $\vee$ (РИДЗУ $[1]=1$)	БАЛ, БМУ, БУР	В, Г, Е (при $У=0$)	C1
Блок. 3	БЛУ ДШ. ЭЛ. ОП. $\vee$ (РИДЗУ $[47]=1$)	БАЛ, БМУ, БУР	М, Л	C1
Блок. 4	БЛУ ДШ. ЭЛ. ОП. $\vee$ (РИДЗУД $[1]=1$)	БАЛ, БМУ, БУР	В, Г, Е (при $У=0$)	C2
Блок. 5	БЛУ ДШ. ЭЛ. ОП. $\vee$ (РИДЗУД $[1]=0$)	БУР	В, Г, Е (при $У=1$)	C2
Блок. 6	БЛУ ДШ. ЭЛ. ОП. $\vee$ (РИДЗУД $[47]=1$)	БАЛ, БМУ, БУР	М, Л	C2
Блок. 7	БЛУ ДШ. АН. $\vee$ ПР. ФИКС. АДР. Д	БМУ	А, Б (коды > 16)	C1
Блок. 8	БЛУ ДШ. АН. $\vee$ (РИДЗУ $[1]=0$) $\vee$ ПР. ФИКС. АДР. Д	БМУ	В, Г (при $У=1$; коды ≥ 32)	C1
Блок. 9	БЛУ ДШ. АН. $\vee$ (РИДЗУ $[1]=1$) $\vee$ ПР. ФИКС. АДР. Д	БМУ	В, Г (при $У=0$; коды $B \geq 48$; коды $G \geq 32$)	C1
БЛУ ДШ. ЭЛ. ОП.	{[.один* $\vee$ КНОПКА ВКЛ. ПИТ.* $\vee$ СЕРВОС СИСТ.* $\vee$ ВВОД* $\vee$ ПОВТ. РССП* $\vee$ ЗАПИСЬ* $\vee$ СЧИТ* $\vee$ УСТ. РССП {40/63}] $\vee$ ОСТАТКОВ ПО АДРЕСУ ДЗУ $\vee$ (ТЧП=1)] $\vee$ ТАКТ*} $\vee$ БЛ. ДШ.	БПУ	все остальные поля	C1, C2

6.7. Блок управляющих регистров и блок прямого управления

Блок управляющих регистров — БУР

БУР предназначен для хранения слова состояния программы и кода выполняемой операции. В нем фиксируются 15 причин программных прерываний и устанавливается приоритетность обслуживания 4-х классов прерываний (программных, обращений к супервизору, внешних и вводно — выводных).

Состав БУР (рис. 6.7.1):

РССП — регистр слова состояния программы;

РВО — регистр выполняемой операции;

РОЦ — регистр ошибок процессора;

РФП — регистр фиксации прерываний;

РФПД — регистр фиксации прерываний дополнительный;

ДШ — дешифраторы элементарных операций (на рис. 6.7.1 не показаны).

Регистры блока БУР и его дешифраторы размещаются в раме С процессора.

Назначение и устройство регистров блока БУР

1. Регистр слова состояния программы — РССП.

РССП — это 64-разрядный регистр, предназначенный для хранения слова состояния программы.

Функциональная схема РССП изображена на рис. 6.7.6.

Отдельные разряды регистра РССП имеют следующие назначения:

а) РССП [0/7] — разряды маски системы. В ЭВМ ЕС-1030 используются только разряды [0/3], соответствующие мультиплексному и селекторным каналам и [7] разряд:

РССП [0] — маска запросов на прерывания от МК;

РССП [1] — маска запросов на прерывания от 1СК;

РССП [2] — — — — — 2СК;

РССП [3] — — — — — 3СК;

РССП [7] — разряд маски внешнего прерывания.

Прерывания от каналов (от ввода—вывода) или внешних сигналов выполняются (разрешены), если соответствующий разряд маски каналов или внешних прерываний находится в состоянии «1».

Информация в РССП [0/7] поступает из регистра Р5 [0/7], [8/15], [16/23], [24/31] с помощью соответствующих элементарных операций. Например, РССП [0/7] := Р5 [0/7].

б) РССП [8/11] — разряды ключа программы.

При обращении процессора к оперативной памяти ключ программы сравнивается с ключом памяти, считанным из памяти ключей защиты.

в) РССП[12/15] — разряды, отражающие состояние системы; РССП[12] — единичное и нулевое значения этого разряда означают использование соответственно кодов КОИ-8 или ДКОИ;

РССП[13] — единичное значение этого разряда означает разрешение прерывания от схем контроля машины, а нулевое — запрещение;

РССП[14] — единичное и нулевое значение этого разряда означают соответственно, что процессор находится в состоянии „Ожидание“ или „Счет“;

РССП[15] — единичное и нулевое значение этого разряда означают соответственно, что процессор находится в состоянии „Задача“ или „Супервизор“.

Информация в РССП[12/15] засылается из регистра Р5[8/15].

г) РССП[16/31] — разряды кода прерывания. Коды прерывания поступают:

- из регистра Р5[16/31] при программных прерываниях, прерывании для вызова супервизора и при выполнении команды *LPSW* (установить ССП);

- от третьего регистра внешних линий (РВЛЗ);

- от триггера аварийного счета времени (ТАСВ);

- от триггера прерывания пульта управления (ТППУ);

- при внешних прерываниях;

- от каналов;

- при прерываниях от ввода—вывода.

д) РССП[32/33] — разряды кода длины выполняемой команды. Это двухразрядный счетчик, который устанавливается в ноль перед началом выполнения команды и к содержимому которого добавляется „1“ после выборки из оперативной памяти каждого полуслова команды. Одновременно с этим адрес команды (разряды 40/63) увеличивается на +2 (см. ниже пункт „З“).

е) РССП[34/35] — разряды кода условия. Код условия устанавливается процессором при выполнении большинства команд с помощью элементарной операции РССП[34/35] := конст.

Значение константы определяется по результату операции. Код условия устанавливается также каналами при выполнении ими команд ввода — вывода.

ж) РССП[36/39] — разряды маски программы.

РССП[36] — единичное значение этого разряда разрешает прерывание при переполнении в операции с фиксированной запятой, а нулевое — запрещает;

РССП[37] — единичное значение разрешает прерывание при переполнении в операциях десятичной арифметики;

РССП[38] — единичное значение разрешает прерывание при исчезновении порядка в операциях с плавающей запятой;

РССП[39] — единичное значение разрешает прерывание при потере значимости мантиссы.

з) РССП[40/63] — двадцатичетырехразрядный счетчик, предназначен для хранения и получения адреса очередной команды.

Адрес очередной команды получается сложением содержимого счетчика с числом, равным количеству байтов выполняемой команды. При выполнении прерываний или команд переходов содержимое счетчика изменяется целиком.

Счетчик построен на триггерах с потенциальными входами. Каждый разряд счетчика состоит из двух самостоятельных триггеров, поэтому получается двухрядовый счетчик:

РССП [40/63] — основной ряд,

РССП [40/63] — вспомогательный ряд.

Счетчик может принимать информацию из регистров Р5 [8/31], РИОП [8/31] (или константу „144“ для диагностических процедур).

В счетчике основного ряда имеется возможность прибавления „+2“, поэтому младший разряд счетчика — РССП [63] не участвует в операции сложения. Добавлением в счетчике „+2“ фиксируется тот факт, что процессор обработал очередное полуслово (два байта) команды и этим подготавливается адрес следующей команды.

Каждый раз после изменения содержимого счетчика информация из основного ряда передается во вспомогательный ряд. Для этого единичные выходы триггеров разрядов основного ряда соединены с единичными входами соответствующих разрядов вспомогательного ряда.

Счетчик имеет вход установки нуля. Обнуление счетчика происходит только перед приемом новой информации (т.е. при полном изменении его содержимого).

Работу счетчика можно характеризовать выполнением следующей последовательности управляющих тактов:

Т—1 РССП [40/63] := 0,

Т—2 РССП [40/63] := Р5 [8/31] ∨ РИОП [8/31] ∨ 144,

Т—3 РССП [40/63] := РССП [40/63],

Т—4 РССП [40/63] := „+2“,

Т—5 РССП [40/63] := РССП [40/63].

При прерываниях и в операциях переходов (ветвлений) выполняются все такты (Т1/Т5), а при последовательном выполнении команд программы выполняются только такты Т—4, Т—5.

Загрузка регистра РССП осуществляется либо с пульта управления, либо специальной командой ЗАГРУЗИТЬ ССП (*LPSW*) по специальным микропрограммам (рис. 6.9.21).

Содержимое разрядов масок может изменяться в процессе выполнения рабочей программы специальными командами УСТАНОВИТЬ МАСКУ СИСТЕМЫ (*SSM*) и УСТАНОВИТЬ МАСКУ ПРОГРАММЫ (*SRM*) по микропрограммам, представленным в альбоме схем (рис. 6.9.7).

Регистр РССП выполнен на ТЭЭ четырех типов: ЕС-Т000/0002, ЕС-Т000/0003, ЕС-2030/0013, ЕС-2030/0045.

На рисунке 6.7.2 изображены четыре младших разряда 60, 61, 62, 63 счетчика РССП [60/63], выполненные на интегральных микросхемах типа Т110 (ТЭЗ ЕС-Т000/0003).

Единичные выходы n -го разряда вспомогательного ряда (РССП" [40/61]) подсоединены к логическим схемам нулевого и единичного входов $n-1$ разряда основного ряда (РССП [40/61]).

Например, выход „1“ РССП" [61] подсоединен ко входам РССП [60]. Нулевые выходы вспомогательного ряда РССП" [40/62] подсоединены к единичным входам основного ряда. Выход „0“ РССП" [61] подсоединяется к единичному входу РССП [61]. Подключение 62 и 63 разрядов имеет некоторое отличие (рис. 6.7.2). После каждой пары разрядов счетчика собрана цепь переноса. На рис. 6.7.2 цепи переноса выполнены на микросхемах АЗ-2, АЗ-3 (РССП" [62/63]=1) и АЗ=1, АЗ=2 (РССП" [60/63]=1). Такое подключение регистра РССП" [40/63] к входной логике регистра РССП [40/63] и наличие цепей переноса обеспечивают работу регистра-счетчика РССП [40/63] в режиме суммирования +2 с целью получения адреса очередной команды программы.

Сигналы:

$$\left. \begin{array}{l} \text{„1“ Р5 [28]} \vee \text{„1“ РИОП [28];} \\ \text{„1“ Р5 [29]} \vee \text{„1“ РИОП [29];} \\ \text{„1“ Р5 [30]} \vee \text{„1“ РИОП [30];} \\ \text{„1“ Р5 [31]} \vee \text{„1“ РИОП [31]} \end{array} \right\} *$$

формируются в виде импульсов отрицательной полярности. Эти сигналы поступают только тогда, когда в схему управления разрядами 40/63 регистра РССП (РССП [40/63]) подан высокий уровень

$$\text{РССП [40/63]} := \text{Р5 [8/31]} \vee \text{РИОП [8/31]}.$$

При отсутствии этого уровня вместо сигналов (*) на соответствующие входы ИС подаются высокие уровни („1“).

Проследить подключение других разрядов РССП" [40/63] и РССП [40/63] можно, используя принципиальные схемы ТЭЗ ЕС-Т000/0003, ЕС-Т000/0018 и функциональную схему регистра РССП (рис. 6.7.4—6.7.6).

При этом следует иметь в виду, что для других групп разрядов: 56/59, 52/55 и т. д. на контакт „а“ (см. рис. 6.7.2) вместо положительного потенциала подключается сигнал переноса из младшей группы (РССП" [60/63]=1, РССП" [56/59]=1 и т. д.).

На контакт „б“ других групп подается единица старшего разряда группы („1“ РССП [59], „1“ РССП [55] и т. д.).

2. Регистр выполняемой операции — РВО.

РВО — это восьмиразрядный регистр, предназначенный для хранения кода выполняемой операции.

Функциональная схема РВО изображена на рис. 6.7.3.

Собственно регистр выполнен на одном ТЭЗ ЕС-Т000/0002 (рис. 6.7.7), расположенном по адресу 1A07. Для согласования выходов регистра РВО со входами регистра РАДЗУ используется еще 5 типов ТЭЗ: ЕС-2030/0069, ЕС-2030/0068, ЕС-2030/0066, ЕС-2030/0061, размещенных соответственно по адресам 2A20, 1A08, 1D15, 2A08, 2A05 (рис. 6.7.3).

Информация в регистр РВО передается из регистра Р5[0/7] в момент прихода управляющего сигнала $РВО := Р5[0/7]$ на контакт 26 (33) (рис. 6.7.3). Элементарная операция РАДЗУ := РВО обеспечивает засылку в регистр адреса ДЗУ начального адреса микропрограммы выполняемой инструкции (команды) рабочей программы.

На рис. 6.7.3. по адресу 1A05 ТЭЗ ЕС-2030/0046 расположены дешифраторы анализов, выдающие сигналы $РВО[0/3] \neq 1111$, $РВО = 44$, $РВО[5/6] \neq 0$, которые используются для ветвления микропрограмм.

Назначение триггера, расположенного по адресу 2A16 (рис. 6.7.3), будет пояснено ниже.

3. Регистр ошибок процессора — РОЦ.

РОЦ предназначен для фиксации и хранения причин программных прерываний, возникающих в процессе выполнения команд процессором.

РОЦ состоит из 18 разрядов.

Разряды 1/15 идентифицируют 15 причин программных прерываний, остальные разряды не используются.

Причины программных прерываний и соответствующие им разряды РОЦ даны в таблице 6.7.1. Информация на РОЦ[4] поступает с выхода схемы сравнения ключа программы и ключа памяти. При несравнении ключей в РОЦ[4] фиксируется ошибка защиты памяти. На входы РОЦ[5, 6] специальной элементарной операцией передаются ошибки оперативной памяти из регистра РОП [1, 0] после каждого обращения к ОП.

После обработки прерывания до начала выполнения новой команды рабочей программы регистр РОЦ всегда устанавливается в ноль.

Адреса вызова микропрограмм обработки прерываний формируются с помощью логических схем, подключенных к нулевым выходам триггеров регистра РОЦ.

Типы ТЭЗов регистра РОЦ и их адреса приведены на рис. 6.7.1.

4. Регистр фиксации прерываний (РФП) и регистр фиксации прерываний дополнительный (РФПД).

РФП и РФПД — это 4-разрядные регистры, предназначенные для фиксации четырех типов запросов на прерывания и для определения приоритета их обработки.

Регистр РФП.

Первый разряд регистра фиксации прерываний РФП[1] устанавливается в „1“ при возникновении программных ошибок и этим фиксирует запрос на программное прерывание. С началом обработки программного прерывания РФП[1] сбрасывается в „0“ и до окончания обслуживания прерывания не может быть установлен в „1“, т.е. очередной запрос на прерывание не может быть принят до окончания обслуживания принятого к обработке запроса. Это обеспечивается за счет маскирования запроса на прерывание в соответствующем данному прерыванию слове состояния программы (см. алгоритм обработки программного прерывания на рис. 6.9.3).

РФП[2] — устанавливается в „1“ при выполнении процессором команды *SUC* — ВЫЗОВ СУПЕРВИЗОРА (см. алгоритм микропрограммы команды *SUC* на рис. 6.9.8) и этим фиксирует запрос на прерывание по обращению к супервизору.

Установка в „0“ и прием очередного запроса на прерывание аналогичны программному прерыванию.

РФП[3] — устанавливается в „1“ для фиксации запроса на внешнее прерывание. Это будет происходить всегда при появлении единицы в любом из разрядов регистра внешних линий с номером 3 (РВЛЗ), или в триггере прерываний пульта управления (ТППУ), или в триггере аварийного счета времени ТАСВ.

РВЛЗ, ТППУ и ТАСВ будут рассмотрены при описании блока прямого управления.

РФП[3] сбрасывается в „0“ после начала обработки запроса. РФП[4] своим единичным состоянием фиксирует запросы на прерывания по вводу — выводу. Сигнал, устанавливающий в „1“ РФП[4], поступает из общего канала. Установка в „0“ и прием повторного запроса на прерывание аналогичны первому разряду РФП[1].

Регистр РФПД

С помощью регистра РФПД и схемы приоритетов определяется приоритет обработки прерываний.

Содержимое регистра РФП передается в регистр РФПД с помощью элементарной операции $\text{РФПД} := \text{РФП}$.

Выходы регистра РФПД подключены к схеме приоритетов, с выхода которой снимается адрес запрашиваемой микропрограммы обработки прерываний и передается в регистр РАДЗУ долговременного запоминающего устройства.

При одновременном появлении нескольких запросов на прерывания схема приоритетов определяет следующую очередность их обслуживания:

1. Программные прерывания и прерывания при обращении к супервизору.
2. Внешние прерывания.
3. Прерывания от ввода — вывода.

Программные прерывания и прерывания при обращении к супервизору имеют один приоритет, т.к. они не могут появиться одновременно.

Здесь мы рассмотрели приоритет обработки запросов только четырех типов прерываний. Пятый тип — прерывание от схем контроля машины — имеет самый высший приоритет. Запрос на прерывание от схем контроля (храняемый в регистре машинных отказов) блокирует рассмотрение всех других запросов на прерывания и обслуживается в первую очередь. Следует отметить, что запрос на прерывание от схем контроля прерывает выполнение текущей команды рабочей программы, а остальные запросы воспринимаются только после ее окончания.

Таблица 6.7.1

РОЦ [1/15]	Код прерывания	Причина прерывания
1	00000001	Неверный код операции
2	00000010	Привилегированная операция
3	00000011	Некорректность ком. ВЫПОЛНИТЬ
4	00000100	Защита памяти
5	00000101	Адресация
6	00000110	Спецификация
7	00000111	Данные
8	00001000	Переполнение с фиксированной запятой
9	00001001	Деление с фиксированной запятой
10	00001010	Десятичное переполнение
11	00001011	Десятичное деление
12	00001100	Переполнение порядка
13	00001101	Исчезновение порядка
14	00001110	Потеря значимости
15	00001111	Деление с плавающей запятой

Блок прямого управления

Блок прямого управления — БПРУ предназначен для обмена информацией между двумя ЭВМ Единой системы или ЭВМ и специальным внешним устройством.

Обмен информацией между двумя вычислительными машинами ЭВМ-1 и ЭВМ-2 организуется с помощью команд: ПРЯМАЯ ЗАПИСЬ, ПРЯМОЕ ЧТЕНИЕ.

Команда ПРЯМАЯ ЗАПИСЬ позволяет байт информации из оперативной памяти ЭВМ-1 заслать в ОП ЭВМ-2.

Команда ПРЯМОЕ ЧТЕНИЕ обеспечивает выполнение обратного действия.

Обмен информацией и управляющими сигналами осуществляется по интерфейсу прямого управления, соединяющему обе машины или машину и специальное ВУ. ЭВМ-1 и ЭВМ-2 могут быть различных типов.

Состав БПРУ (рис. 6.7.8):

РВЛ0—РВЛ3 — четыре регистра внешних линий;

РВСЛ — регистр внешних сигнальных линий;
ТППУ — триггер прерывания пульта управления;
ТАСВ — триггер аварийного счета времени.

Интерфейс прямого управления содержит:

— 8 выходных линий информации (ШИН 1 [0/7]);

— 8 входных линий информации (ШИН 2 [0/7]);

— 8 линий синхронизации (ЛС [0/7]);

— 6 линий внешних сигналов (ВС [2/7]);

— 2 выходные линии управления (выходная линия записи

ВЫХ — ЛЗП, выходная линия чтения **ВЫХ** — ЛЧТ);

— 2 входные линии управления (входная линия записи **ВХ** — ЛЗП, входная линия чтения **ВХ** — ЛЧТ).

РВЛ0 [0/7] — восьмиразрядный регистр внешних линий, предназначенный для приема информационного байта из регистра **Р5 [0/7]** блока **БАЛ** (считанного из ОП) и выдачи его в ЭВМ-2 по линиям ШИН 1 [0/7].

РВЛ1 [0,7] — восьмиразрядный регистр внешних линий, предназначенный для приема информационного байта от ЭВМ-2 по линиям ШИН 2 [0/7] и выдачи его в регистр **Р7** блока **БАЛ** для последующей записи в ОП.

РВЛ2 — восьмиразрядный регистр внешних линий, предназначенный для выдачи синхросигналов в ЭВМ-2 по линиям ЛС [0/7]. В качестве синхросигналов передается информация, взятая из разрядов [8/15] (формата *SI*) команд **ПРЯМАЯ ЗАПИСЬ** или **ПРЯМОЕ ЧТЕНИЕ**, которая хранится в регистре **Р8 [0/7]** блока **БАЛ**.

РВЛ3 [2/7] — шестиразрядный регистр внешних линий, предназначенный для приема внешних сигналов от ЭВМ-2 по линиям ВС [2/7]. Прием внешних сигналов в регистр **РВЛ3 [2/7]** осуществляется каждым полутактом **С1**.

РВСЛ [0/1] — двухразрядный регистр внешних сигнальных линий, предназначенный для формирования и выдачи в ЭВМ-2 сигналов управления записью и чтением („Л—ЗП“), („Л—ЧТ“), передаваемых по двум выходным линиям управления (**ВЫХ**—ЛЗП, **ВЫХ**—ЛЧТ).

ТАСВ — триггер аварийного счета времени, предназначенный для фиксации факта перехода содержимого таймера от положительного числа к отрицательному. В этом случае триггер **ТАСВ** устанавливается в „1“, что служит причиной внешнего прерывания в ЭВМ-1.

ТППУ — триггер прерывания пульта управления, предназначенный для образования сигнала внешнего прерывания при нажатии кнопки прерывания на ПУ.

Управление блоком **БПРУ** микропрограммное. На структурной схеме (рис. 6.7.8) слева показаны сигналы элементарных операций, вырабатываемых в процессе выполнения микропрограмм команд **ПРЯМАЯ ЗАПИСЬ** и **ПРЯМОЕ ЧТЕНИЕ**. Алгоритм микропрограммы **ПРЯМОЕ ЧТЕНИЕ** представлен на рис. 6.9.9.

Не приводя подробного разбора алгоритмов, отметим, что в случае отсутствия маскирования внешних прерываний в словах состояния программ появление единицы хотя бы в одном из разрядов РВЛ2 (блока БПРУ ЭВМ-1) вызовет внешнее прерывание в ЭВМ-2. Аналогично появление единицы хотя бы в одном разряде РВЛ3 (блока БПРУ ЭВМ-1) служит причиной внешнего прерывания в ЭВМ-1. Восьмибитный код прерывания в старом ССП устанавливается в соответствии со значениями ТАСВ, ТППУ, РВЛ3[2/7] (ТАСВ и ТППУ соответствуют 0 и 1 разрядам РВЛ3). После записи кода прерывания в регистр РССП все разряды регистра РВЛ3[0/7] сбрасываются в ноль.

Если происходит прерывание от внешнего источника, все запросы, ожидающие рассмотрения, воспринимаются одновременно (например, запрос от ЭВМ-2 и от ТППУ). Каждый запрос рассматривается только один раз. Если до начала обработки приходит несколько запросов от одного источника, происходит только одно прерывание.

Регистры РВЛ0, РВЛ1, РВЛ2 выполнены на ТЭЗ ЕС-Т000/0002, а регистр РВЛ3 — на ТЭЗ ЕС-2030/0068. Триггеры ТАСВ и ТППУ расположены в ТЭЗ ЕС-2030/0048 и ЕС-2030/0022. Принципиальные схемы некоторых из названных ТЭЗ можно посмотреть на рис. 6.7.7, 6.7.9.

6.8. Блок обращения к оперативной памяти

Назначение и структурная схема БООП. БООП служит для удовлетворения запросов на обращение к оперативной памяти со стороны каналов, таймера, процессора и блока диагностики.

Его основные функции состоят в следующем:

- считывание из ОП команд рабочей программы (инструкций);
- выборка из памяти и засылка в память данных по указанному адресу;
- определение очередности удовлетворения запросов на обращение к ОП.

БООП обеспечивает подключение к процессору от 1 до 4 модулей памяти (ОП1—ОП4) емкостью по 128 К байт с циклом обращения 1,5 мкс и временем выборки 0,85 мкс.

Период обращения к ОП, обеспечиваемый БООП, составляет 2,01 мкс при рабочем такте машины 0,67 мкс.

Состав БООП.

БООП состоит из:

- узла запросов и очередности обращения к ОП;
- узла управления обращением к ОП.

Узел запросов и очередности предназначен для приема и запоминания запросов на обращение к ОП и определения их приоритета.

В узел запроса и очередности входят следующие элементы:

РЗПР — 6-разрядный регистр запросов, служащий для приема запросов на обращение к ОП;

РОЧ — 6-разрядный регистр очереди, используемый для организации очереди принятых к обслуживанию запросов;

ОЧ — комбинационная схема, подключенная к выходам регистра РОЧ, определяющая очередность обслуживания запросов в соответствии с их приоритетом.

Узел управления обращением к ОП формирует последовательность управляющих сигналов, необходимых для обращения к ОП при выборке или записи информации.

В узел управления обращением к ОП входят:

РАСП — 29-разрядный регистр адресного слова памяти, принимающий и хранящий адрес обращения к ОП принятого к исполнению запроса;

ЦТИ — цепочка формирования тактовых импульсов, состоящая из 6 триггеров и используемая при выработке сигналов обращения к ОП;

РКЗ — 4-разрядный регистр ключей защиты, служащий для хранения ключа программы;

СРПКЗ — схема сравнения, служащая для определения ошибок защиты памяти, выявляющихся при сравнении ключа программы и ключа памяти;

ОА — узел анализа ошибки адресации, проверяющий на равенство нулю [0/4] разряды регистра РАСП;

ОС — узел анализа ошибок спецификации, связанных с нарушением целочисленных границ при обращении к ОП;

РОП — 3-разрядный регистр ошибок памяти, фиксирующий ошибки защиты памяти (2-й разряд), ошибки адресации (1-й разряд) и ошибки спецификации (0-й разряд).

Кроме перечисленных элементов в состав узла управления обращением к ОП входит ряд схем, обеспечивающих формирование сигналов:

- прием адреса ($\text{Пр А ОП1} \div \text{Пр А ОП4}$);
- выдача числа ($\text{ВЧ ОП1} \div \text{ВЧ ОП4}$);
- позиции байтов ($\text{ПОЗ 1Б} \div \text{ПОЗ 4Б}$);
- „Запись“. Этому сигналу соответствует нулевое значение триггера записи ($\text{ТЗП} = 0$);
- „Чтение“ ($\text{ТЗП} = 1$);
- блокировка записи ($\text{БЛ ЗП ОП1} \div \text{БЛ ЗП ОП4}$), которая выдается в случаях обнаружения ошибок в ОП;
- сигналы адреса ячейки памяти (АДР [0/14]).

Сигналы прием адреса и выдача числа выдаются только в одно из устройств памяти ОП1 ÷ ОП4, определяемое 5 и 6 разрядами регистра РАСП. Сочетание 00 в этих разрядах адресует первый модуль, 01 — второй, 10 — третий, 11 — четвертый. Остальные сигналы передаются параллельно во все модули памяти (ОП1—ОП4). На блок-схеме БООП показаны пунктиром только два модуля оперативной памяти ОП1 и ОП2. Модули ОП1—ОП4 являются самостоятельными, не входящими в состав БООП устройствами.

Буквами КП на блок схеме обозначены кабельные передатчики, обеспечивающие усиление сигналов перед передачей их в другие блоки.

Полная структурная схема блока БООП с перечислением всех входящих и выходящих сигналов изображена на рис. 6.8.2. Временная диаграмма работы процессора с ОП изображена на рис. 6.8.3.

Узел запросов и очередности обращения к ОП

Запрос на обращение к ОП осуществляется установкой в „1“ соответствующего запрашиваемому устройству разряда регистра РЗПР. Запрос процессора фиксируется установкой в „1“ четвертого разряда, запрос таймера (на коррективную машинных часов) запоминается во 2 разряде, а запрос канала — в нулевом разряде РЗПР.

Сигналы элементарных операций, соответствующие названным запросам на схемах и в микропрограммах обозначаются следующим образом:

РЗПР [4] := 1,

РЗПР [2] := 1,

РЗПР [0] := 1.

Разряды 1, 3, 5 регистра РЗПР используются только при мультисистемной работе.

Запрос обслуживается сразу, если он единственный и ОП в момент поступления запроса свободна. Если одновременно поступают несколько запросов, то сначала обслуживается запрос, обладающий более высоким приоритетом. Самым высоким приоритетом при обращении к оперативной памяти обладает канал, затем таймер, а процессору отведен низший приоритет. Приоритет обращения к ОП между мультиплексным и селекторным каналами определяются в общем канале, через который осуществляется связь каналов с процессором.

Одновременно с приемом запроса запускается цепочка тактовых импульсов узла управления обращением к ОП, которая выдает тактирующие импульсы ТИ1*, ТИ2, ТИ2*, ТИ3, ТИ3*, ТИ4 (см. временную диаграмму рис. 6.8.3).

Триггеры ТИ тактовой цепочки, обозначенные звездочкой, устанавливаются в „1“ в полутакте С1, а опрашиваются (т. е. используются для формирования управляющих сигналов) в полутакте С2. Триггеры, устанавливающиеся в „1“ в полутакте С2, а опрашивающиеся в полутакте С1, обозначены без звездочек. Установку и опрос триггеров можно проследить по временной диаграмме (рис. 6.8.3). Запуск цепочки тактовых импульсов осуществляется установкой первого триггера цепочки ТИ1* в единицу одним из поступающих запросов (РЗПР [0] := 1, РЗПР [2] := 1, РЗПР [4] := 1) по тактовому импульсу С1. Переход от такта к такту происходит автоматически. После такта ТИ4 всегда запускается такт ТИ1*. При отсутствии очередных запросов на обращение к ОП работа

тактовой цепочки обрывается на такте ТИ2. Если же в очереди есть запросы, то цепочка „продолжается“ и обслуживает очередной запрос.

Если запросы приходят в моменты, когда тактовая цепочка работает, то они запоминаются и ждут своей очереди.

Цепочка тактовых импульсов за один цикл работы обеспечивает формирование последовательности управляющих сигналов для одного обращения к ОП.

По импульсу ТИ1* цепочки тактовых импульсов содержимое регистра РЗПР (принятые запросы) передается в регистр очереди РОЧ, на выходе которого собрана схема приоритета. На блок-схеме она обозначена ОЧ.

Эта схема определяет очередь устройствам (каналу, таймеру, процессору) для обращения к ОП. То устройство, запрос которого в данный момент обладает наивысшим приоритетом, извещается специальным сигналом ($ОЧ=0, 2, 4$) о том, что будет обслуживаться его запрос.

Сигналом $ОЧ=0$ каналу сообщается о том, что его запрос принят к обслуживанию, таймеру выдается сигнал $ОЧ=2$, а процессору — $ОЧ=4$.

Затем соответствующий принятому запросу разряд регистра РЗПР устанавливается в ноль и, таким образом, РЗПР подготавливается к приему следующего запроса от этого же устройства.

После выдачи сигнала „очереди“ управление передается узлу управления обращением к ОП.

Узел управления обращением к ОП

Этот узел предназначен для управления процессом выборки или записи информации в ОП.

Последовательность выдачи управляющих сигналов при обращении к ОП определяется состоянием управляющих триггеров и триггеров цепочки тактовых импульсов ЦТИ, запускаемой одновременно с поступлением первого запроса в регистр РЗПР.

Управляющие сигналы блока БООП, как и в других блоках процессора, выдаются (тактируются) синхроимпульсами серий С1 и С2 (см. временную диаграмму).

Управляющие триггеры

Управляющие триггеры упоминались при рассмотрении блок-схемы процессора. Рассмотрим здесь особенности работы этих триггеров и их связь с импульсами, сформированными цепочкой ЦТИ.

ТЗП — триггер записи.

Состояние триггера записи определяет режим обращения к ОП.

Режиму „Запись“ соответствует нулевое значение триггера $ТЗП=0$, а режиму „Чтение“ — единичное $ТЗП=1$.

Всегда после пуска тактовой цепочки триггер ТЗП устанавливается в единичное состояние по импульсу ТИ1*, т. е. импульсом ТИ1* включается режим „Чтение“.

Для включения режима „Запись“ необходимо, чтобы одновременно присутствовали следующие три сигнала:

— ТИ2;

— $ОЧ = 0 \vee 2 \vee 4$;

— $ТЗП := 0$.

Сигналы ТИ2 и $ОЧ = 0 \vee 2 \vee 4$ формируются аппаратно в схемах БООП. А сигнал $ТЗП := 0$, только при обращении к ОП от процессора или таймера, микропрограммно.

При совпадении во времени вышеуказанных сигналов, т. е. при $ТИ2 \wedge (ОЧ = 0 \vee 2 \vee 4) \wedge ТЗП := 0$ триггер ТЗП устанавливается в „0“ и этим состоянием задает во всех ОП (ОП1 — ОП4) режим „Запись“.

При организации блоком БООП режима „Чтение“ сигнал элементарной операции $ТЗП := 0$ отсутствует и поэтому триггер ТЗП остается в единичном состоянии, в которое он был установлен импульсом ТИ1*, то есть при обращениях к ОП только для считывания информации триггер ТЗП всегда будет сохранять состояние „1“.

ТДБ — триггер длины байтов

Триггер ТДБ своим состоянием определяет длину обрабатываемых процессором данных.

Если $ТДБ = 1$, то обрабатываются данные длиной в двойное слово. При $ТДБ = 0$ обрабатываются данные длиной в одно слово. Условие установки триггера ТДБ в „1“ следующее:

$$ТИ2 \wedge ОЧ = 4 \wedge ТДБ := 1$$

Триггер ТДБ участвует в работе блока БООП только при обращении к ОП со стороны процессора. Поэтому в условии и показан только один из сигналов очереди ($ОЧ = 4$).

Сигнал $ТДБ := 1$ формируется микропрограммно.

В нулевое состояние триггер ТДБ устанавливается по импульсу ТИ1*.

ТФАЦ — триггер фиксированных адресов процессора.

Триггер ТФАЦ своим единичным состоянием указывает на то, что организуется обращение к ОП по фиксированному адресу.

Устанавливается в „1“ при выполнении условия:

$$C1 \wedge ТФАЦ := 1,$$

т. е. по синхроимпульсу $C1$ при одновременно поступившем сигнале $ТФАЦ := 1$.

Устанавливается в „0“ при

$$ТИ3 \wedge C2 \wedge ОЧ = 4.$$

Если $ТФАЦ=1$, то в разряды РАСП [16/23] передается содержимое разрядов РИДЗУ [48/55], а разряды РАСП [12/15] устанавливаются в ноль.

ТДАН — триггер данных.

В оперативной памяти машины размещаются как команды рабочей программы решаемой задачи (задач), так и данные в виде исходных, промежуточных и конечных результатов вычислений. Единичное состояние триггера данных $ТДАН=1$ означает, что из ОП выбирается очередная команда. Когда $ТДАН=0$, это означает, что информация в ОП представляет собой данные (т.е. операнды), используемые или формируемые при вычислениях.

ТДАН устанавливается в „1“ микропрограммно, а сбрасывается в ноль аппаратно по ТИЗ*.

ТФАК — триггер фиксированных адресов канала.

Используется для задания фиксированных адресов оперативной памяти при обращении к ней каналов.

Устанавливается в „1“ по

$$ТИ2 \wedge ОЧ = 0 \wedge ТФАК : = 1.$$

Сбрасывается в „0“ по ТИ1*.

Триггер ТФАК устанавливается в „1“ и при обращении по фиксированному адресу ОП процессора по условию:

$$ТФАЦ = 1 \wedge ОЧ = 4 \wedge ТИ2.$$

Эта установка в „1“ триггера ТФАК при обращении к ОП процессора связана со схемным формированием фиксированного адреса.

Работа узла обращения к ОП

После запуска ЦТИ, определения приоритета поступившего запроса и выдачи сигнала $ОЧ = 0 \vee 2 \vee 4$ управление передается узлу управления обращением к ОП.

Устройство, запрос которого обслуживается в данный момент, выдает в БООП адрес, который поступает на 29-разрядный регистр РАСП в такте ТИ2 (см. временную диаграмму работы процессора с ОП и блок-схему БООП).

Состояния управляющих триггеров ТФАЦ, ТДАН и ТФАК определяют на входах РАСП источник адреса. При обращении к ОП процессора за выборкой инструкций адрес принимается из регистра РССП [40/63], при выборке или записи операндов — из сумматора СМ [8/31], а при обращении по фиксированным адресам — из регистра РИДЗУ [48/55] или из регистра фиксированных адресов канала (Фикс адр кан). При обращении к ОП из каналов адрес поступает из общего канала (Адр кан).

В такте ТИ2 (или ТИЗ в зависимости от режима „Запись“, „Чтение“) в регистр адреса памяти ключей защиты РАПКЗ (ПКЗ

на блок-схеме БООП изображена слева внизу) заносится адрес из РАСП [5/12] или из общего канала (Адр. ПКЗ из ОК). Одновременно с этим в регистр ключей защиты РКЗ заносится ключ программы из регистра РССП [8/11] (в случае обращения к ОП процессора) или из коммутатора ключей защиты канала (КЗКАН), если обращение к ОП идет от канала. В этом же такте сигналом „считывание ПКЗ“ (СЧ ПКЗ) запускается память ключей защиты.

В блоке БООП предусмотрена возможность записи информации в ОП не только словами, но и одним или несколькими байтами. Для этого введены разряды РАСП [24/25], которые указывают количество записываемых байтов. Информация о количестве записываемых байтов заносится в РАСП [24/25] из регистра РИДЗУ [54/55] при обращении к ОП процессора, или с помощью потенциалов маркера: (МКР-[0/1]) при обращении к ОП из каналов.

При записи информации по байтам дешифратор позиций записываемых байтов (ДШ ПОЗ БАЙТОВ) выдает сигналы каждого записываемого байта (ПОЗ 1Б—ПОЗ 4Б).

Записываемый одиночный байт может замещать в ОП любой байт адресуемого слова.

При записи двух и большего числа байтов записываемые (и замещаемые) байты могут располагаться в слове только рядом, т.е. вплотную друг к другу. Разряды РАСП [22/23], поступающие на вход дешифратора ДШ ПОЗ БАЙТОВ, определяют позицию байта в слове (1, 2, 3, 4), начиная с которой записывается один байт или группа байтов. А разряды РАСП [24/25] определяют количество записываемых байтов.

Если операция выполняется над полями переменной длины, то для того, чтобы записать четыре байта, подается сигнал „Блокировка позиции байта“ (БЛ ПОЗ Б), который обеспечивает выдачу всех четырех сигналов позиций байтов (ПОЗ 1Б—ПОЗ 4Б).

При считывании из ОП дешифратор ДШ ПОЗ БАЙТОВ выдает все 4 сигнала ПОЗ 1Б—ПОЗ 4Б, т.е. считывание всегда осуществляется словами.

В такте ТИЗ дешифратором модуля (ДШ Модуля) вырабатывается сигнал „прием адреса“ (ПР А1—ПР А4) длительностью примерно 200 нс, которым запускается соответствующий модуль ОП (ОП1—ОП4). К этому времени после считывания ключа памяти в схеме сравнения СРК производится сравнение ключа программы, хранящегося в регистре РКЗ, с ключом памяти, считанным в РИПКЗ.

Результат сравнения выявляет ошибку защиты, которая может фиксироваться в регистре РОП [2].

При считывании, в случае отсутствия зафиксированных в регистре РОП [0/2] ошибок, схема выдачи числа (ВЫД ЧИСЛА) вырабатывает один из 4-х сигналов ВЧ ОП1—ВЧ ОП4. Эти сигналы открывают выходы регистров информации РИОП1—РИОП4 для выдачи ее в устройства. Если же при считывании будут обнаружены ошибки (адресации, спецификации или защиты памяти по

чению), то эти ошибки фиксируются в регистре РОП [2], выходы которого подключены ко входу схемы „ВЫД ЧИСЛА“, и блокируют выдачу сигналов ВЧ ОП1 — ВЧ ОП4.

При записи, в случае наличия в регистре РОП [0/2] хотя бы одной из трех ошибок, схема блокировки записи (Бл записи) превращает запись в оперативную память, выдавая сигнал „Блокировка записи“ (БЛ ЗП) длительностью 670 нс. Этот сигнал переводит ОП из режима записи в режим считывания и, таким образом, содержимое адресованной ячейки не изменяется. Содержимое регистра РОП в такте ТИЗ* передается в регистр ошибок процессора РОЦ [4/6], если обслуживается процессор, или в регистр подканала, при обслуживании запроса канала. Выявленная при записи информации ошибка защиты памяти в регистре РОП [2] не фиксируется, а сразу передается в регистр ошибок процессора РОЦ [4] или в регистр подканала.

Для того, чтобы защита памяти осуществлялась как в режиме записи так и в режиме считывания, четвертый разряд ключа памяти должен быть в единичном состоянии, т. е. РИПКЗ [4] = 1.

Если РИПКЗ [4] = 0, то при считывании ошибка защиты не проверяется. Анализ и выявление ошибок защиты, адресации и спецификации блокируется, если обращение к ОП выполняется по фиксированным адресам.

Если ошибка произошла в ОП, то есть в процессор поступили потенциалы „СБОЙ РАОП“ или „СБОЙ РИОП“ из ОП, то блок диагностики при помощи потенциала „Стоп управление ОП“ блокирует установку триггера ТИ1* и тем самым останавливает работу блока БООП. После этого блок диагностики осуществляет запись ситуации ЭВМ (состояние всех регистров) в служебную область ОП.

После выработки всех управляющих сигналов тактом ТИ4 запускается снова такт ТИ1* и производится передача содержимого регистра РЗПР в регистр очереди РОЧ, т. е. проверяется наличие следующего наиболее приоритетного запроса. Если после передачи РЗПР в РОЧ окажется, что запросов нет, то тактовая цепочка ЦТИ останавливается на такте ТИ2 с последующей установкой триггера ТИ1* в „0“ в случае, если в этот момент не поступит новый запрос.

Последовательность выдачи основных управляющих сигналов блока БООП в режимах „Чтение“ инструкции и „Записи“ двойного слова показана на временной диаграмме работы процессора с ОП. Каждый цикл одного обращения к ОП при считывании инструкции или записи двойного слова начинается по сигналу элементарной операции РЗПР [4] := 1. На временной диаграмме над каждым импульсом тактовой цепочки (ТИ1*, ТИ2 и т. д.) указаны действия, которые выполняются за время данного такта. Если при записи действия применен знак :=, то это обозначает сигнал элементарной операции, вырабатываемый микропрограммно. На-

пример, $ТРКП := 1$, $РИОП := РКП$ ($РКП$ — регистр коммутатора процессора).

Аналогичные записи сделаны над управляющими сигналами, вырабатываемыми блоком БООП.

Блок БООП обеспечивает возможность обращения к ОП и ПКЗ с пульта управления в наладочном режиме.

6.9. Микропрограммы процессора

В процессоре ЭВМ ЕС-1030 используются следующие микропрограммы:

1. Микропрограмма **ВЫБОРКА** — обеспечивающая выборку из ОП команд рабочей программы и операндов, участвующих в операции.

2. Микропрограммы обслуживания прерываний, являющиеся частью системы математического обеспечения ЕС-1030, используются системой прерываний.

3. Микропрограммы машинных команд ЕС-1030, используемые процессором при вычислениях.

4. Диагностические микропрограммы, используемые для диагностики состояния машины и поиска неисправностей.

5. Микропрограммы пульта управления, предназначенные для выполнения пультовых операций, таких как **СБРОС СИСТЕМЫ**, **ПОВТОРИТЬ РССП** и др.

Кроме перечисленных в набор микропрограмм входит еще ряд обслуживающих микропрограмм, обеспечивающих условные и безусловные переходы, переключения программных состояний и другие действия.

В техническом описании машины ЕС-1030 в книгах 3—12 помещен полный набор микропрограмм и соответствующих им алгоритмов. Для примера в таблице 6.9.1, приведена часть микропрограммы **ВЫБОРКА**.

Информация в микропрограмме размещена следующим образом. В графе «адрес десятичный» указан десятичный адрес текущей (выполняемой в данный момент) микрокоманды.

В графах РАДЗУ 0/3, 4/7, 8/11 указан шестнадцатиричный адрес текущей микрокоманды.

В графах „С1“ и „С2“ указаны все элементарные операции, выполняемые текущей микрокомандой соответственно в полутактах С1 и С2. Элементарные операции анализов размещены в графе „С1“.

Под элементарной операцией анализа понимается проверка состояний триггеров, счетчиков и др. условий. Например, в первой строке таблицы 6.9.1 запись вида $ТИВ = 1$ (Б29) обозначает элементарную операцию анализа состояния триггера интервалов времени ТИВ. Проверяется два возможных состояния триггера: $ТИВ = 1$ и $ТИВ = 0$. Данная операция анализа выполняется с помощью кода 29 поля „Б“ микрокоманды (рис. 6.6.2) и соответ-

ствующего дешифратора, на входы которого, кроме кода 29 поля „Б“, подключены выходы триггера ТИВ.

Запись вида РФПД:=РФП (В24) обозначает, что это элементарная операция передачи содержимого регистра фиксации прерываний (РФП) в регистр фиксации прерываний дополнительный (РФПД).

В операциях анализа применяются знаки $=$, $\neq$, $>$, $<$. Знак $:=$ соответствует элементарной операции передачи.

В графах „РИДЗУ 52/55, 56/59, 60/63“ и „Возможные переходы“ указывается адрес, который засылается в РАДЗУ для выборки из ДЗУ следующей микрокоманды. Исключение составляют случаи условных переходов и смены страниц ДЗУ.

Графа „РИДЗУ 52/55“ в таблице заполняется только в том случае, если в текущей микрокоманде есть элементарная операция РАДЗУ:=конст. В этом случае константа, записанная в РИДЗУ [52/55], передается в РАДЗУ [0/3] как часть адреса следующей микрокоманды. Этим действием выполняется переход на другую „страницу“ ДЗУ. В противном случае в РАДЗУ [0/3] остается предыдущее (старое) значение страницы ДЗУ.

В графах „РИДЗУ 56/59, 60/63“ указывается содержимое РИДЗУ [56/59] и [60/63] текущей микрокоманды, которое передается в РАДЗУ [4/11] как часть адреса очередной микрокоманды.

В графе „Возможные переходы“ указываются значения РАДЗУ [8/11] следующей микрокоманды, обусловленные анализами условий в текущей микрокоманде. Например, для микрокоманды, записанной в первой строке таблицы 6.9.1 (по десятичному адресу 256) возможны два адреса перехода.

Эти адреса определяются состоянием триггера ТИВ. Если триггер ТИВ=0, т.е. условие ТИВ=1 не выполняется, то очередная микрокоманда выбирается по адресу 1.0.1 (десятичный 257). Если триггер ТИВ=1 (т.е. условие анализа выполняется), то следующая микрокоманда будет выбираться по адресу 1.0.3. Микрокомандой, записанной по адресу 1.0.3, осуществляется смена страницы ДЗУ (переход на страницу 15 по адресу 15.3.0).

Записи БЛОК (А63, В63), АНРФПД (Б23), (см. микрокоманду по адресу 1.0.1) обозначают анализ содержимого регистра фиксации прерываний дополнительного (РФПД) на наличие в нем запросов на прерывания. При наличии запросов организуется переход в соответствии с их приоритетом по одному из следующих адресов: 1.0.9, 1.0.10, 1.0.13, 1.0.14.

Замечания:

1. В параграфе 6.6. было отмечено, что при выполнении условий анализов в полях А, Б, В, Г адрес микрокоманды (указанный в графе „РИДЗУ 52/55, 56/59, 60/63“) увеличивается соответственно на 1, 2, 4, 8, и таким образом, получается адрес перехода.

Так, при выполнении условия анализа триггера ТИВ=1 (т.е. ТИВ находится в «1»), к адресу 1.0.1 добавляется 2 и получается адрес перехода 1.0.3.

В графе „Возможные переходы“ этот адрес перехода указан только цифрой 3, а цифры 1.0 подразумеваются.

2. При организации анализов состояний регистров с помощью так называемой „Блокировки“ (Блок) правило изменения адреса микрокоманды на 1, 2, 4, 8 (в соответствии с анализируемыми полями А, Б, В, Г) может нарушаться. Это и имеет место в микрокоманде по адресу 1.0.1. Возможные адреса перехода из этой микрокоманды 1.0.9, 1.0.10, 1.0.13, 1.0.14 не соответствуют вышеуказанному правилу.

Использование микропрограмм, записанных в таблицы вида 6.9.1, затруднительно. В таких таблицах очень трудно увидеть всю цепочку выполняемых микрокоманд, т. к. часто в микропрограмме имеют место переходы через несколько страниц.

На практике при работе с микропрограммами удобнее пользоваться алгоритмами микропрограмм, в которых последовательность выполняемых действий (микрокоманд) представляется в виде некоторой совокупности вычислительных и логических блоков. Вычислительные блоки изображаются прямоугольниками, а логические — ромбами. Блоки соединяются друг с другом стрелками, которые отражают последовательность выполнения микрокоманд. Над каждым блоком записывается адрес соответствующей ему микрокоманды.

Внутри каждого блока записываются элементарные операции или другие действия, выполняемые в полутаках работы машины С1 и С2. Содержание этих записей соответствует записям в графах „С1“ и „С2“ таблицы 6.9.1. Стрелки, выходящие из логических блоков, указывают на возможные адреса переходов в случаях выполнения или невыполнения записанных внутри ромба условий анализа.

Микропрограмма ВЫБОРКА инструкций и операндов

Основное назначение микропрограммы ВЫБОРКА — это выборка команд рабочей программы из оперативной памяти и выборка операндов из местной памяти и оперативной памяти.

Алгоритм микропрограммы ВЫБОРКА из-за его громоздкости мы не будем раскрывать во всех деталях. Рассмотрим лишь основные идеи с привязкой их к адресам алгоритма микропрограммы ВЫБОРКА, представленного в альбоме на рис. 6.9.10.

Укрупненный алгоритм микропрограммы ВЫБОРКА изображен на рис. 6.3.1.

При рассмотрении алгоритма следует еще раз вернуться к главе 1 и параграфам 6.1—6.8 с тем, чтобы четко представить себе все форматы команд, регистровые операнды, операнды в оперативной памяти и вспомнить назначение и структуру микрокоманды. Особое внимание надо уделить способам формирования адресов очередных микрокоманд. Необходимо также знать назначение и принципы работы всех регистров и счетчиков, используемых в блоках процессора.

В начале выполнения рабочей программы или после выполнения очередной команды управление передается микропрограмме **ВЫБОРКА** (начальный адрес 1.0.15 рис. 6.9.10), которая на первом этапе своей работы анализирует программные состояния процессора и проверяет наличие запросов на прерывания.

Считывание информации из ОП осуществляется словами. Команды различных форматов занимают в ОП полуслово (формат *RR*), слово (форматы *RX*, *RS*, *SI*) или полтора слова (формат *SS*). Причем команды всех форматов могут располагаться в ОП начиная с границы слова или с границы полуслова. Поэтому чтение одной команды из ОП осуществляется за одно или два обращения.

Если при чтении из ОП текущей команды была считана частично или полностью последующая, то выполнение микропрограммы **ВЫБОРКА** начинается не с адреса 1.0.15, а с адресов: 1.1.13, 1.1.9 или др.

Выборка следующей команды рабочей программы из ОП будет выполняться только в том случае, если процессор одновременно находится в состоянии „Работа“ и „Счет“. Если в процессоре включено состояние „Стоп“ (триггер рабочего состояния $TPC=1$), то организуется зацикливание микрокоманды по адресу 1.0.15 до тех пор, пока триггер рабочего состояния не будет сброшен в „0“ (процессор переведен в состояние „Работа“).

Если процессор находится в состоянии „Работа“ и „Ожидание“, то микропрограмма анализирует содержимое регистра фиксации прерываний дополнительного РФПД и, если оно не равно нулю, организует выход на обработку прерываний по запросам (на рис. 6.9.10 адрес 1.0.1 — Ан РФД (Б)).

Выборка очередной команды из ОП производится (только в состоянии „Счет“ и „Работа“) по адресу, который находится в текущем слове состояния программы, хранимом в регистре РССП [40/63].

Указание на то, что из ОП следует прочесть инструкцию, дает единичное состояние триггера данных ТДАН, который устанавливается на это состояние элементарной операцией $ТДАН := 1$ одновременно с выставлением в микрокоманде запроса на обращение к ОП $РЗПР[4] := 1$ (там же адрес 1.0.2).

Во время выполнения данной инструкции адрес в РССП [40/63] увеличивается на 2, 4, 6 в зависимости от ее длины, т.е. от числа считанных полуслов (адрес 1.0.7).

Нарушение обычной последовательности выбираемых инструкций может происходить при выполнении команд условных и безусловных переходов, прерываниях или при вмешательстве оператора.

Как было отмечено, в ЭВМ ЕС-1030 применяются пять форматов инструкций (команд): *RR* — длина равна полуслову, *RX*, *RS*, *SI* — длина равна одному слову, *SS* — длина равна полтора слова.

Во всех инструкциях первые два разряда определяют тип и длину формата.

Первые восемь разрядов инструкции представляют собой код операции и посылаются в регистр выполняемой операции (РВО) из регистра Р5.

Адрес 1.1.2, $P5 := \text{РИОП}$, $PVO := P5[0/7]$... продолжение выборки инструкции с границы слова.

Адрес 1.1.3, $P5[0/15] := \text{РИОП}[16/31]$, ... $PVO := P5[0/7]$ продолжение выборки инструкции с границы полуслова.

Выполнение инструкции продолжается только в том случае, если выборка инструкции из ОП осуществлялась без ошибок защиты, адресации или спецификации (при чтении инструкции ошибка спецификации фиксируется, если младший разряд адреса $\text{РАСП}[23] = 1$, т. к. при чтении из ОП слов $\text{РАСП}[23]$ всегда должен быть равен нулю). При фиксации в РОЦ [4/6] одной из ошибок инструкция не выполняется и происходит прерывание программы.

В зависимости от формата выбранной инструкции микропрограмма ВЫБОРКА разветвляется на 4 направления. Ответвление осуществляется засылкой в регистр адреса ДЗУ первых двух разрядов кода операции. Это выполняется элементарными операциями:

— $\text{РАДЗУ}[9/10] := \text{РИОП}[0/1]$ (в алгоритме этой записи по адресу 1.1.2 соответствует $\text{РАДЗУ} := \text{РИОП}[0/1]$),

— $\text{РАДЗУ}[9/10] := \text{РИОП}[16/17]$ (в алгоритме по адресу 1.1.3, $\text{РАДЗУ} := \text{РИОП}[16/17]$).

Передача в $\text{РАДЗУ}[9/10]$ осуществляется с так называемой „Блокировкой“ [адреса 1.1.2 и 1.1.3 — Блок (Б, В)], которая обеспечивает формирование адреса ветвления на форматы RR (1.2.8), RX (1.2.10), RS , SI (1.2.12), SS (1.2.14). По адресам 1.1.2 и 1.1.3 осуществляется также анализ содержимого регистра ошибок процессора $\text{РОЦ}[4/6] = 0$ и, если ошибки есть, выход на прерывания по адресам: 1.2.9; 1.2.11; 1.2.13; 1.2.15.

Формат RR . Для этого формата оба операнда находятся в местной памяти. Тип используемых регистров (РОН или РПЗ) определяется вторым разрядом кода операции, хранящегося в первом байте регистра Р5. Если $P5[2] = 1$, то обращение организуется к регистрам с плавающей запятой. Это обеспечивается передачей $\text{РАМП}[1] := P5[2]$ (адрес 1.1.2).

С помощью анализа состояний второго и третьего разрядов регистра выполняемой операции $PVO[2]$ и $PVO[3]$ микропрограмма ВЫБОРКА внутри формата RR разветвляется на четыре направления (см. адреса микрокоманд 1.2.8 и 1.3.0 микропрограммы ВЫБОРКА).

Для операций перечисленных в первой строке таблицы „Система команд ЕС-1030“ (рис. 1.2.8) — SRM , $BALR$, $BCTR$, BCR , SSK , ISK , SUC регистр $PVO[2/3] = 00$.

Для арифметических операций с фиксированной запятой ФЗ (на рис. 1.2.8 вторая строка) и операций загрузки (*LPR*, *LNR* и т. д.) РВО[2/3]=01. Арифметические операции с плавающей запятой длинные ПЗДл. (третья строка таблицы), короткие ПЗКор. (четвертая строка таблицы) и находящиеся в этих же группах операции загрузки имеют соответственно значения РВО[2/3]=10 и РВО[2/3]=11. Операции с плавающей запятой длинные обеспечивают обработку операндов длиной в два слова, а короткие — только длиной в одно слово.

При РВО[2/3]=00 или 01 считанные из общих регистров (РОН) местной памяти первый и второй операнды засылаются соответственно в регистры РЗ и Р1. После этого микропрограмма ВЫБОРКА передает управление микропрограмме выбранной команды, адрес которой засылается в регистр адреса ДЗУ элементарной операцией РАДЗУ[4/11]:=РВО[0/7].

Для операций с плавающей запятой адреса операндов должны иметь значение только 0, 2, 4, 6, иначе фиксируется ошибка спецификации и операция не выполняется.

При выборке команд, использующих операнды длиной в одно слово (эти команды указаны во второй и четвертой строках таблицы „Система команд ЕС-1030“ для формата *RR*. РВО[2/3]=01 ∨ 11), микропрограмма ВЫБОРКА проверяет адрес первого операнда в МП, который обязательно должен иметь нулевой и третий разряды равными нулю. Эта проверка выполняется элементарной операцией анализа Р8[0,3]=00 (адрес 1.3.14). В Р8[0/3] предварительно заносится адрес первого операнда (*R1*). Адрес второго операнда (*R2*) проверяется в микропрограмме самой команды.

Если операция выполняется над словами двойной длины РВО[2/3]=10, то в микропрограмме ВЫБОРКА проверяются адреса старших слов операндов двойной длины. Проверка адресов состоит в анализах Р8[0,3]=00, Р8[4,7]=00 (микрокоманды 1.2.8 и 1.3.0).

В Р8[0/7] предварительно засылаются адреса старших слов 1 и 2 операндов. Младшие слова операндов находятся в соседних регистрах МП, адреса которых получаются добавлением единицы к адресам регистров, указанным в команде. Эта модификация адреса реализуется элементарными операциями: РСА:=+1, РАМП[2/5]:=РСА (см. адрес 1.3.14). РСА — это вспомогательный регистр-счетчик.

После выборки слов двойной длины первый операнд размещается в регистрах РЗ и Р4, второй — в регистрах Р1 и Р2. Содержимое регистров процессора после выборки формата *RR* показано на рис. 6.9.11.

Формат *RX*. Так как команда формата *RX* имеет длину в 4 байта (одно слово), то она может быть выбрана из ОП одним или двумя обращениями.

Одно обращение к ОП будет в том случае, если младшие разряды адреса команды имеют значение 00 (РССП[62/63]=00), т. е. команда занимает в ОП целиком одну ячейку памяти (целое слово в 32 разряда).

Два обращения к ОП потребуется в случае, если РССП[62/63]=10, т. е. команда занимает младшие 16 разрядов одной ячейки (слова) ОП и старшие 16 разрядов в следующей ячейке (следующем слове) ОП.

Проверка места расположения команды формата *RX* в оперативной памяти (на границе слова или границе полуслова) осуществляется микрокомандой по адресу 1.0.6 анализом РССП[62]=1. Если РССП[62]≠1, то команда расположена в ОП с границы слова, в противном случае (РССП[62]=1) — с границы полуслова. Данная проверка выполняется для всех форматов.

После выборки команды анализируется возникновение ошибок защиты, адресации и спецификации. При наличии ошибок РОЦ [4/6]≠000 команда не выполняется, а организуется выход на прерывание (адреса 1.1.2, 1.1.3).

В команде формата *RX* первый операнд находится в МП, а второй — в ОП. Адрес второго операнда получается двойной модификацией (этот адрес представляет собой сумму содержимого базового регистра (Р4), регистра индекса (Р2) и смещения Р1 [20/31]). Если в инструкции поле базового регистра (разряды [16/19] инструкции) или индекса (разряды [12/15] инструкции) равны нулю, то при образовании адреса используется не содержимое регистра РОН „0“, а нулевое значение соответствующей компоненты адреса.

Проверка на равенство нулю адресов регистров базы и индекса осуществляется анализами РЗ[16/19]=0, Р8[4/7]≠0 (адрес 1.2.10, рис. 6.9.12). После формирования адреса второго операнда микропрограмма ВЫБОРКА внутри формата *RX* разветвляется на 4 направления для выполнения 4 групп операций. Эти группы операций определяются значением РВО[2/3]=00, 01, 10, 11 (см. формат *RX* в таблице „Система команд ЕС-1030“ на рис. 1.2.8).

В операциях с фиксированной запятой, выполняемых над полусловами (РВО[2/3]=00, см. пятую строку таблицы „Система команд ЕС-1030“ команда *STH* и др.), ошибка спецификации фиксируется только тогда, когда младший разряд адреса второго операнда не равен нулю, т. е. операнд расположен в ОП не на границе полуслова.

Адрес первого операнда посылается в Р8[0/3] (адрес 1.1.2 на рис. 6.9.10 (Р5=РИОП... Р8:=Р5[8/15]), адрес второго операнда (после его формирования микрокомандами 1.4.6, 1.4.8, 1.4.9 $A2 = (B2) + (X2) + D2$)) передается в регистр Р4 микрокомандой 1.5.8. К моменту выполнения микрокоманд 1.4.6, 1.4.8, 1.4.9 в регистре РИМП находилось содержимое регистра базы (В2), в регистре Р2 — содержимое регистра индекса (Х2), а в регистре Р1 — значение смещения D2.

Первый операнд после его считывания размещается в P3 и P2, а второй в P1 (см. адреса 1.5.8 и 1.5.9).

Содержимое регистров процессора после выборки инструкций и операндов при значениях РВО [0/3]=0100, 0101, 0110, 0111 показано в таблице рис. 6.9.11.

Форматы *RS*, *SI*. Как и в случае команд формата *RX*, выборка команд форматов *RS* и *SI* может потребовать одно или два обращения к ОП. При выборке команд форматов *RS* или *SI* адрес операнда из ОП получается сложением содержимого регистра базы (*B*) со смещением (*D*). Это действие реализуется микрокомандами: 1.2.12, 1.7.2, 1.7.3, 1.7.8. Значение смещения (*D*) хранилось в P1. Адрес базы хранился в регистре P3[16/19]. Содержимое регистра базы (*B*) последовательностью элементарных операций:

$РСА := P3[16/19]$,

$РАМП := ПСА$,

СЧ (считывание из МП),

$P5 := РИМП$,

$P2 := P5$ засылается в регистр P2. Затем микрокомандой по адресу 1.7.8 вычисляется адрес операнда в оперативной памяти: (*B*) + *D*

$PВ := P2 (B) \rightarrow \text{регистр } PВ$

$РА := P1 (\text{смещение } D \rightarrow \text{в } РА)$

На выходе сумматора получим сумму $РА + PВ$, которая передается в регистр P5 элементарной операцией $P5 := СМ$ (адрес 1.7.12).

Полученный адрес передается в P4 ($P4 := P5$). Как и в формате *RX*, содержимое регистра базы участвует в сложении только в том случае, когда номер регистра не равен нулю. (Анализ P3[16/19]=0, адрес 1.2.12). Первый операнд, определяемый разрядами [8/11] команды (см. формат *RS*), считывается из МП и посылается в P1 ($P5 := РИМП$, $P1 := P5$, см. микрокоманду по адресу 1.7.10).

При считывании операнда из ОП ошибка спецификации не фиксируется, поэтому микрокомандой по адресу 1.1.13 РОП[0] := 0. Проверка адреса на спецификацию (если это нужно) выполняется в микропрограмме команды.

Содержимое регистров процессора после выборки команд форматов *RS* и *SI* показано в таблице рис. 6.9.11.

Формат *SS*. Команда формата *SS* выбирается всегда двумя обращениями к ОП, т. к. команда в ОП занимает полторы ячейки памяти (три полуслова). Ошибки защиты, спецификации и адресации проверяются после каждого обращения к ОП. При обнаружении ошибки команда не выполняется. В зависимости от РССП [62] при обращении к ОП считываются первые 2 или 4 байта инструкции. Адрес инструкции в РССП [40/63] увеличивается на 2 или на 4 и по новому значению РССП [40/63] считывается осталь-

Таблица 6.9.1

Адрес десятичный	РАДЗУ			C1	C2	РИДЗУ			Возможные переходы	Примечания
	0/3	4/7	8/11			52/55	56/59	60/63		
256	1	0	0	ТИВ = 1 (Б29)	РФПД : = РФП (Б24)	0	1	3		
257	1	0	1	БЛОК (А63, Б63); АН РФПД (Б23) ТРС : = 1, если (ПЕРЕКЛ ≠ АВТ) (Н35)		0	15	9, 10, 13, 14		
258	1	0	2	РЗПР [4] : = 1 (Б15) ТДАН : = 1 (Б31)	РОЦ [1/3, 7/21] : = 0 (Л8) РОЦ [4/6] : = 0 (Б23)	0	5			
259	1	0	3	РАДЗУ : = КОНСТ 15 (Ж1, С1, Л15)		15	3	0		
260	1	0	4	Р5 : = 0 (Л11) РВ : = 0 (А14)	РОЦ [4/6] : = РОП (И15) Р1 [0/7] : = Р5 [0/7] (Н4) Р1 [8/19] : = Р5 [8/19] (М4)	0	6			
261	1	0	5	РССП [32/33] : = 0 (Е28) ОЧ ≠ 4 (А62) РОЦ [16, 19, 21] ≠ 0 (Б22)	РОЦ [6] : = 1 (при усл.) (Е28)	0	4	5, 7		
262	1	0	6	РМО [5/8] : = КР5 (П1) ТРС : = 0 (У1, Е23) РССП [62=1] (А61)	РЗ [0] : = Р5 [0] (И10) РЗ [1/31] : = Р5 [1/31] (М5)	1	2	3		
263	1	0	7	РССП [40/63] : = + 2 (М14) РССП [32/33] : = + 1		1	14			
264	1	0	8							

Адрес десятичный	РАДЗУ			C1	C2	РИДЗУ			Возможные переходы	Примечания
	0/3	4/7	8/11			52/55	56/59	60/63		
265	1	0	9	РА : = 0 (ЛЗ) РВ : = 0 (А14) Р5 : = СМ (Д2)			10	7	15	при РФПД=1
266	1	0	10	Р5 : = РССП' [0/13] (Д8)			14	0		при РФПД=2
267	1	0	11	РССП [14] = 0 (Б16)			0	0	2	
268	1	0	12	РАДЗУ : = конст "14" (Ж1, С1, Л14) РМО [5/8] : = КР5 (П1)	РССП [16/31] : = РВЛЗ, ТППУ. (V1, Е31)	14	7	0		
269	1	0	13	Р5 : = 0 (Д11)	РССП [16/31] : = Р5 [16/31] (М7)		0	12		при РФПД=3
270	1	0	14		РОБ [0] : = 1 (В21)		15	2		при РФПД=4
271	1	0	15	ТИС : = ТРС (Б14) ТРС=1 (В48)			0	11	15	
272	1	1	0		РФПД : = РФП (В24)		0	1		
273	1	1	1		РФПД : = РФП (В24)		0	1		

Адрес десятичный	РАДЗУ			C1	C2	РИДЗУ		Возможные переходы	Примечания
	0/3	4/7	8/11			52/55	56/59		
274	1	1	2	P5 := РИОП (Д1); РСА := 0 (Е27) РССП [40/63] := +2 (М14) РССП [32/33] := +1 РОЦ [4/6] ≠ 0 (А59) БЛОК (В63) РАДЗУ [9/10] := РИОП [0/1] (В19)	РВО := P5 [0/7] (Е27), РАМП [1] := { СЧ(Ж21) = P5 (2) } РАМП [2/5] := { (Л11) = P5 [12/15] } Р8 := P5 [8/15] Р3 [16/19] := P5 [16/19]; Р1 [20/31] := P5 [20/31] (И12)	2	8	9 ÷ 15	
275	1	1	3	РСА ≠ 0 (Е27) P5 [0/15] := РИОП [16/31] (Д17) РССП [40/63] := +2 (М14) РССП [32/33] := +1 РОЦ [4/6] ≠ 0 (А59) БЛОК (В63) РАДЗУ [9/10] := РИОП [16/17] (В20)	Р8 := P5 [8/13] РАМП [2/5] := { (Л11) = P5 [12/15] } РВО := P5 [0/7] (Е27) РАМП [1] := { СЧ(Ж21) = P5 [2] }	3	0	1 ÷ 7	
276	1	1	4	ТИС := ТРС (В14) ТРС=1 (В48)		0	11	15	
277	1	1	5	ТИС := ТРС (В14) ТРС=1 (В48)		0	11	15	
278	1	1	6	ТИС := ТРС (В14) ТРС=1 (В48)		0	11	15	
279	1	1	7	ТИС := ТРС (В14) ТРС=1 (В48)		0	11	15	

Адрес деся- тичный	РАДЗУ			C1	C2	РИДЗУ		Возмож- ные пе- реходы	Примечания
	0/3	4/7	8/11			52/55	56/59/60/63		
280	1	1	8	РФП [1] : = 1 (РОЦ ≠ 0) (M15) ТИВ = 1 (B29)	ТРКП : = 0 (Ж21)	1	9	11	
281	1	1	9	РССП" [40/63] : = =РССП[40/63] РМО [11] : = КРССП [40/63] (B46) РЗПР [4] : = (B15) ТДАН : = 1 (E31)		1	13	.	
282	1	1	10						
283	1	1	11	РАДЗУ : = конст "15" (Ж1, С1, Д15)		15	3	0	
284	1	1	12	РВ : = 0 (A14) АН РФД (B23) Р5 : = 0 (Д11)	Р1 [0/7] : = Р5 [0/7] (H4) Р1 [8/19] : = Р5 [8/19] (M4)	2	0	1,2 5/7	
285	1	1	13	ОЧ ≠ 4 (A62) РОЦ [16, 19, 21] ≠ 0 (B22) ТРС : = 1 (ПЕРЕКЛ ≠ АВТ) (H35)	РФПД : = РФП (B24)	1	12	13,15	
286	1	1	14	РССП " [40/63] : = РССП [40/63] РМО [11] : = КРССП [40/63] РВ : = 0 (A14) РА : = 0 (Л3)	(B46)	12	12		
287	1	1	15	РССП [40/63] : = +2 (M14) РССП [32/33] : = +1		1	14		

ная часть инструкции. Код операции посылается в РВО, разряды инструкции [8/15], определяющие длину (длины) операндов — в Р8 и РС1.

В формате SS оба операнда находятся в ОП. Адреса операндов получают сложением содержимого общих регистров, адреса которых указаны полями 16/19, 32/35 инструкции, со смещениями, указанными в полях 20/31, 36/37 соответственно. Адреса операндов засылаются в регистр Р4 (А1) и в регистры Р2, Р3 (А2). Младшие разряды этих адресов могут иметь любое значение, т. е. операнды могут начинаться в любом байте памяти. Формат применяется в операциях с полями переменной длины и в десятичной арифметике.

Содержимое регистров процессора после выборки формата SS показано в таблице рис. 6.9.11.

Обслуживание прерываний

Система прерываний позволяет процессору изменять свое состояние при возникновении определенных условий вне системы, в устройствах ввода — вывода или в самом процессоре. Прерывания делятся на пять классов: прерывания от схем контроля машины, программные прерывания, прерывания при вызове супервизора, внешние прерывания и прерывания от ввода — вывода.

Действия при прерываниях.

Прерывание состоит в запоминании текущего ССП в качестве старого ССП и выборке нового ССП.

Работа возобновляется в состоянии, указанном новым ССП. Новое ССП не проверяется на наличие ошибок программирования, когда оно становится текущим ССП. Эта проверка производится во время выполнения следующей инструкции (команды). Старое ССП содержит адрес инструкции, которая выполнялась бы следующей, если бы прерывание не произошло, и код длины последней выполненной инструкции..

Прерывания происходят только в тех случаях, если процессор не замаскирован для данного источника прерывания. Прерывания от ввода — вывода и внешние прерывания могут быть замаскированы маской системы, четыре из 15 программных прерываний могут быть замаскированы маской программы, а прерывания от схем контроля машины — маской контроля машины.

Прерывание всегда происходит после того, как закончится выполнение текущей инструкции, и перед началом выполнения следующей инструкции. Однако наличие прерывания может повлиять на выполнение текущей инструкции.

Чтобы сделать возможными соответствующие программные действия, следующие за прерыванием, идентифицируется причина прерывания и обеспечивается определение расположения в памяти последней выполненной до прерывания инструкции.

Если процессор получил команду на переход в состояние «Стоп», то перед остановам заканчивается выполнение текущей инструкции и обрабатываются незамаскированные прерывания, которые или уже ожидали обработки, или произошли до окончания последней инструкции.

Более подробные данные идентификации причины прерываний приведены в таблице 6.9.2.

Микропрограммы обслуживания прерываний.

После выполнения очередной инструкции проверяются запросы на прерывания. Во время выполнения инструкции одновременно может возникнуть несколько запросов на прерывание. Одновременные запросы на прерывания обслуживаются в заранее установленном порядке.

Самым высоким приоритетом обладает прерывание от схем контроля машины. Когда возникает запрос на прерывание, выполнение текущей инструкции прекращается, происходит автоматическая запись ситуации, во время которой состояние регистров процессора записывается в диагностическую область памяти, после этого выполняется прерывание от схем контроля машины. Затем обслуживаются программные прерывания, прерывания при обращении к супервизору, внешние прерывания, прерывания от ввода — вывода.

Прерывания от ввода — вывода могут быть замаскированы разрядами 0/3 РССП, внешние прерывания — разрядом 7 РССП, 4 из программных прерываний — разрядами 36/39 РССП, прерывания от схем контроля — разрядом 13 РССП.

Прерывание программы происходит, если процессор не замаскирован для данного источника прерывания.

При выполнении прерывания запоминается старое ССП, т.е. фиксируется причина прерывания (РССП[16/31]) и информация, позволяющая установить расположение в памяти последней выполнившейся инструкции. Программа продолжается по новому ССП, считанному из определенной фиксированной ячейки ОП, предназначенной именно для этой цели.

Если существуют несколько запросов, в первую очередь обслуживается прерывание с наивысшим приоритетом — старое ССП засылается в ОП, а новое считывается из соответствующей ячейки, затем без выполнения какой-нибудь инструкции это новое ССП запоминается как старое для следующего прерывания и считывается новое. Это продолжается до тех пор, пока все запросы не будут обслужены. Запрос на прерывание, поступающий после того, как несколько прерываний уже обслужено, удовлетворяется в соответствии с очередностью, имеющейся в момент поступления этого запроса.

Запросы прерывания фиксируются в регистре фиксации прерываний — РФП. Каждый разряд этого регистра соответствует од-

ному классу прерывания. После выполнения каждой инструкции при наличии запросов определяется прерывание, обладающее в этот момент более высоким приоритетом ($РФПД := РФП, АИ РФПД$).

В зависимости от анализа РФПД микропрограмма отвечает для обслуживания соответствующих прерываний.

Программные прерывания.

В зависимости от причины прерывания текущая инструкция завершается, или подавляется (не выполняется). Если есть признак программного прерывания $РФП[1]=1$, то поочередно анализируются все разряды РОЦ, начиная с 1-го разряда для идентификации причины прерывания (каждый разряд РОЦ соответствует одной из программных ошибок, и соответствующий код прерывания устанавливается в $РССП[16/31]$. Если прерывание программы вызывается несколькими причинами, то в $РССП$ помещается двоичный код наименьшего разряда (со значением 1) РОЦ. Код прерывания передается из $РИДЗУ[52/55]$ через сумматор ($РВ := КОНСТ, РА := 0$) и $Р5$. После этого старое $ССП$ записывается в фиксированной ячейке 0040 и из фиксированной ячейки 104 считывается новое $ССП$.

Прерывание от супервизора.

Запрос на это прерывание возникает при выполнении инструкции **ВЫЗВАТЬ СУПЕРВИЗОР**. Код прерывания устанавливается при выполнении инструкции, после чего микропрограммой обработки прерываний старое $ССП$ записывается в фиксированную ячейку 32 и из фиксированной ячейки 96 считывается новое $ССП$.

Прерывание от ввода — вывода.

Источниками прерываний этого типа являются устройства ввода — вывода. Запросы на прерывание могут поступать в любой момент времени, но анализ их наличия производится после выполнения очередной инструкции процессора. Прерывание выполняется, запускается микропрограмма прерывания от ввода — вывода, если процессор не замаскирован для данного канала и если нет других запросов на прерывание (на программное прерывание, на прерывание от супервизора и на прерывание от внешнего источника). В этом случае каналам выдается сигнал $РФПД = 4 \wedge РОБ[0] = 1$ и микропрограмма прерываний ввода — вывода закидывается на микроинструкции. Программа продолжается только после установки в ноль $РОБ[0]$ (регистра общей блокировки), см. алгоритм на рис. 6.9.6, а это означает, что код прерывания, т.е. номер канала и устройства, каналами уже записан в разряды $РССП[21/31]$.

После этого анализируется отмена запроса на прерывание (состояние триггера РОБ [1], находящегося в канале). Если имеется случай отмены запроса на прерывание РОБ[1]=1, то запоминание текущего РССП, как старого, и считывание нового ССП не происходит, запрос на прерывание от ввода — вывода снимается, РФП[4]:=0 и микропрограмма разветвляется на выборку следующей инструкции, если нет других запросов на прерывание. В случае отсутствия отмены РОБ[1]=0 содержимое РССП записывается в фиксированную ячейку 56, а из фиксированной ячейки 120 выбирается новое ССП.

Внешние прерывания.

Запросы на эти прерывания поступают от таймера, от кнопки прерывания на пульте оператора и от внешних объектов. Прерывание может произойти только тогда, когда РССП[7]=1. Источник прерывания определяется разрядами [24/31] кода прерывания.

Запросы на эти прерывания поступают от таймера, от кнопки времени. Запросы от внешних источников хранятся в шестиразрядном регистре РВЛЗ до тех пор, пока не будут восприняты процессором. При прерывании содержимое этого регистра передается в разряды [26/31] РССП как код прерывания. На РССП[24] передается значение триггера аварийного счета времени ТАСВ, а на РССП[25] — триггера прерывания пульта управления ТППУ. После этого разряды уже воспринятых запросов устанавливаются в „0“, таким образом, если от одного источника пришло несколько запросов, происходит только одно прерывание.

Старое ССП записывается в фиксированную ячейку 24, новое берется из фиксированной ячейки 88.

После обслуживания всех запросов, а также в случае, если обнаруживается, что запросов на прерывание не было, проверяется триггер рабочего состояния — ТРС (который установлен в „1“ после выполнения инструкции, если переключатель режима на ПУ находится в состоянии „Шаг“) и если ТРС=1, то включается индикатор состояния „Руч“ на пульте, и процессор останавливается (происходит заикливание на микроинструкции 1.0.15).

Если же ТРС=0, процессор не прекращает работу. В этом случае процессор, если он не находится в состоянии ожидания, запрашивает ОП для считывания инструкции по адресу, находящемуся в последнем выбранном ССП, или по адресу в текущем ССП (если никакого запроса о прерывании не было).

Если процессор находится в состоянии ожидания, то следующая инструкция не будет выполняться, пока не изменится состояние процессора соответствующим прерыванием от ввода — вывода или внешним прерыванием).

Источник прерывания	Код прерывания, разряды 16—31	Разряды маски	Код длины	Выполнение инструкции
---------------------	-------------------------------	---------------	-----------	-----------------------

Прерывания от схем контроля машины (старое ССП 48, новое ССП 112, приоритет 1)

Машинный сбой	00000000 00000000	13	X	Подавляется
---------------	-------------------	----	---	-------------

Программные прерывания (старое ССП 40, новое ССП 104, приоритет 2)

Операция	00000000 00000001		1, 2, 3	подавляется
Привилегированная операция	00000000 00000010		1, 2	подавляется
Инструкция „Выполнить“	00000000 00000011		2	подавляется
Защита	00000000 00000100		0, 2, 3	подавляется
Адресация	00000000 00000101		0, 1, 2, 3	подавляется
Спецификация	00000000 00000110		1, 2, 3	подавляется
Данные	00000000 00000111		2, 3	подавляется
Переполнение с фиксированной запятой	00000000 00001000	36	1, 2	завершается
Деление с фиксированной запятой	00000000 00001001		1, 2	подавляется
Десятичное переполнение	00000000 00001010	37	3	или завершается завершается
Десятичное деление	00000000 00001011		3	подавляется
Переполнение порядка	00000000 00001100		1, 2	подавляется
Исчезновение порядка	00000000 00001111	38	1, 2	завершается
Значимость	00000000 00001110	39	1, 2	завершается
Деление с плавающей запятой	00000000 00001111		1, 2	подавляется

Прерывания при вызове супервизора (старое ССП 32, новое ССП 96, приоритет 2)

Код операции в инструкции	00000000 чччччччч			завершается
---------------------------	-------------------	--	--	-------------

Внешние прерывания (старое ССП 24, новое ССП 88, приоритет 3)

Таймер	00000000 1ппппппп	7	X	завершается
Клавиша прерывания	00000000 п1пппппп	7	X	завершается
Внешний сигнал 2	00000000 пп1ппппп	7	X—	завершается
Внешний сигнал 3	00000000 ппп1пппп	7	X	— „ —
Внешний сигнал 4	00000000 пппп1ппп	7	X	— „ —
Внешний сигнал 5	00000000 ппппп1пп	7	X	— „ —

Источник прерывания	Код прерывания, разряды 16—31	Разряды маски	Код длины	Выполнение инструкции
Внешний сигнал 6	00000000 пппппп1	7	X	завершается
Внешний сигнал 7	00000000 пппппп1	7	X	—, —
Прерывания от ввода — вывода (старое ССП 56, новое ССП 120, приоритет 4)				
Канал 0	00000000 ааааааа	0	X	завершается
Канал 1	00000001 ааааааа	1	X	—, —
Канал 2	00000010 ааааааа	2	X	—, —
Канал 3	00000011 ааааааа	3	X	—, —

Примечание: а — разряды адреса устройства; п — другие условия внешнего прерывания; ч — разряды полей R1 и R2 инструкции ВЫЗВАТЬ СУПЕРВИЗОР; X — значение этих разрядов непредсказуемо.

6.10. Пульт управления

Пульт управления (ПУ) предназначен для обеспечения наладочных и рабочих режимов машины, индикации ее состояния, контроля и управления процессором.

ПУ (рис. 6.10.1) разделен на три части: пульт оператора, пульт ручной работы и инженерный пульт.

Пульт оператора (ПО)

ПО предназначен для включения, отключения и индикации питания, индикации состояния машины и процессора, загрузки начальной программы, прерывания программы.

Дистанционное включение питания устройств машины производится нажатием кнопки ВКЛ ПИТ (Включение питания). После завершения операции „Включение питания“ при помощи специальной микропрограммы автоматически производится приведение машины в исходное состояние (сброс).

Отключение питания производится нажатием кнопки ОТКЛ ПИТ (Отключение питания). Аварийная ситуация в системе питания индицируется лампочками АВАРИЯ.

Нажатием кнопки ПЕРЕРЫВ (Прерывание) можно вызвать с ПУ внешнее прерывание текущей программы.

Для ввода начальной программы на регистре адресных переключателей ввода набирается адрес устройства ввода, а переключателем АДРЕС КАНАЛА выбирается канал. Нажатием кнопки ВВОД производится сброс машины в исходное состояние, запускается выбранное внешнее устройство. Процесс ввода индицируется лампой ВВОД, которая горит во время ввода и гаснет после загрузки начальной программы.

Пульт ручной работы позволяет оператору приводить машину в исходное состояние и вмешиваться в ход выполнения программы.

Приведение машины в исходное состояние производится нажатием кнопки СБРОС СИСТЕМЫ. При этом происходит обращение к микропрограмме сброса, которая сбрасывает регистр фиксации прерываний (РФП), регистр ошибок процессора (РОЦ), регистр очередности (РОЧ) и регистр приоритета обращения к памяти, счетчики (РС1—РС6), восстанавливается четность общих регистров, регистров с плавающей запятой и регистра слова состояния программы (РССП).

Нажатием кнопки ПОВТ РССП (Повторить РССП) можно повторить выполнение программы. При нажатии этой кнопки выполняются те же действия, что и при нажатии кнопки СБРОС СИСТЕМЫ, но кроме того, из нулевого адреса оперативной памяти считывается ССП начальной программы и заносится в регистр РССП. Таким образом ССП начальной программы снова становится текущим и программа повторяется.

Кнопкой УСТ РССП[40/63] (Установить РССП[40/63]) можно адрес, набранный на регистре адреса пульта управления заслать в РССП[40/63], т.е. изменить последовательность выполнения команд программы.

Нажатием кнопки ЗАПИСЬ можно производить занесение информации, набранной на регистре информации ПУ, в ячейку, определяемую адресом, набранным на регистре адреса ПУ, и положением переключателя выбора памяти (ОП, МП, ПКЗ, ПМК). Кнопка действует, когда выполнение программы приостановлено и горит лампа РУЧ (лампа ручного управления). Аналогично производится считывание информации при нажатии кнопки СЧИТ (Считывание).

Машина ЕС-1030 может работать в трех режимах — автоматическом, шаговом и одиночном. Выбор режима работы производится переключателем на три положения: ШАГ, АВТ, ОДИН.

В автоматическом режиме при нажатии кнопки ПУСК выполняется последовательность команд по программе. В шаговом режиме при нажатии кнопки ПУСК выполняется одна команда, обрабатываются незамаскированные прерывания и процессор останавливается.

В одиночном режиме при нажатии кнопки НАЧ (Начало) выполняется одна микрокоманда, адрес которой находится в регистре адреса ДЗУ.

Положение переключателя выбора рода работы можно менять только при нахождении процессора в состоянии „Стоп“.

Нажатием кнопки ОСТАНОВ можно остановить выполнение программы. Останов происходит после выполнения текущей команды (загорается лампа РУЧ). Если процессор не может закончить

выполнение команды из-за машинного сбоя или имеется непрерывный поток прерываний, нажатие этой кнопки не вызывает останова процессора.

Если останов процессора произошел после нажатия на кнопку останова, то выполнение программы можно продолжить нажатием кнопки пуска.

Пуск программы осуществляется нажатием на кнопку ПУСК.

Процессор при этом начинает работать в соответствии с положением переключателя рода работы. Если пуск производится после нормального останова и переключатель рода работы находится в положении АВТ (автоматический), то продолжается выполнение программы и гаснет лампа РУЧ. Если после сброса системы кнопку пуска нажать без предварительного введения нового адреса команды в РССП[40/63], то это приведет к непредвиденным результатам.

Можно остановить процессор при сравнении адресов команды или данных с адресом, набранным на регистре адреса пульта управления. Тип останова определяется переключателем на три положения: КОМ (Команда), ОБР (Обработка), ДАН (Данные). Если останов не требуется, то переключатель находится в положении ОБР. Если останов произошел по адресу команды, то в РССП[40/63] находится адрес следующей команды, которая еще не выполнена. При нажатии на кнопку пуска продолжается выполнение программы с этого адреса. Аналогично выполняется останов по адресу данных при положении переключателя ДАН. Если переключатель сравнения адресов находится в положении КОМ или ДАН горит лампа ПРОВ (Проверка).

Органы индикации, расположенные на пульте ручной работы, позволяют индицировать состояние ряда регистров процессора, а также выполнять индикацию адресов.

На пульте ручной работы расположены средства управления блока контроля, блока диагностики и каналов.

Инженерный пульт (ПИ)

ПИ содержит средства управления, которые используются только для инженерного обслуживания. Здесь в основном расположены органы управления, с помощью которых можно организовать разные режимы выполнения микрокоманд, непрерывное чтение микрокоманд по заданному с пульта управления адресу, останов по адресу микрокоманд и т. д. Ниже описывается назначение этих органов.

Тумблер останова по адресу ДЗУ. Для останова по адресу микрокоманды тумблер из положения ОБРАБОТКА переключается в положение ОСТ ДЗУ (Останов ДЗУ) на регистре адреса пульта управления (РА ПУ) набирается адрес микрокоманды, по которому требуется останов, переключатель рода работы устанавливается в положение ШАГ или АВТ и нажимается кнопка пуска. При совпадении адресов происходит останов.

Тумблер приема адреса в регистр РА ДЗУ. Этот тумблер действует при положении переключателя режимов работы ОДИН. Тумблер предназначен для приема адреса в РА ДЗУ с ПУ или следующего адреса из регистра РИДЗУ [56/63]. Когда тумблер находится в положении АДР ПУ; для приема адреса в регистр РАДЗУ из РАПУ [12/23] нажимается кнопка НАЧ. Если тумблер находится в положении АДР ДЗУ, то при нажатии на кнопку НАЧ происходит прием следующего адреса из регистра РИДЗУ [56/63].

Тумблер выполнения элементарной операции (ЦИКЛ ТАКТ). Тумблер действует при положении переключателя режимов работы в положении ОДИН. С помощью этого тумблера организуется выполнение микрокоманды в тактовом или циклическом режиме.

Переключатель РЕЖИМЫ ДЗУ. Переключатель используется в одиночном режиме для выполнения элементарных операций в полутакте С1, в полутакте С2 или в С1 и С2 одновременно. Переключателем можно пользоваться, когда адрес микрокоманды находится в РАДЗУ.

Тумблер непрерывного считывания с ДЗУ (ЧТ СЕР — чтение серий). С помощью этого тумблера в одиночном режиме обеспечивается постоянный сигнал приема в РАДЗУ адреса с пульта управления, что обеспечивает постоянное считывание микрокоманды из ДЗУ по этому адресу.

Тумблер блокировки дешифраторов (БЛ ДШ). При включении этого тумблера в одиночном режиме блокируются дешифраторы элементарных операций. При этом микрокоманда считывается с ДЗУ при нажатии кнопки НАЧ, но не выполняется.

На пульте управления кроме перечисленных органов управления располагаются органы проверки оперативной памяти, позволяющие проверить ОП в различных режимах записи считывания: нулей, единиц, тяжелого и обратного тяжелого кода и каналов.

ОГЛАВЛЕНИЕ

	Стр.
Глава 6. Процессор ЭВМ ЕС-1030	3
6.1. Назначение и технические характеристики процессора	3
6.2. Структурная схема и краткая характеристика основных блоков процессора	4
6.3. Взаимодействие блоков процессора при выполнении команд	10
6.4. Блок арифметическо-логический	13
6.5. Блок синхронизации	38
6.6. Блок микропрограммного управления	40
6.7. Блок управляющих регистров и блок прямого управления	49
6.8. Блок обращения к оперативной памяти	57
6.9. Микропрограммы процессора	65
6.10. Пульт управления	82

Редактор *Фромченко Р. Е.*

Корректор *Молчанова М. А.*

Сдано в набор 10 декабря 1976 г. Подписано к печати 21 февраля 1977 г.

Г-171709 Объем 5,38 п. л., 4,07 уч.-изд. л. Зак. 799.

Бесплатно/

Типография ВАС